

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
им. ГУМАРБЕКА ДАУКЕЕВА»
Кафедра IT – инжиниринг

«ДОПУЩЕН К ЗАЩИТЕ»
Зав. кафедрой PhD, доцент Досжанова А.А.
_____ « ____ » _____ 2020 г.

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка информационной системы «Турист-путеводитель»
Специальность 5В070400 – Вычислительная техника и программное обеспечение
Выполнила: Керимбекова Ж.А. Группа ВТ-16-2
Научный руководитель: PhD, доцент Досжанова А.А.

Консультанты:

по экономической части: к.э.н., профессор Габелашвили К.Р.
(учёная степень, звание, Ф.И.О.)
_____ « ____ » _____ 2020 г.

по безопасности жизнедеятельности: доцент Приходько Н.Г.
(учёная степень, звание, Ф.И.О.)
_____ « ____ » _____ 2020 г.

по программному обеспечению: ст.преп. Майкотов М.Н.
(учёная степень, звание, Ф.И.О.)
_____ « ____ » _____ 2020 г.

Нормоконтролер: ст.преп. Абсатарова Б.Р.
(учёная степень, звание, Ф.И.О.)
_____ « ____ » _____ 2020 г.

Рецензент: _____
(учёная степень, звание, Ф.И.О.)
_____ « ____ » _____ 2020 г.

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
им. ГУМАРБЕКА ДАУКЕЕВА»

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5В070400 – «Вычислительная техника и программное обеспечение»

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту Керимбековой Жание Әнуарқызы

Тема проекта: Разработка информационной системы «Турист-путеводитель»

Утверждена приказом по университету № ___ от «___» _____ 2020 г.

Срок сдачи законченного проекта «___» _____ 2020 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): руководство по программированию на JAVA, интегрированная среда Android Studio, данные преддипломной практики.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта:

- а) анализ и исследование предметной области;
- б) проектирование приложения;
- в) реализация приложения;
- г) экономическое обоснование эффективности;
- д) вопросы безопасности жизнедеятельности и охраны труда.

Перечень графического материала (с точным указанием обязательных чертежей): имеется 13 таблиц и 35 иллюстраций.

Основная рекомендуемая литература:

1 Герберт Шилдт. Java. Полное руководство, 10-е издание = Java. The Complete Reference, 10th Edition. — М.: «Диалектика», 2018. — 1488 с.

2 Кей С. Хорстманн. Java SE 9. Базовый курс = Core Java SE 9 for the Impatient. — М.: «Вильямс», 2018. — 576 с.

3 Кей С. Хорстманн. Java SE 8. Вводный курс = Java SE 8 for the Really Impatient. — М.: «Вильямс», 2014. — 208 с.

Консультация по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Габелашвили К.Р	24.04.2020	
Безопасности жизнедеятельности	Приходько Н.Г	24.04.2020	
Программная часть	Майкотов М.Н	14.05.2020	
Нормконтролер	Абсатарова Б.Р	18.05.2020	

ГРАФИК

подготовки дипломной работы (проекта)

Наименования разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечания
Анализ и исследование предметной области	01.11.2019– 20.12.2019	
Проектирование приложения	21.12.2019 – 20.02.2020	
Программная реализация	21.02.2020 – 20.04.2020	

Дата выдачи задания « ____ » _____ 2020 г.

Заведующий кафедрой _____ А.А.Досжанова
(подпись)

Научный руководитель проекта _____ А.А.Досжанова
(подпись)

Задание принял к исполнению студент _____ Ж.А.Керимбекова
(подпись)

АҢДАТПА

Дипломдық жоба Алматы қаласының тұрғындары мен туристеріне арналған ақпараттық жүйені әзірлеуге арналған. Әзірленген бағдарлама пайдаланушыға өз маршрутын құруға көмектеседі, туристерге міндетті түрде баратын орындар тізімін ұсынады. Туризм тақырыбы өзекті, бірақ қазіргі уақытта ішкі туризм даму кезеңінде.

Дипломдық жоба Android – қосымшасы форматында орындалды. Мобильді қосымша туристік орындар туралы ақпараттылық, жаңа туристерді тарту тиімділігін арттыру үшін құрылған. Дипломдық жобаны іске асыру үшін Firebase, Android studio, Javascript сияқты технологиялар қолданылды.

Бұдан басқа, бітіру жұмысында Қазақстандағы ішкі туризмге талдау жүргізіледі, сондай-ақ өндіріс жүйесін енгізуден өмір сүру қауіпсіздігі мен экономикалық тиімділік мәселесі қарастырылады.

АННОТАЦИЯ

Дипломный проект посвящен разработке информационной системы для горожан и туристов города Алматы. Разработанная программа поможет пользователю построить свой маршрут, предложит список мест, которые обязательно посещают туристы. Тема туризма актуальна, но на данный момент внутренний туризм находится на этапе развития.

Дипломный проект был выполнен в формате Android – приложения. Мобильное приложение создано для информативности о туристических местах, повышения эффективности привлечения новых туристов. Для реализации дипломного проекта были применены такие технологии, как Firebase, Android studio, Java script.

Кроме того, в выпускной работе проводится анализ внутреннего туризма в Казахстане, а также рассматривается вопрос безопасности жизнедеятельности и экономической эффективности от внедрения системы в производство.

ABSTRACT

The diploma project is dedicated to the development of an information system for citizens and tourists of Almaty. The developed program will help the user to build their own route and offer a list of places that tourists must visit. The topic of tourism is relevant, but at the moment domestic tourism is at the stage of development.

The diploma project was made in the format of an Android application. The mobile app is designed to provide information about tourist places and increase the effectiveness of attracting new tourists. Technologies such as Firebase, Android studio, and Javascript were used to implement the diploma project.

In addition, the final work analyzes domestic tourism in Kazakhstan, as well as considers the issue of life safety and economic efficiency from the introduction of the system into production.

Содержание

Введение	8
1. Аналитическая часть	9
1.1 Обзор и анализ туризма в Казахстане	9
1.2 Решения ИТ для сферы туризма	Error! Bookmark not defined. 3
1.3 Анализ существующих информационных систем. Error! Bookmark not defined.	Error! Bookmark not defined. 8
2. Проектирование программного обеспечения	19
2.1 Требования к программному продукту	19
2.2 Структура программного обеспечения	20
2.3 Построение UML - диаграмм	21
2.4 Среда разработки FireBase	25
2.5 Язык программирования Java	26
2.6 Операционная система Android	27
3. Разработка программного обеспечения	30
3.1 Установка Android Studio	30
3.2 Интерфейс программного продукта	36
3.3 Тестирование	47
4. Технико-экономическое обоснование	48
4.1 Определение объема и трудоемкости разработки ПП	48
4.2 Расчет затрат на разработку ПП	48
4.3 Сравнительный анализ эксплуатационных затрат до и после внедрения	52
4.4 Вывод по экономической части	55
5. Безопасность жизнедеятельности	56
5.1 Анализ потенциально опасных и вредных факторов в офисе воздействующих на персонал	56
5.2 Расчет интегральной оценки условий труда и определение тяжести труда	57
5.3 Расчет искусственного освещения	58
5.4 Расчет уровня шума	60
5.5 Вывод по части БЖД	62
Заключение	63
Список литературы	64
Приложение А (Техническое задание)	66
Приложение Б (Листинг программы)	67
Приложение В (Акт внедрения)	80

Введение

Среди разнообразия развития рынка есть ряд, способов выработки, новых направлений деятельности предприятий вызывают значительное внимание к научным исследованиям и практическим нововведениям. На протяжении последних лет быстро развиваются информационные технологии.

В период всего времени использование компьютерных систем привело к созданию тенденции на высоконадежный набор, направленных на возможность использования информационных систем. Это продвижение сказалось на высоком процессе разработки различных, встроенных в уникальные объекты информационно технологических систем. В данное время информационные технологии являются присущей частью жизни. Информационные технологии, сейчас приобретают особую важность системы с обработкой информации для всех видов деятельности, особую важность имеют в общественной жизни. На сегодняшний день очень трудно выдвинуть организацию не применяющую социальные сети и сайты. Важный момент разработки информационной системы, это рассмотреть данную область. Определить вид деятельности, сделать полный анализ изучаемой нами области, выявить и рассмотреть все недостатки, входящие в сущность. Рассмотреть зависимость между сущностями, создавая визуальное видение исследуемой задачи.

Целью работы является разработка информационной системы для туристов города Алматы “Welcome to Kazakhstan”. Изначальный этап разработки информационной системы составляет этап — это изучение, анализ деятельности туризма для возможного совершенствования и упорядочение методы работы.

Разработанное мобильное приложение должно быть удобным в использовании и предельно понятным для пользователей. Пользователь не может вносить корректировки в структуру и схему данных.

Разработанная информационная система должна состоять из таких функций как:

- сформировать предложение для иностранного туриста;
- помощь в выборе места исходя из запросов;
- предоставить достоверную и полную информацию выбранного места;
- возможность бронирования авиабилетов;
- возможность бронирования отеля;
- предоставить готовый маршрут к выбранному месту.

1 Аналитическая часть

1.1 Обзор и анализ туризма в Казахстане

Сфера туризма приходится одной из высокодоходных и активно развивающихся бизнес-сегментов международных услуг. Учитывая динамичный и стабильный рост туризма, воздействие на экономику общественного сектора и уровень жизни общества, Правительство решило включить отрасль туризма Казахстана в приоритетную программу. Наиважнейшим направлением Республики Казахстан является отрасль –туризм. В Казахстане главное задачей развития сферы туризма является организовать конкурентоспособную туристическую базу, на основе которой будет реализоваться условия развития туризма, включать такие сферы как экономика и дальнейшее развитие на международном рынке.

Казахстан имеет все возможности для продвижения разных видов туризма, учитывая природный потенциал: экологического, делового, экстремально и культурно-развлекательного. В последние годы отрасль туризма в Казахстане оценивается как поступательный и стабильный, состояние отмечено международными экспертами в сфере туризма. В ближайшее время нашу страну можно рассмотреть, как одну из известнейших экономик путешествий. Иностранные эксперты рассматривают продвижение туризма в Казахстане и выделяют такие плюсы республики как история и богатство культурного наследия, гостеприимство, дружелюбность и разнообразие туристкой инфраструктуры[1].

Туризм входит в несырьевой сектор экономика, Президентом Республики Казахстан поставлена задача развитие этого важнейшего сектора. В ответственную задачу входит: в скором времени войти в число 50 конкурентоспособных стран мира. Важно понимать – туристский комплекс сможет стать мощнейшим ускорителем темпа роста экономики республики.

Количество обслуженных посетителей по типам туризма по итогам января-сентября
(тыс человек, по данным Комитета по статистике РК)

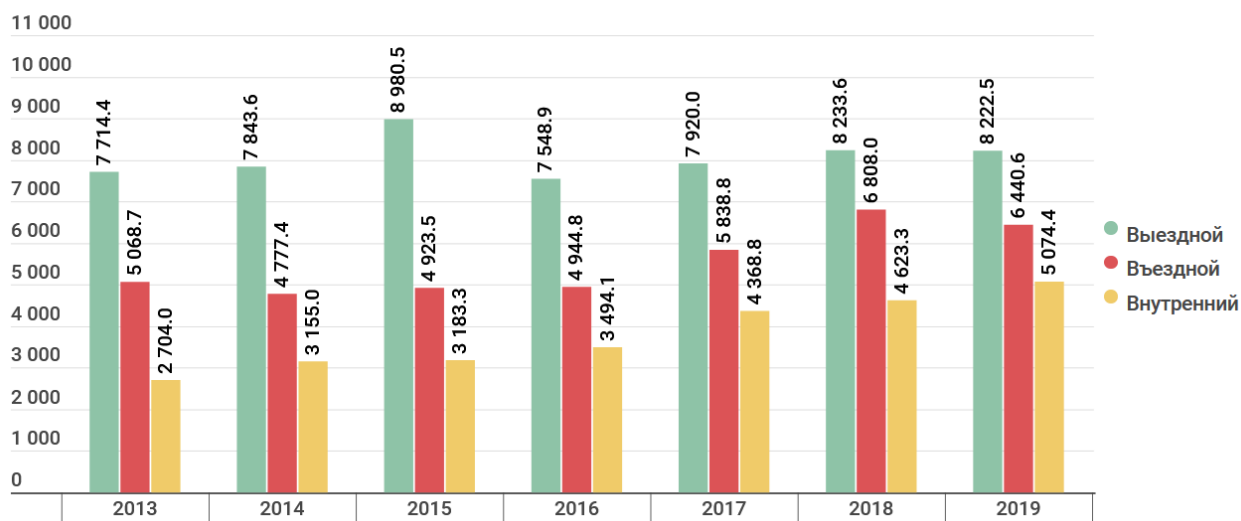


Рисунок 1 – Данные по видам туризма 2013-2019 годах

Рассмотрев данные приведенные в таблице, показано, что количество въехавших в страну иностранных туристов в 2019 году увеличилось на 1372 по сравнению с 2013 годом. В этот же период рассмотреть, объем внутреннего туризма увеличился на 2370 человек, что в процентном содержании %. В тенденции показателей объема выездного туризма важно отметить незначительный рост: так в 2019 году по сравнению с 2013 годом число выездных туристов увеличилось на 508 человек.

Выбор региона для разрабатываемого приложения.

Алматы – Южная столица Казахстана, – известная китайцам, арабам, грекам еще с времен тысячелетий нашей эры, прославлен одним из древних городов Семи Рек. Долгое время по городу проходил разноязычный караван Шелкового пути- связывающий культуры Востока и Запада. В Алматинской области находятся редкие ресурсосочетания для современного туристического спроса. Алматы - прекрасный город-сад, возникнувший в райских местах великого Тянь-Шаня. Тянущиеся в небо снежные вершины гор Алатау, величественно глядят на город с самых высот. Аромат садов и знаменитых алатауских тополей и елей, завораживающие пейзажи и звучание горных рек не оставят никого равнодушными к Южной столице.

На территории Алматы распределились 23 главных ландшафтных зон для различного функционирования, степени и сборников благоприятности круглогодичного туристической деятельности. На других территориях организуют научный туризм, специфические виды туризма по сезонам и другим

условиям. К особым объектам туризма относятся парки, заповедники, государственные и природные хозяйства.

Наибольший интерес к Южной столице проявляют туристы с таких стран как: США, Россия, Индия, Германия. Туристов с этих стран притягивает экотуризм Казахстана. Алматы представляется центром туризма: за 9 месяцев посетили 984,7 туристов, не только по причине природы, но и различных активностей. Ниже приведем статистику прибытия туристов в города Казахстана.

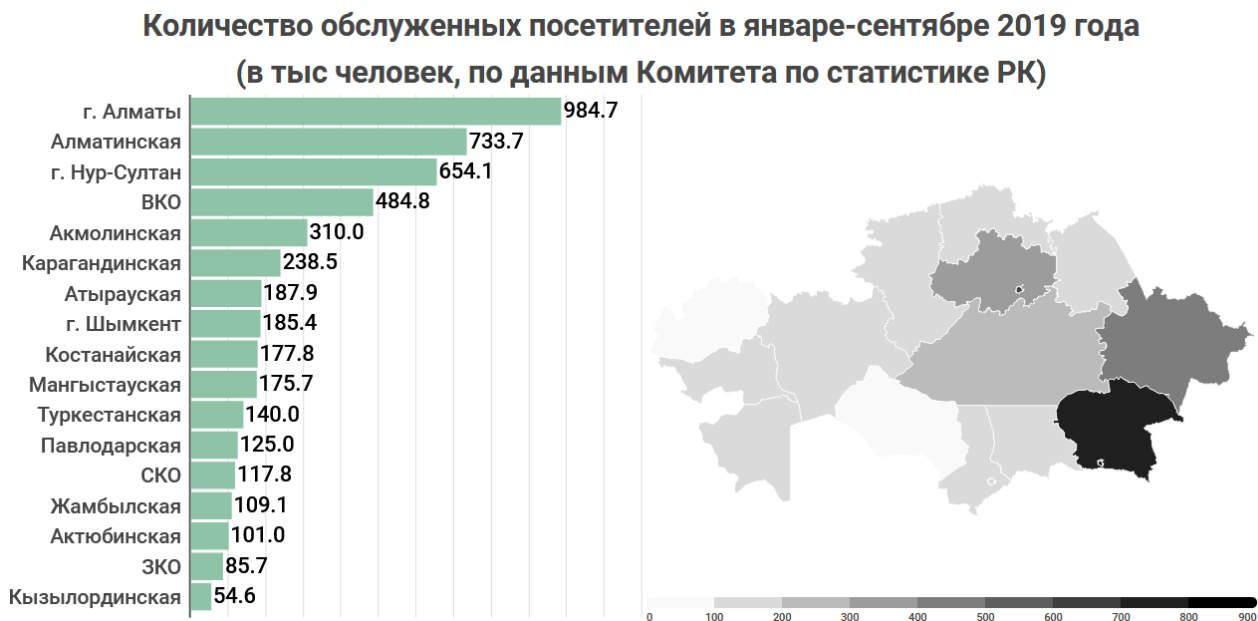


Рисунок 2 – Количество посетителей за январь – сентябрь 2019 года

Важно отметить, что столица выходит на третье место – пользовались услугами разного рода отелей 654.1 человек. Алматинская область заняло второе место, больше на 80 тыс. человек. Меньше клиентов приходит на Кызылординскую область, всего 54.6 человек. Привлечение инвестиций в сферу туризма, сперва необходимо выработать план действий по достижению целей и каждый регион страны стремился для вхождения Казахстана в ряд конкурентоспособных, популярных туристских кластеров.

Произведя анализ, нужно отметить следующие проблемы для привлечения инвестиций.

1. недостаточное продвижение информационных систем в сфере туризма;
2. турфирмы предпочитают заниматься развитием выездного туризма;
3. несоответствующая стоимость качества;
4. недостаточная PR реклама среди иностранных гостей.

Основные цели, которые мы должны придерживаться для этого вида деятельности:

- развитие и усиление дружеских и культурных отношений с зарубежными странами;
- привить нашему населению уважения и чувства патриотизма иностранными гражданами Республики Казахстан;
- укрепление и повышение интеллектуального и духовного уровня жизни;
- привлечь население к активным действиям и занятиям спортом;
- сформировать бережное отношение к природе, гражданами и туристами.

Типы программ для туристов

В зависимости от цели деятельности туризма, информационные системы можно определить на четыре типа:

- 1) познавательный;
- 2) оздоровительный;
- 3) активный;
- 4) рекреационный.

Познавательный вид туризма предназначены для осмотра памятников истории с пребыванием в одном городе, а маршрутные – пребывание в нескольких городах с маршрутами по достопримечательностям.

Оздоровительный вид туризма направлен на улучшения состояния туриста, самое массовое медицинское направление, является экологическим и входит в организацию оздоровительного комплекса.

Активный туризм имеет спортивную цель, развит в странах где сохранены уникальные ландшафты. Активный вид отдыха представляет собой – на лыжах, восхождения в горы, на лыжах, на велосипеде.

На рекреационный вид туризма играет важнейшую роль природный потенциал региона. Для многих стран этот тип туризма приходит самым распространенным и массовым.

В данной дипломной работе необходимо разработать информационную систему для туристической области. При разработке информационной системы главной целью является высокая скорость определения запросов по месту достопримечательности, информации, просмотра услуги и т.д.

В качестве предметной области рассматривается сфера туризма.

В последние годы наблюдается рост в сфере туризма, существенно повысилась статистика в мировой экономике. “Экономический и социальный феномен XX века” термин чаще начал использоваться в деятельности туризма в научной и современной литературе. Повысить преимущества позволяющих рост “новой эры туризма”, является разработка совершенного информационного проекта. Современные пользователи мобильных устройств склонны к организации собственных туров и выбору альтернативных каналов, нежели традиционным покупкам туров.

В идеале, информационная система должна решить следующие задачи:

1. увеличить поток туристов в несколько раз;
2. повысить туристический потенциал города;
3. развитие экотуризма;
4. предоставить уникально отобранные места в городе.

1.2 Решения ИТ для сферы туризма

Формируя главный фактор конкурентоспособного туристического рынка, эффективные информационные технологии, дают возможность развитию внутреннего и въездного туризма страны. Республика ежегодно теряет возможных клиентов, поднятия экономики на высший уровень из-за низкого уровня осведомленности о туристских возможностях Казахстана.

Введение в комплекс мер развитие для информационно технологической стороны.

Информационные технологии с отличительными свойствами, имеют значение в развитии общества, выделить следующие наиболее важные, целесообразные представления.

1) Информационные системы дают возможность активизировать и эффективное пользование информационными ресурсами общества, которые входят в наиболее важный фактор. Практика показывает, что эффективное использование информационных систем приводит к существенной экономии других видов ресурсов: энергии, полезные ископаемые, научные знания, социального времени.

2) Информационные системы оптимизируют процессы, которые занимают все большее место в обществе. Известно, что продвижение цивилизации вверх приходит на становление информационного общества, в котором ценности не материальные, а преимущественно, информация и знания.

Большинство развитых стран вынуждены осваивать информационные технологии, для дальнейшей деятельности, которые в той или иной мере связаны друг с другом.

Мобильные устройства справляются со своей функцией – являясь средством коммуникации. В то же время, основательно вошедшие в нашу жизнь смартфоны не только средство для коммуникации, но и настолько же функциональны. Смартфоны являются копией компьютера, в функции которого входят поиск необходимой нам информации используя Интернет-ресурсы.

В современных условиях смартфоны, планшетные ПК и другие виды устройств производят на базе ОС Android. Проект данной дипломной работы был разработан на базе ОС Android в виде приложения на смартфон. В данный момент

много туристских сайтов города Алматы, но нет удобного, информационного приложения для смартфонов.

Получая информацию, о привлекательных особенностях о местонахождении важно отметить о необходимости разного рода видео средств. В конечном итоге применяя информационные технологии возрастает качество туристических услуг, не изменяя их человеческого содержания.

Информационные системы представляют функционирование авиакомпаний. Важная роль в организации, контроля и управления авиа-операциями большую роль играют информационные системы, при получении информации и бронировании авиабилетов.

Электронный резерв, введение информации по месту, оценка уровня обслуживания в гостинице невозможно представить без информационных технологий. Компьютеры, смартфоны используются для резервирования, введении учета и контроля. Вышеуказанные системы включены в разработанное приложение, для оптимизации путешествий туристов.

Анализ существующих информационных систем для сферы туризма

Представителями туристических услуг применяется ряд информационных технологий, в целях обеспечить спутниковое оповещение всех международных событий, ведение бизнеса с помощью средств сообщения (например, допустимость сделать звонок в любой части мира). Чтобы получить информацию о месте пребывания, узнать о его особенностях также необходимы средства информационных технологий. В случае чего применение современных технологий в сфере туристических услуг возрастает, не изменяя их человеческого содержания.

Информационные технологии играют огромную роль в развитии деятельности авиакомпаний. Обеспечивают организацию, управление и контроль авиаопераций в информационной системе, помогают планировать маршруты и ведут анализ расписания. Включают в себя начиная с контроля систем перевозок до покупки авиабилетов. Снабжение высокими информационными система также относится к гостиничному бизнесу, в современное время высокий уровень достигается с применением новых систем. Новые информационные системы учитывают автоматизацию гостиниц, резервирование онлайн, введение новых способов улучшения обслуживания в сфере гостиничных комплексов. Обеспечить высокий уровень обслуживания в гостиничном комплексе возможен с помощью информационных систем. Новые системы берут на себя ответственность в автоматизации гостиничных комплексов, резервирование онлайн, введение новых технологий, способствует улучшению качества даже при сокращении персонала. Автоматизированные системы включают в себя: повышение уровня производительности, уровня полученных знаний у высоких управленческих работников; Компьютерные системы широко распространены в информационных центрах гостиниц. Благодаря высокому уровню развития

вычислительных систем, происходит учет посетителей, контроль за инвентарем и учет поставки питания.

Разработка мобильного приложения необходима во многих сферах, для более комфортной жизни. Мобильные приложения имеют высокий спрос и в сфере путешествий. Когда под рукой нет карты, ноутбука или справочника, смартфон – поможет найти дорогу или информацию в незнакомом городе, тем самым сэкономят не только деньги, но и время.

Приведем анализ существующих информационных систем:



Рисунок 3 – Информационный сайт «visitalmaty.kz»

Информационный сайт «visitalmaty.kz» разработан центром Казахстанской ассоциации туристов, специально для горожан и гостей города (рисунок 3). Целью центра было предоставить профессиональные интернет-консультации о туристическом потенциале города Алматы, куда входят карта, путеводители, карта горных маршрутов и информация по туроператорам. Также участвуют в составлении экскурсий и выпуском сувенирной продукции. Для иностранных гостей выдают SIM-карт мобильной связи. По словам разработчиков, опыт

центра основывается на иностранных развитых центрах, как: Дубай, Лондон, Париж и Барселона. Информационный сайт функционирует через Интернет ресурсы и браузера, в дальнейшем будет в использовании на семи языках. На сайт можно найти основную информацию о городе, достопримечательностях, популярной национальной кухне и всегда обновляется календарь мероприятий.

Недостатком разработки является в том, что он ориентирован только для Интернет браузера, не имеет приложения для смартфона что было бы очень удобно. Для дальнейшего развития сайта необходимо информировать горожан и гостей в соц. сетях.

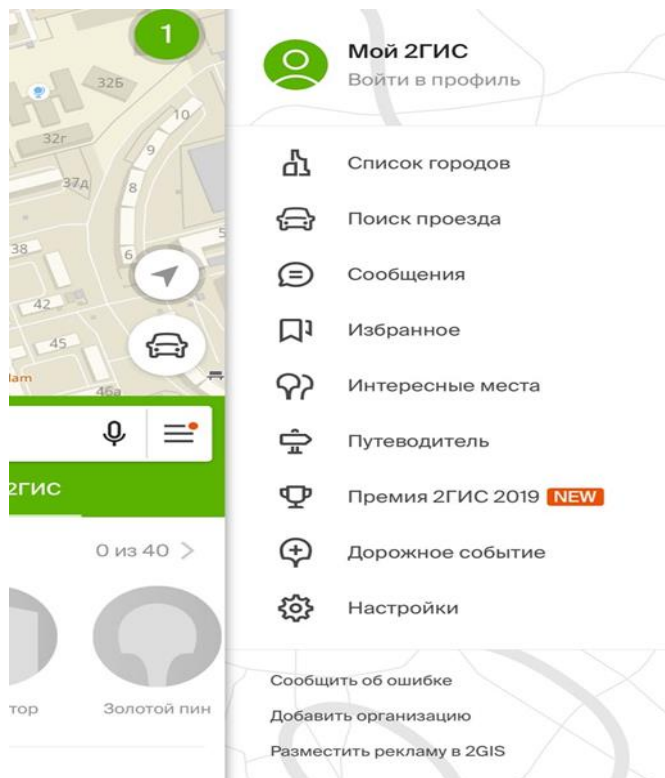


Рисунок 4 – Интерфейс мобильного приложения «2 ГИС»

«2ГИС» это справочник организаций, офлайн- карта, разработанная международной картографической компанией, выпускающая одноименные карты для смартфонов с 1999 года (рисунок 4). Положительные стороны приложения:

- является справочником, показывает организации, время работы и вход в здание;
- детальная карта города в офлайн режиме;
- позволяет построить удобные маршруты на авто или автобусе;
- ведет онлайн маршрут во время пробок, уведомляет о камерах ГИБДД на дорогах города. Приложение доступно на платформы Android, iOS, Windows

Phone, при скачивании не требует максимальных технических требований. С 2016 года компания прекратила улучшать разработанное приложение для Windows Phone.

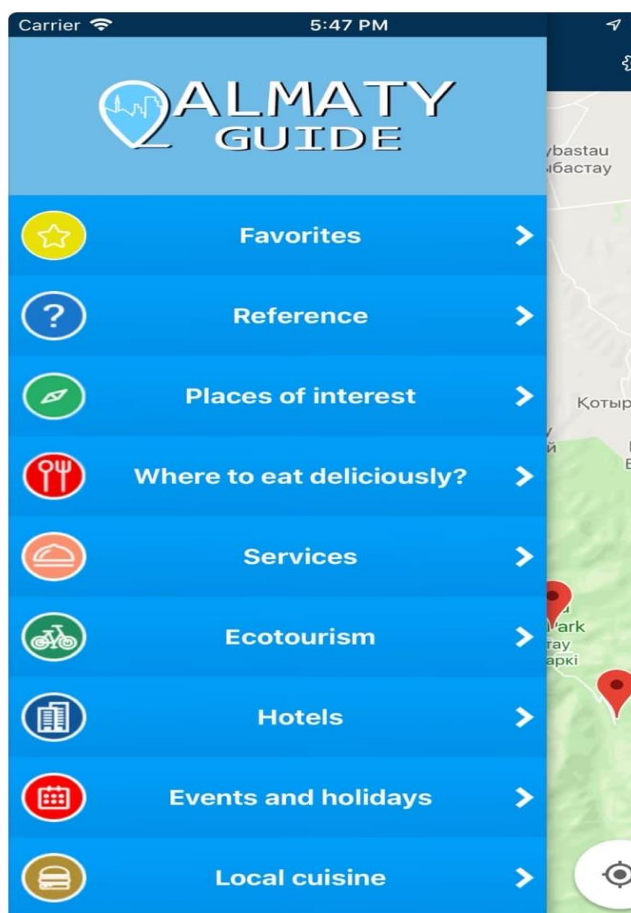


Рисунок 5 – Приложение «Almaty Guide»

Еще одно бесплатное приложение «Almaty Guide» появилось в 2018 году (рисунок 5) – простое и удобное приложения включило в себя информации по интересующим местам и с добавлением понравившийся заведений в городе. Вся информация предоставляется сайтом «www.inalmaty.kz». Узнавать о важных событиях города, находить развлечения по категориям, просмотр свежих фотоотчетов и мероприятий города можно через это приложение. Минус приложения в том, что не часто обновляется для исправления ошибок, при этом приложение очень долго работает.

Из этого следует, что в настоящее время существуют информационные системы для туристов, но нет востребованности из-за низкого качества. Существующие приложения должны пройти обновления и внести нововведения для удобства горожан и гостей города. В то же время заметен интерес к путешествиям через приложения на смартфоне.

1.3 Постановка задачи

В данной дипломной работе, создавая значительное, информативное и безопасное приложение, использовали современные технологии которые позволяют создать приложение с минимальными затратами.

Требования при разработке информационной системы (приложения) туристической компании:

- интерфейс приложения должен быть интуитивно понятным и удобным для пользователей;
- приложение должно полностью соответствовать к целевой аудитории;
- доступ администраторам к изменению исходного кода, для дальнейшего его улучшения;
- в приложении должна быть возможность бронирования отеля онлайн, включить возможность покупки авиабилетов онлайн;
- предусмотрена административная часть, где будет идти контроль за действиями в приложении;
- приложение должно выдавать полную информацию по выбранному месту, туру;
- исходный код должен комментироваться, для дальнейшего его изменения;
- приложение должно показывать пользователю его местонахождение.

2 Проектирование программного обеспечения

В данной части представим и охарактеризуем этап проектирования базы данных. Описать функциональную структуру приложения и основание программного обеспечения. Необходимо рассмотреть типы базы данных, какое значение будет оказывать в дальнейшей разработке программного обеспечения. Важно учитывать все преимущества и недостатки базы данных.

В нашем разрабатываемом приложении необходимо рассматривать предметную область: перечень сущностей и базу данных. База данных в первую очередь должна иметь связь между категориями информации.

2.1 Требования к программе

Основываясь на свойства разрабатываемой информационной системы, отвечают последующим требованиям:

- возможность перенесения базы данных без изменений на новую разработанную платформу;
- информационная система имеет полную связь с Базой данных;
- информационная система работает в настоящем времени, с операциями ввода\вывода, чтение, запись;
- программное обеспечение должно быть понятным и простым для администратора, для дальнейшей его работы с ИС;

Выполнение вышеуказанных требований позволит разработать простую и высокоэффективную информационную систему, для повышения производительности труда программистов и в свою очередь высокий спрос от клиентов. Функционирование модели данных, в составе программы написан как приложение на ОС Android, на языке Java и с использованием данных на базе Firebase.

Разработанное приложение работает с двумя модулями:

- модуль управления;
- модуль экранных форм.

Модуль управления реализовывает интерфейс пользователя, составляет запросы и предоставляет ответы на него.

Модуль экранных форм получает вопрос от клиента, делает обработку и после отправляет подходящую страницу с использованием.

2.2 Структура программного обеспечения

При разработке приложения структура программного обеспечения включает в себя различные инструменты и программные основы.

Инструменты при программном обеспечении можно разделить по группам:

- системное: определено для обширных действий в использовании. К ним относятся операционные системы с подтипами, для решения проблем в механизме есть система техническим образом действия.

- дополнительный: ПО предназначено для выполнения и связи с разработчиком в виде текстовых редакторов, видео, графических редакторов и т.д.;

- инструментальное: структура программ для разработки системного программного обеспечения. Необходимые системы для создания, корректирование кода и проводить запуски программ. На данный момент востребованы программы с различными инструментами, языками программирования и связью с базой данных.

С обозначения структуры приложения и начинается разработка программного продукта. В первую очередь это необходимо для представления программы о его составе и как можно будет облегчить создание системы. Необходимо знать за что отвечает каждый компонент, в виде компонентов заключаются вкладки. Структурная схема представляет состав и связь между частями. Структурная схема отображена на рисунке 2.1. все действия выполняются в главном окне, для удобства в использовании. Интерфейс разработан понятливым и простым, комментируя на каждом этапе, для дальнейшего развития приложения.

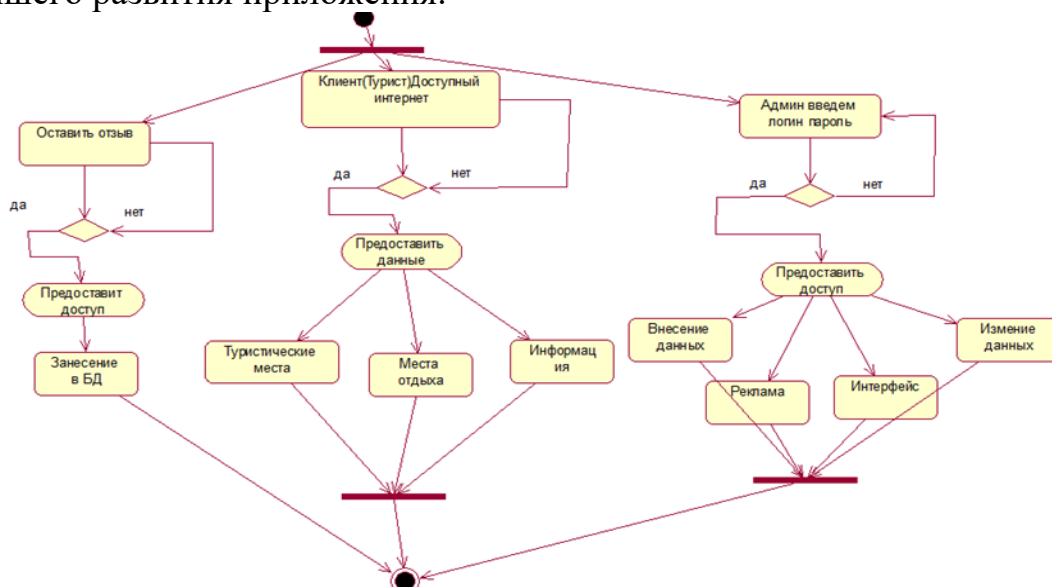


Рисунок 2.1 – Структурная схема программы

Если приложение для пользователя имеет доступ к интернету, приложение сможет предоставить данные. Данные по туристическим местам, местам отдыха и полную интересующую информацию. Предоставив доступ клиенту, при написании отзыва о месте, отзыв внедряется в базу данных.

Приложение для пользователя может оставлять заявки на бронирование авиа - билетов и отеля. Приложения для администратора должно входить в базу через логин и пароль. После ввода логина и пароля предоставляется доступ к внесению данных и внедрению рекламы. Изменение данных осуществляется в разработке кода, который написан с комментариями на каждом его этапе. Удобно тем что, приложение сможет обновляться вне зависимости от разработчика. Обновление интерфейса по всем параметрам соответствующего дизайна, также будет обновляться от его необходимости. В это входят обновление и удаление фотографий достопримечательностей, добавление новых мест в городе. Приложение было разработано на операционной системе Windows 10. Для создания диаграмм мы использовали веб-программу “draw.oi” от американской компании Microsoft office. Разработка главного программного обеспечения создавалось на ОС Android Studio. Для создания, облегчения и скорости разработки важную роль играет инструментальный вид программного обеспечения.

2.3 Построение UML-диаграмм

Во многих методах объектно-ориентированного проектирования используют язык UML. В конце 80-х и в начале 90-х разработали унифицированный язык моделирования (UML, Unified Model Language), который являлся приемником языков объектно-ориентированного языка проектирования. По стандартам моделирования прошел OMG (Object Management Group), сейчас обладает стандартом OMG. UML один из универсальных языков для моделирования и анализа системы в программном обеспечении. На данном языке есть возможность для описания классов, объектов и компоненты в разных областях, передавая аспект разрабатываемого приложения. Для дальнейшей разработки приложения, необходимо рассмотреть структуру и выявить все возможности для развития приложения.

Диаграмма видов использования приложения указана на рисунке 2.2. В приложении связаны два актера – администратор и турист. Администратор- это человек отвечающий за все действия в ПО и с доступом ко всем категориям. Пользователь –турист установивший мобильное приложение для поиска интересующих его мест.

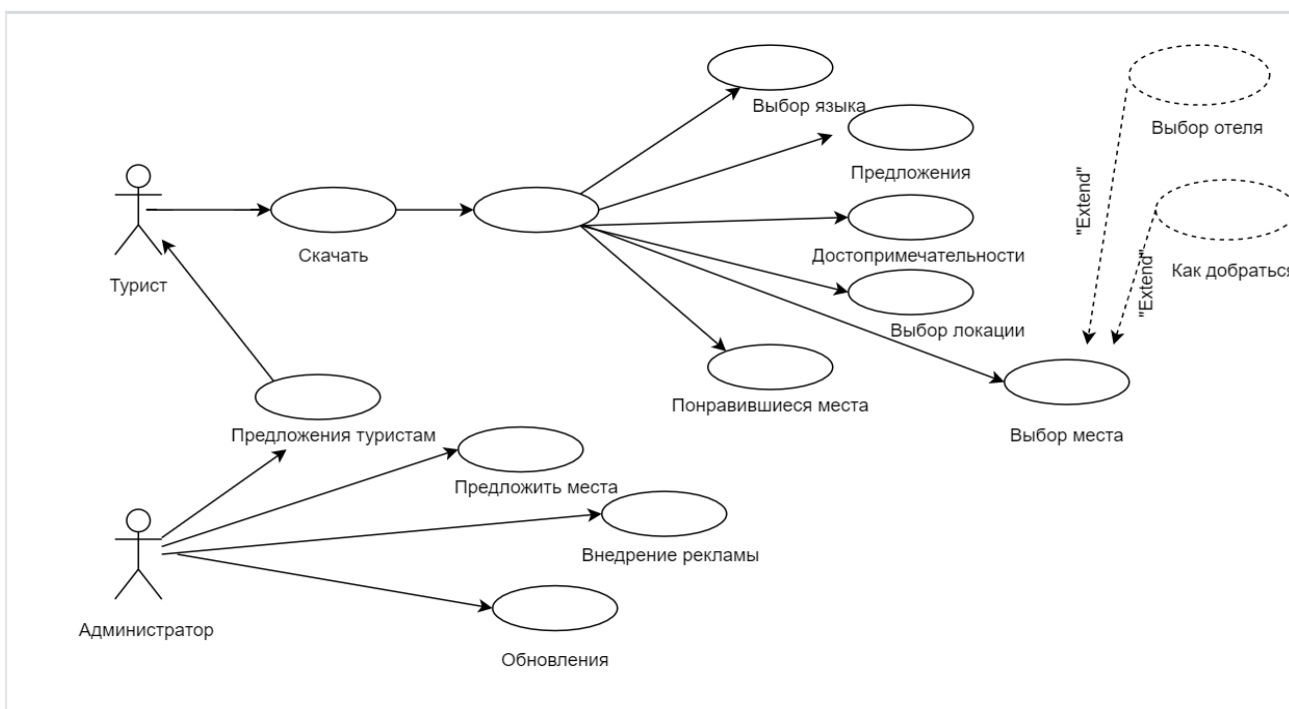


Рисунок 2.2- Диаграмма вариантов использования приложения

Сплошные линии указывают на отношения между ассоциациями, которые показывают вариант использования. Пользователь, запустив приложение имеет возможность переключаться между вкладками.

Пунктирными линиями указывают связь расширения (extend) или включения (include), которые возникают между вариантами использования.

Отношение включения (include) появляется, когда один вариант использования включается в другой точке и варианты становятся компонентами. Отношение расширения (extend) отображает два варианта использования в приложении. Только при определенных условиях используется вариант расширения. На рисунке 2.2 расширяющим элементом будет «Выбор отеля», который не является вариантом использования – «Выбор места».

Пользователь - турист после запуска выбирает язык и может воспользоваться готовыми предложениями для туристов. Может посмотреть информацию о достопримечательностях, т.е. телефон, часы работы, адрес и как проехать к выбранному месту. Пользователь также может воспользоваться вкладкой понравившийся мест. При выборе категории, перед ним выйдет готовый список рекомендованных для посещения. Пользователь будет иметь доступ к осуществлению бронирования авиабилета и выбором, бронированием отеля. Может посмотреть свои заявки на бронирование на сайте booking.com. Пользователь сможет приобрести авиабилета на сайте aviasales.ru, нажатием кнопки на консоли.

Администратор следит за обновлениями, внедряет рекламы и обновляет

информации по месту. Имеет полный доступ к кодам, коды комментированы на каждом этапе, для дальнейшего развития приложения.

Основой программной системы является – класс. Класс имеет отношение и к языкам программирования, для выполнения автоматически генерации кода, в соответствии с классами UML и программными классами. Класс имеет атрибут, операции и наименование. В виде диаграммы выглядит в виде прямоугольника, с тремя областями, верхняя включает в себе название, в средней – описание свойств (атрибут), в нижней – название операций, выполняемых объектами класса.

Для показа связи между классом-потомком и классом- родителем используют на диаграммах класса обобщение. Внедряют на диаграмму, при обнаружении нескольких классов, разных видов, при этом общие элементы поднимаются на уровень выше и образуют класс-родитель.

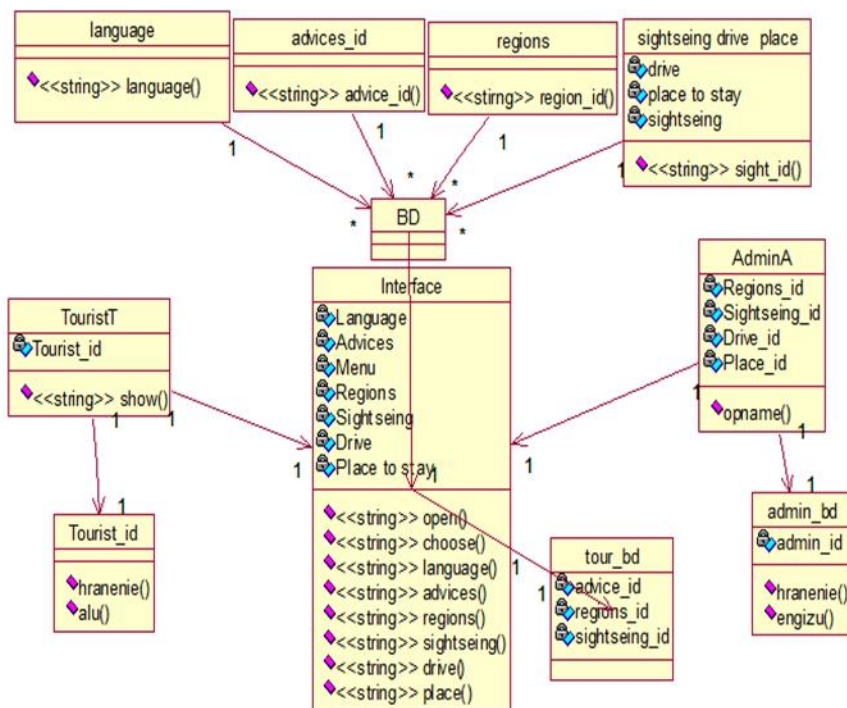


Рисунок 2.3- Взаимодействие с БД

При написании концептуальной модели, написаны атрибуты и сущности, из полученных данных была спроектирована логическая модель данных. База данных состоит из следующих таблиц:

- 1) список языков для пользователей;
- 2) данные о пользователях; - данные добавленные пользователем;
- 3) рекомендации;

- 4) список регионов;
- 5) связь между таблицами;
- 6) действия администратора в приложении;
- 7) категории пунктов меню;
- 8) фотографии всех категорий;
- 9) перечисление категорий;
- 10) указывает путь к выбранному месту.

База данных в диаграмме исходят ко всем элементам потока данных по запросу – SQL. При начале работы приложения основная задача лежит на базе данных, соответственно имеющий отношение к диску. При не выявлении базы данных на диске, то произойдет обновление после подключения к Интернету. Следовательно, пользователь имеет доступ к интересующей его категории.

После выбора категории в главном меню можно увидеть активность “Категория”, обращаясь к базе данных отображает их на экране. Выбрав категорию во вкладке меню отображается активность “Подкатегория”. Механизм передачи отвечает за передачу информации о категории между активностями, активность относится к базе данных и запросом выводит подкатегории. Выбирая любую категорию, соответственно предоставляется доступ на изменение базы данных. Из этого следует что база данных всегда будет иметь информацию о подкатегориях и в дальнейшем не придется обновлять информацию после каждого запуска. Выход из этих активностей осуществляется с помощью кнопки “Назад”. Активность «Табличный вид» выводит список всех объектов, при запросе с базы данных. При подключении к Интернету вычисляется местонахождение пользователя через Google Maps. Также в полях есть активность для бронирования авиабилетов на сайте Aviasales.ru, бронирование отеля на сайте Booking.com.

Активности представлены в виде TabView, в одном из которых находится карта, а в других – текстовые поля. Активность запрашивает из базы данных информацию об объекте и заполняет текстовые поля. Описание концепции разрабатываемого приложения приведен на рисунке.

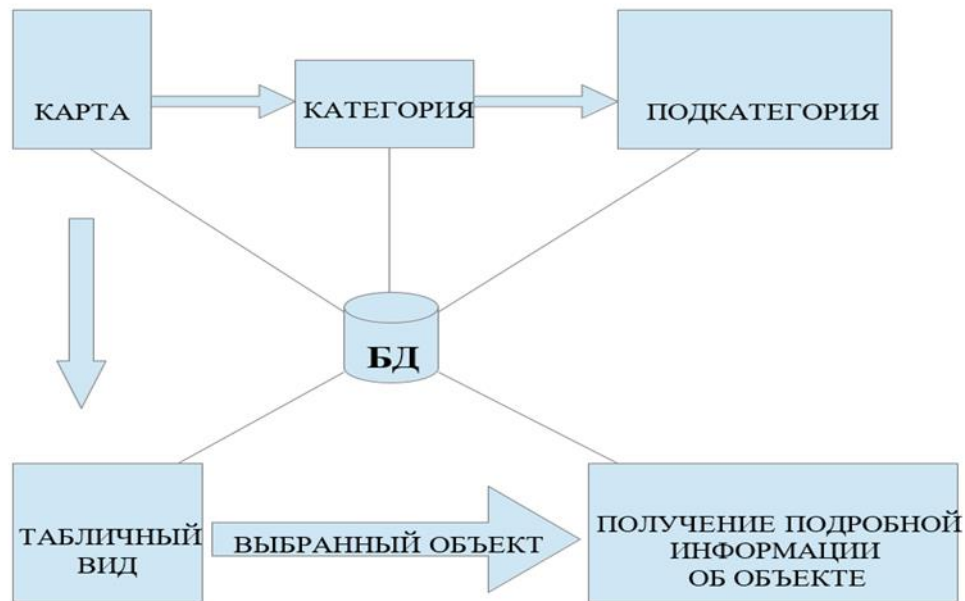


Рисунок 2.4 – Концепция работа БД приложения

2.4 Среда разработки FireBase

СУБД часть связывающая интерфейс с базой данных и пользователем или разработанным приложением, организовывая легкий доступ и обеспечивает данными. Разработчик считывает информацию, считывает, может обновлять и удалять с помощью – СУБД.

Firestore – это один из многофункциональных видов СУБД. Выполняет услуги бэкэнд (Baas), для разработки мобильных приложений. Он предоставляет множество функций, таких как аутентификация и безопасность, хранилище данных в реальном времени и хранилище файлов, аналитика, Push-уведомления, AdMod и многие другие.

SDK Firestore предоставляется для Android, iOS, Web, NodeJS, C ++ и Java Server. Основным сервисом является СУБД облачного класса NoSQL, благодаря которому разработчики могут синхронизировать и хранить данные между клиентами[5][6][7]. Приложения под операционные системы IOS Android интегрированы, для приложений JavaScript, Java, Objective-C и Node.js осуществлен API, включает работу напрямую с базами данных в виде REST из JavaScript-фреймворков, включая AngularJS, React, Vue.js, Ember.js и Backbone.js[8]. Рассмотрен API шифрования данных[9].

Компания выпустила веб-редактор кода Firepa с лицензией MIT. Веб-редактор обеспечивает совместную работа нескольким разработчикам с одним документом, в одно и то же время. Инновация стала основой редакторов Stash Realtime Editor фирмы Atlassian и Koding[11][12]. Ещё одно нововведенный

проект компании- бесплатный мессенджер, выпущенный с лицензией MIT – Firechat [13].

Firebase работает быстро в создании качественных приложений, заинтересованные пользователи повышают аудиторию и доходы. В платформу входят множество функций, серверный код для разработки мобильных сервисов, также функции для расширения аудитории.

Разработчик на Firebase связывает интуитивно все API, не добавляя нам лишней работы с другими пакетами. Выбрав то что нам нужно, используем преимущества решения интеграций. При использовании платформу с инфраструктурой Google, есть возможность обновления разработки на каждом этапе.

Главные предпочтения при работе с платформой FireBase, это:

- упрощенная работа и высокоскоростная разработка качественных приложений, благодаря связанному API, разнообразие инструментов;
- платформа имеет готовые инфраструктуры для работы;
- в основу платформы входит аналитический инструмент, разработан специально для смартфонов, позволяя получать данные о действиях пользователя;
- Firebase работает на платформах Android, iOS, JavaScript и C++, что продвигает платформу на уровень выше;
- при нагрузке на приложение, нет необходимости в замене сервера или подключать дополнительные ресурсы, Firebase решит все нагрузки;
- бесплатная связь по почте, разработчики Firebase ответят на все вопросы.

2.5 Язык программирования Java

Программы на Java транслируемые на переходят на Java байт-код, программа обрабатывает байтовый код и интерпретатором передает инструкции по оборудованию, все операции производятся на виртуальной машине Java (JVM).

Операционная система в данном способе байт- код не является зависим от операционной системы что и является преобладанием данного языка программирования. Java- приложения создаются на любом устройстве для соответствующего виртуального устройства. Одним из важных особенностей является гибкая система технологии Java, безопасность в системе полностью под контролем операционной системы. Совершая операции превышающие установленные разрешения, попытка несанкционированного доступа к данным, соединением операционной системы, приводит к немедленному прерыванию. К недостаткам относят понижение производительности виртуальной машины, к ряду модернизаций входит увеличение скорости выполнения программ на языке Java:

Сохранение первоначального кода на виртуальной машине, транслируя использование технологий байт кода во время разработки программы (JTI технология).

Программные средства, обеспечивают быструю обработку байт кода, например, поддерживаемая процессорами архитектурой ARM – технология Jazelle. По информации сайта shootout.alioth.debian.org, для семь разновидных задач выполнения программ на Java в полтора- два раза больше, чем на C\C++, в нескольких случаях Java быстрее, в отдельных в 7 раз медленнее [5].

Использование Java для большинство виртуальных машин в 15-30 больше чем использование программы на C\C++, исследование проведено компанией Google. Компания отметила невысокий уровень производительности и большое использование памяти на Java, сравнивая программами C\C++ [6][7][8].

Java часто используют для разработки мобильного приложения, под операционную систему Android. Код приложения компилируется в байт код, для использования выбранной разработчиком виртуальной машины. Во время компиляции так же используется инструмент, разработанный Google – Android SDK (Software Development Kit).

Разработать приложение можно в среде NetBeans, Android Studio, Eclipse, использовать дополнительные инструменты в Android Development Tools (ADT) или IntelliJ Idea, версия JDK должна быть 5.0 или выше. Компания Google 8 декабря 2014 года, официально приняли разработку Android Studio под ОС Android.

2.6 Операционная система Android

Среда разработки Android Studio является операционной системы для многих электронных приспособлений, такие как: смартфоны, планшеты, электронные книги, проигрыватели, игровые приставки, ноутбуки, смартбуки, телевизоров и других устройств. В 2015 году операционная система Android была создана для бытовых роботов и автомобильно-развлекательных систем. Имеет свою среду для разработки Java от Google и выполнен на ядре от Linux[7].

Основана на ядре Linux[7] и собственной реализации виртуальной машины Java от Google. Создание Open Handset Alliance (ОНА) было инициировано компанией Google, на данный момент производит поддержку и развитие платформы. Android позволяет разрабатывать Java-приложения, управлением которых занимается библиотека компании Google. Android Native Development Kit через этот инструмент можно отправить библиотеки и компоненты приложения на C\C++ и других языках.

По всему миру на 2014 год было продано 86 % смартфонов на операционной системе Android[8]. За весь период разработки смартфонов на

Android активировано 2 млрд устройств, объявили это разработчикам на конференции Google в 2017 году.

Приложение с операционной системой Android имеют нестандартный байт код для виртуальных машин Dalvik. Разрабатывая приложения есть доступ ко многим библиотекам, такие как:

- библиотека поддерживающая формат MPEG-4, H.264, MP3, AAC, AMR, JPEG и PNG – Packet Video OpenCORE;
- SGL обеспечивает приложение двухмерной графикой, обеспечение трехмерной графикой с OpenGL ES 1.0 ES 2.0;
- обеспечение доступом к к 2D/3D с помощью библиотека Surface Manager;
- WebKit предоставляет доступ к браузеру, обрабатывает JavaScript;
- FreeType обрабатывает шрифты;
- Firebase это СУБД доступная для всех разрабатываемых программ.

Сравнивая с другими приложениями на Linux, Android приложения имеют свои дополнительные правила [65]:

1. отравление данных между приложениями;
2. доступ к таким файлам как XML, PNG, JPEG;
3. мониторинг состояния строки;
4. управление приложениями.

Google рекомендует инструменты для скачивания для x86-машин с установленным Windows XP, Windows Vista, Windows 7, Linux, MacOS Software Development Kit. Для разработки предлагает более новый JDK 5. Для разработки приложения Android используется язык программирования Java. Также для удобства разработки Android- приложений используют плагин IntelliJIdea. В дополнении пользуются комплексной разработкой от Eclipse – Motodev Studio for Android. Дающий возможность работать напрямую с Google SDK.

В дополнении в 2009 году ADT опубликовал –Android Native Development Kit(NDK)[68] – набор библиотек и инструментов, с помощью которого реализуется отрезок приложения на C\C++. Для скорости разработки предлагают использовать в участках кода.

В 2013 году Google предложили среду разработки Android Studio на основе IntelliJ IDEA от компании JetBrains.

Google Android

Для разработки Android-приложения нет необходимости дополнительных устройств, кроме устройства Android, также можно подключиться к эмулятору. Все это реализовано Embarcadero RAD Studio – XE5 релиз которого состоялся в 2013 году.

В Android 4.4 разработан механизм для оптимизации памяти. Это виртуальная машина Dalvik Art (Android Runtime).

В Android 5 проработан дизайн на базе Material Design, также увеличен Project Volta - режим энергосбережения, выбор остановили на виртуальной машине ART.

В Android 6 добавили интеллектуальный расход энергии от Doze и доступ выхода в Интернет, использовать приложения на фоне App Standby.

Общая схема работы приложения Android

Android приложения при своей разработке применяет окна, точно так же как Windows, но в нашей программной системе окна носят название – Activity. Каждое окно содержит собственные особенности и жизненный цикл.

При работе с новым окном создается методом onCreate (), при запуске данного метода, приложение инициализируется и выдает его компоненты. При вызове следующих методов onStart () и onResume (), выходит окно отображения при его разработке, восстановлении и переключение с другого приложения. Методы сворачивания это onPause () и onStop (). Для закрытия окна используем метод onDestroy (), также в данном виде сохраняются пользовательские параметры и данные. Полную информацию по вызовам методов посмотреть можно на официальном сайте.

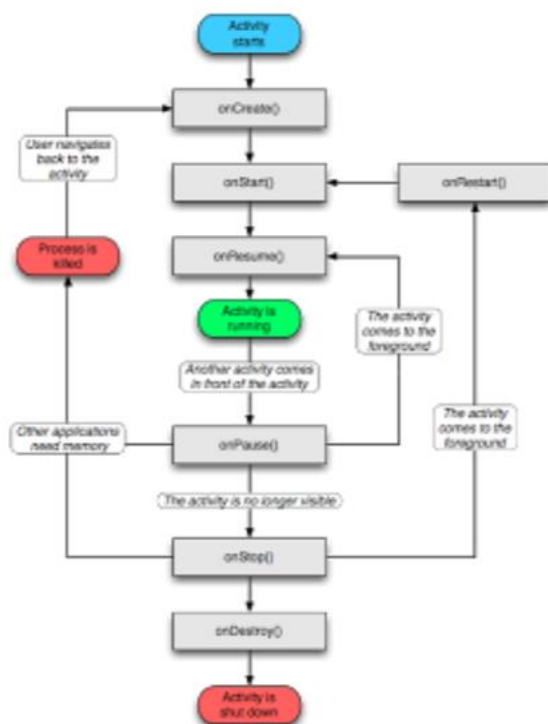


Рисунок 2.5 - Схема жизненного цикла приложения Android

3 Разработка программного обеспечения

Разработанная информационная система будет в виде приложения с оконным интерфейсом, вкладками внутри которых будет представляться нужная информация. Вкладки будут распределяться по категориям туристических мест, внутри будет информация по месту, с описывающими фотографиями. Основное направление приложения, это предоставление полной информации. Также в главном меню будет отображаться вкладка, пройдя по которой можно будет забронировать номер в отеле и приобрести авиабилеты. Основываясь на все удобства для горожан и гостей города, приложение будет разработана с целью повышения внутреннего туризма в Казахстане.

3.1 Установка Android Studio

Android Studio для Windows, Mac OS X и Linux предоставлен компанией Google, можно скачать с официального сайта компании. Перед загрузкой необходимо убедиться, соответствует ли компьютер и операционная система всем требованиям:

Для ОС Windows

- Microsoft Windows 7/8/10, 64-разрядная;
- рекомендуемая оперативная память от 6 ГБ до 8 ГБ;
- 2 ГБ свободного места для образа эмулятора;
- минимальное разрешение экрана с 1280 на 800 пикселей;
- JDK 8;
- для ускоренной работы эмулятора, 64- разрядная операционная система и процессор Intel, с функциями Intel VT и Intel EM64T.

Запустив **android-studio-bundle-143.2821654-windows.exe**, начнется процесс установки. Выводится диалоговое окно для дальнейшей установки **Android Studio**, показанное на рисунке 3.5:



Рисунок 3.1 - Настройка Android Studio

Для перехода в следующее окно установки пройдем по кнопке «Далее», где можно отказаться от установки установки **Android SDK** (входит в комплект установки) и **Android Virtual Device (AVD)**:

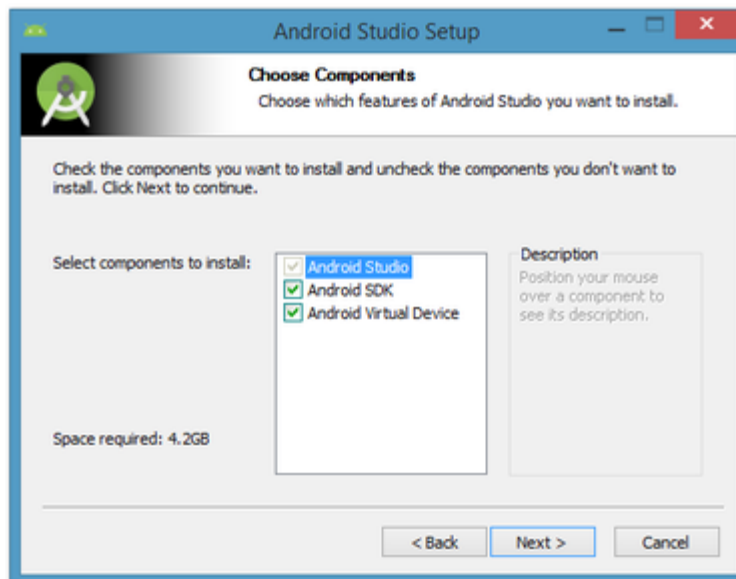


Рисунок 3.2 - Выбор перед установкой Android SDK и AVD?

Я сохранила настройки по умолчанию, после нажатия кнопки «Далее», выводится окно для лицензионного соглашения. Приняв все условия, продолжить установку.

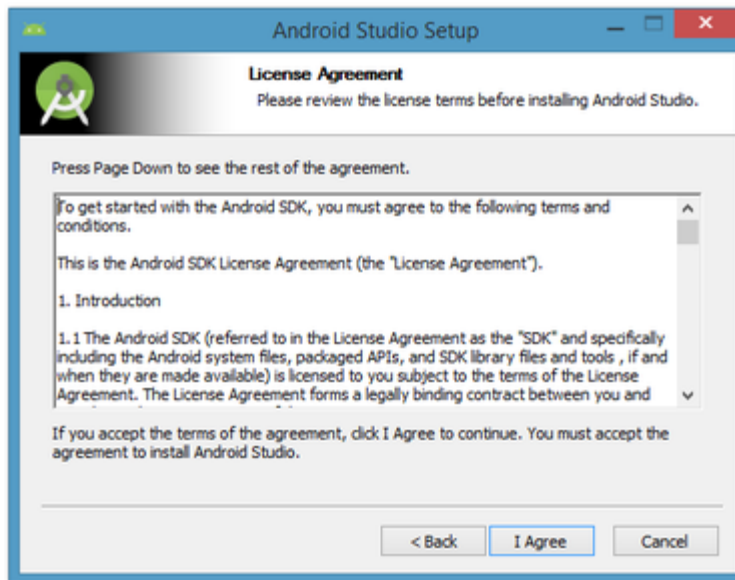


Рисунок 3.3 - Принятие условия лицензионного соглашения для продолжения установки.

В следующем диалоговом окне предлагается изменить папку установки **Android Studio** и **Android SDK**:

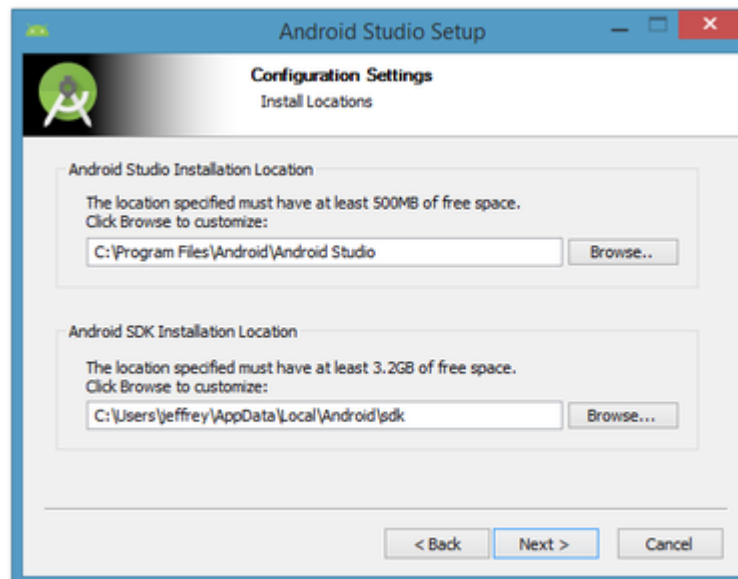


Рисунок 3.4 - Укажите место установки Android Studio и Android SDK

Выбрала настройки по умолчанию, нажав на кнопку «Далее» перехожу в следующее окно. Затем мы переходим на окно установки:

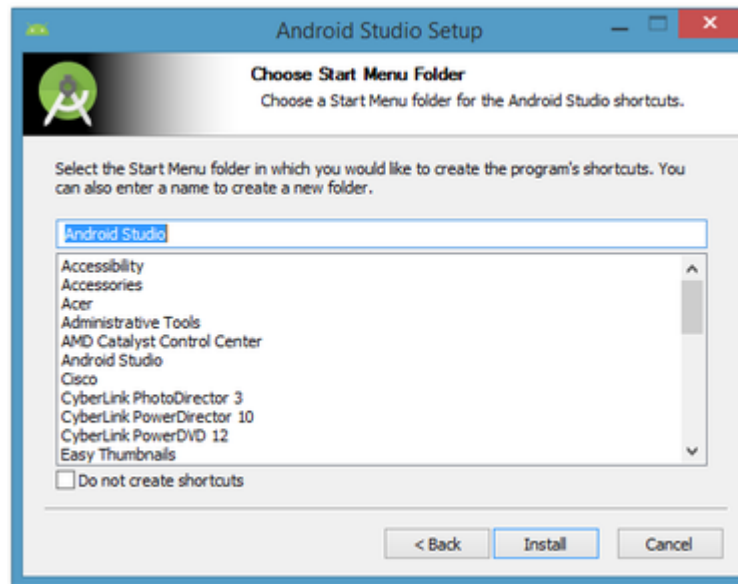


Рисунок 3.5 - Создайте ярлык для Android Studio

В ходе установки нам необходимо выбрать вид установки **Android Studio** и **Android SDK**. Далее в диалоговом окне выйдет сообщение и завершении установки.

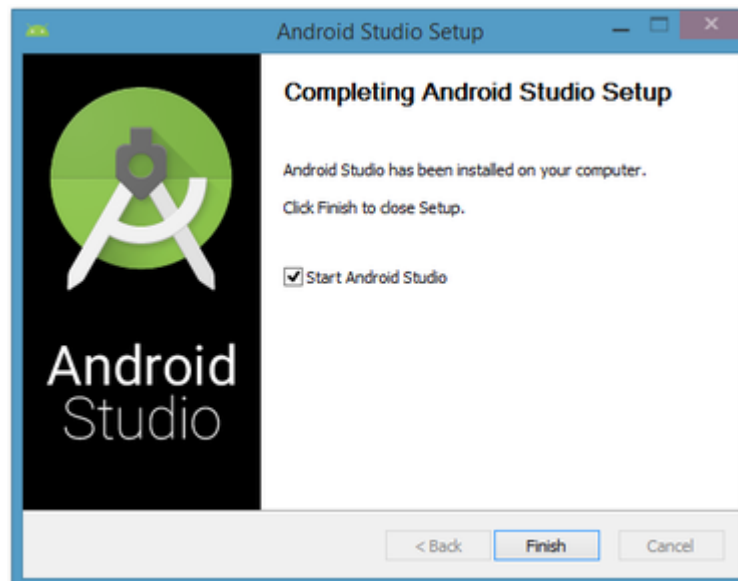


Рисунок 3.6 - Установите флажок «Запустить Android Studio»

Для завершения установки надо нажать на кнопку «Готово» и приступить к созданию мобильного приложения.

После первого запуска было установлено несколько дополнительных параметров для дальнейшей разработки. Приняты настройки по умолчанию, выбран тип **SDK**- стандартная конфигурация показано на рисунке 3.7.

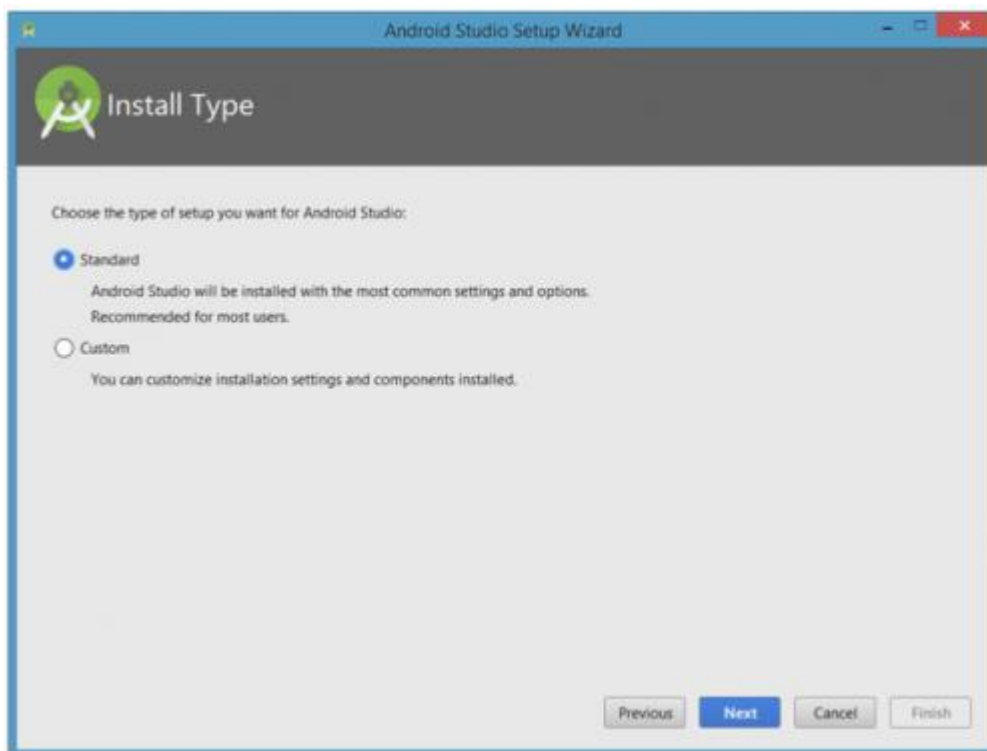


Рисунок 3.7 - Выберите тип установки

В каждом браузере есть папка, в которую по умолчанию загружаются все файлы, иногда необходимо изменить место хранения файлов. При нехватке оперативной памяти компьютера, решение проблемы и для ускорения работы – использовать эмулятор или **Android-устройство**.

Создание нового проекта

На данном этапе у нас запущен и настроен **Android Studio**, нажав на кнопку «Начать новый проект Android Studio» перед нами появится диалоговое окно «Создать новый проект». Далее нам будет предложено выбрать шаблон приложения, основное направление. Мной был выбран «**Empty Activity**» шаблон для разработки главного меню приложения. Показано на рисунке 3.8.

Пространство проекта состоит из меню, панели инструментов, компонентов, рабочей области и состояния строки.

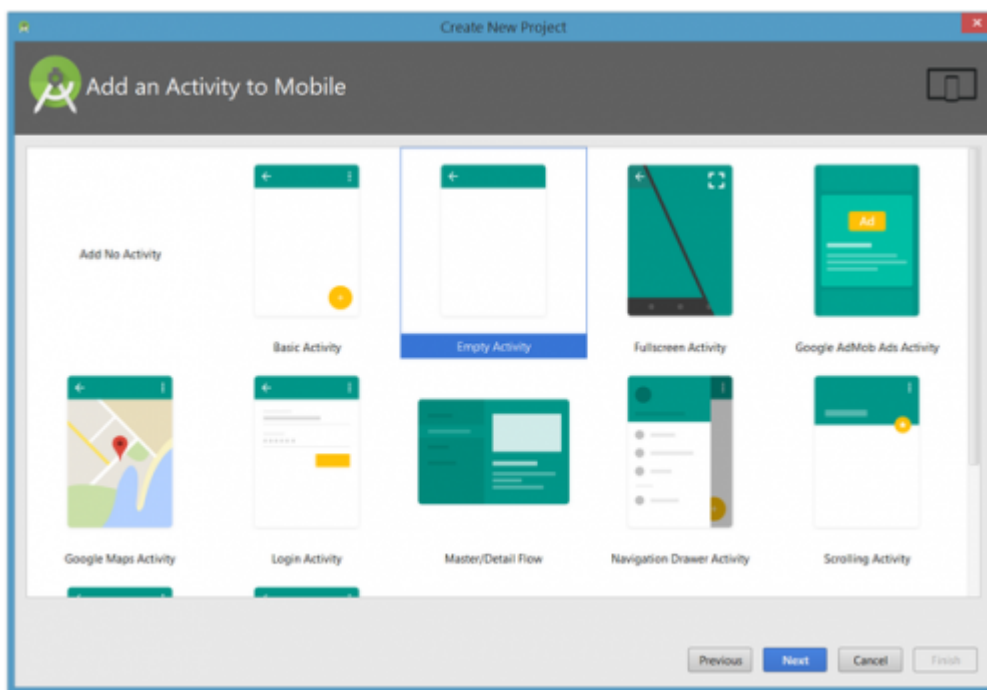


Рисунок 3.8 - Укажите шаблон направления

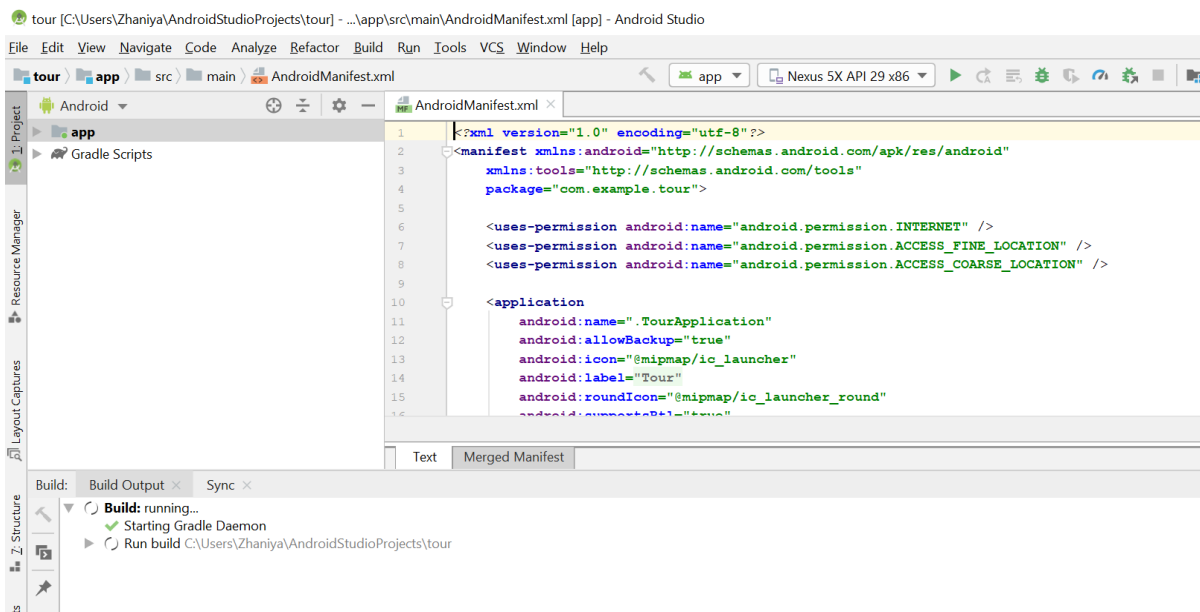


Рисунок 3.9 - Рабочее пространство на Android Studio

3.2 Интерфейс разработанного продукта

За последние годы технологии выдвинулись и вошли в нашу жизнь, что сейчас неотъемлемой частью путешествия являются смартфоны и планшеты. При помощи смартфона в один клик можно забронировать отель, купить авиабилеты, получить информацию о достопримечательностях и найти хорошее заведение. В наши дни в Play Market существует большое количество предложений для туристов с Android устройствами.

Разработав приложения со всеми удобствами для исследования города туристами, я надеюсь на рост внутреннего туризма в Казахстане. Приложение называется “Welcome to Kazakhstan”. Главный интерфейс программы представлен на рисунке 3.10.

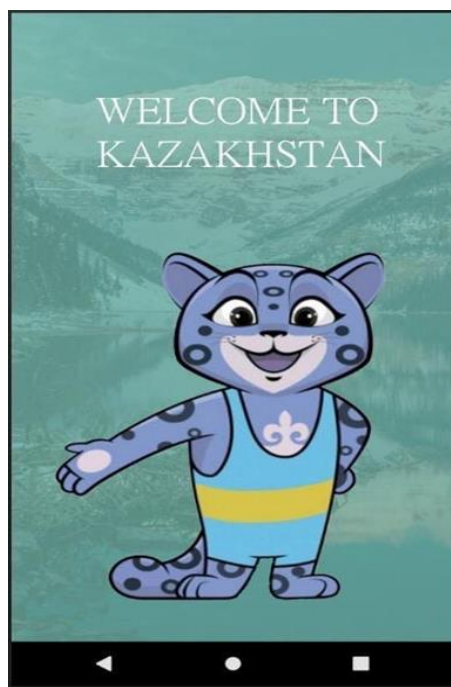


Рисунок 3.10 - Интерфейс “Welcome to Kazakhstan”

При входе в приложение нас встретит символ города Алматы, снежный барс, при клике переведется на следующую страницу приложения. После этого нам предстоит выбрать подходящий язык, в приложение загружены Русский и Английский (рисунок 3.10). Далее приложение будет готово к использованию.



Рисунок 3.11 - Выбор языка для пользователей

Перед нами главное окно программы, меню с вкладками, к каждой вкладке разработаны свои задачи (рисунок 3.11). Нажав на вкладку «Регионы», перед нами выйдет список загруженных городов, выбрать которые необходимо для дальнейшей работы. Ниже находится вкладку «факты», нажав на которую мы получим информацию о городе. Далее находятся вкладка “карта”, которая переведет пользователя на Google Maps и покажет местоположение объекта и пользователя. Есть возможность бронирование номера в отеле. При всем этом, прямо из приложения можно приобрести авиабилеты. Все действия по приложению будут описаны ниже по отдельности.

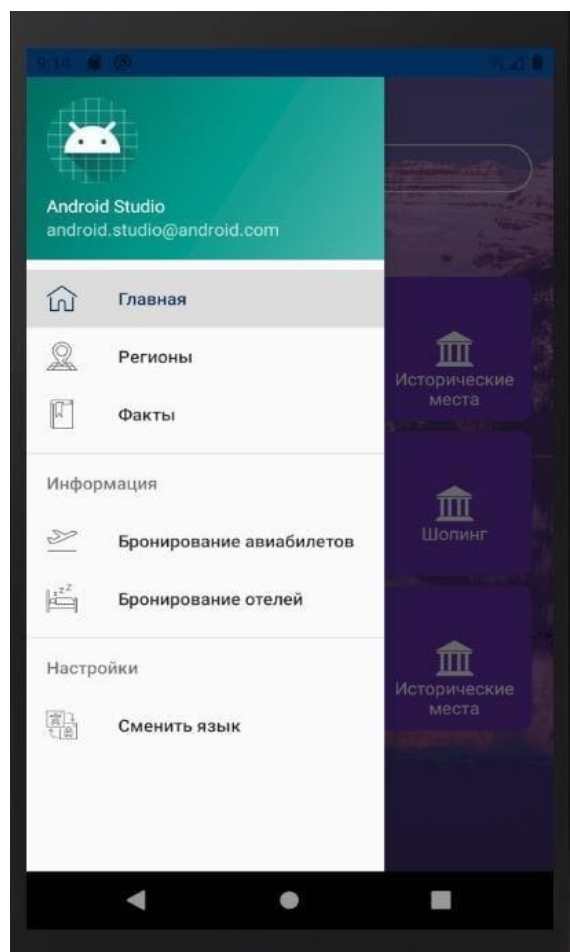


Рисунок 3.12 - Главная страница приложения с вкладками

🔍 Найдите лучшие предложения на проживание
697 000+ отелей, апартаментов, вилл и не только...

Место/название отеля:
например, город, регион, район или отель

Дата заезда День Месяц Дата отъезда День Месяц

Точные даты поездки пока неизвестны

Цель поездки: Работа Отдых

Гости

Рисунок 3.13 - Вкладка бронирование отеля

Booking.com – мобильная версия сайта по бронированию отелей (рисунок 3.13). Сайт имеет большую популярность среди туристов и не только. Через приложение можно легко забронировать номер в отеле выбранного города. В базе данных у приложения полмиллиона гостиниц по всему миру. На данный момент это один из самых больших ресурсов. Нет необходимости бродить по сайтам в поиске подходящей гостиницы, через сайт можно выбрать гостиницу за несколько минут.

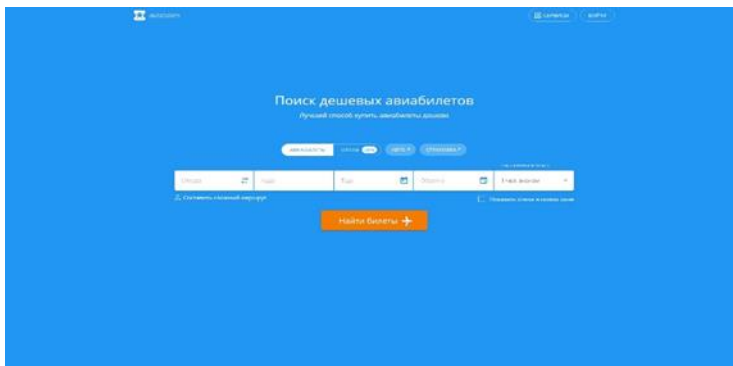


Рисунок 3.14 - Вкладка бронирование авиабилетов

Одно из популярных приложений для бронирования (рисунок 3.14) и покупки дешевых авиабилетов. Приложение направлено на максимальное удобство для пользователя, чтобы можно было купить билет по выгодней цене на ближайший рейс. В дополнении можно рассмотреть билеты на год вперед и выбрать с выгодной ценой.

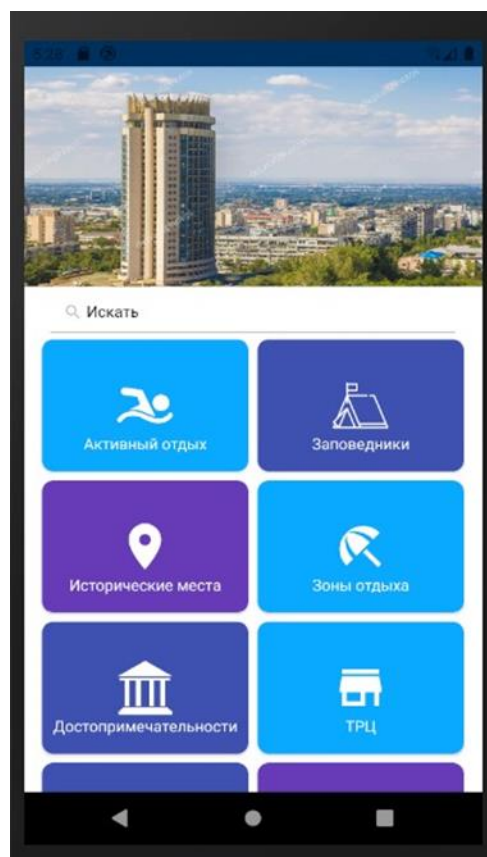


Рисунок 3.15 - Вкладка меню «Искать»

После главного меню, перед нами выйдет поле для поиска интересующих нас категорий. Позволяет поиск мест по запросу или по представленным разработчиком категориям. Категории доступны в следующих направлениях:

- активный отдых;
- заповедники;
- исторические места;
- торговые центры;
- зоны отдыха;
- достопримечательности.



Рисунок 3.16 - Вкладка «ТРЦ»

Для дальнейшей работа в приложении достаточно лишь кликнуть по картинке, затем она отображается на специальной панели. После каждого клика получаем информацию по месту.

В Android приложении мы использовали адаптер, для упрощения связанных с друг – другом элементов. В листинге ниже приведен отрывок кода, выполняющий фрагмент категории:

```

1 package com.example.tour.ui.category;
2
3 import ...
4
19
20 public class CategoryAdapter extends RecyclerView.Adapter<CategoryAdapter.ViewHolder> {
21
22     private ArrayList<CategoryItem> mValues;
23     private Context mContext;
24     protected ItemListener mListener;
25
26     private static int currentPosition = 0;
27
28     public CategoryAdapter(Context context, ArrayList<CategoryItem> values, ItemListener itemListener) {
29         mValues = values;
30         mContext = context;
31         mListener = itemListener;
32     }
33
34     public class ViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener {
35
36         private TextView textView;
37         private ImageView imageView;
38         private RelativeLayout relativeLayout;
39         private CategoryItem item;
40
41         ViewHolder(View v) {
42             super(v);

```

Рисунок 3.17 - Листинг категорий программы

Адаптеры подключены при работе с виджетами, которые дополняют android.widget.AdapterView: Spinner, GridView, ListView. У нас есть набор объектов и компонент View, назначение адаптера заключается в том, чтобы предоставить дочерние виды для компонентов.

```

92     intent.putExtra(name: "item_under10", "Aikai State Nature Reserve");
93     intent.putExtra(name: "item_under11", "Aksu-Zhabaglinsky reserve");
94     intent.putExtra(name: "item_under12", "City Nomad");
95     intent.putExtra(name: "item_under13", "Saki mounds");
96     intent.putExtra(name: "item_under14", "Tamgaly Tas");
97     intent.putExtra(name: "item_under15", "ESENTAI MALL");
98     intent.putExtra(name: "item_under16", "MEGA Alma-Ata");
99     intent.putExtra(name: "item_under17", "Aport mall");
100    intent.putExtra(name: "item_under18", "Colibri mall");
101    intent.putExtra(name: "item_under19", "Dostyk Plaza");
102    intent.putExtra(name: "item_under20", "CUS");
103    intent.putExtra(name: "item_under21", "Ramstore all-in");
104    intent.putExtra(name: "item_under22", "ADK");
105    intent.putExtra(name: "item_under23", "Globus");
106    mContext.startActivity(intent);
107
108
109
110
111    notifyDataSetChanged();
112 }
113
114 }
115
116 @Override
117 public int getItemCount() { return mValues.size(); }
120

```

Рисунок 3.18 - Подкатегории программы

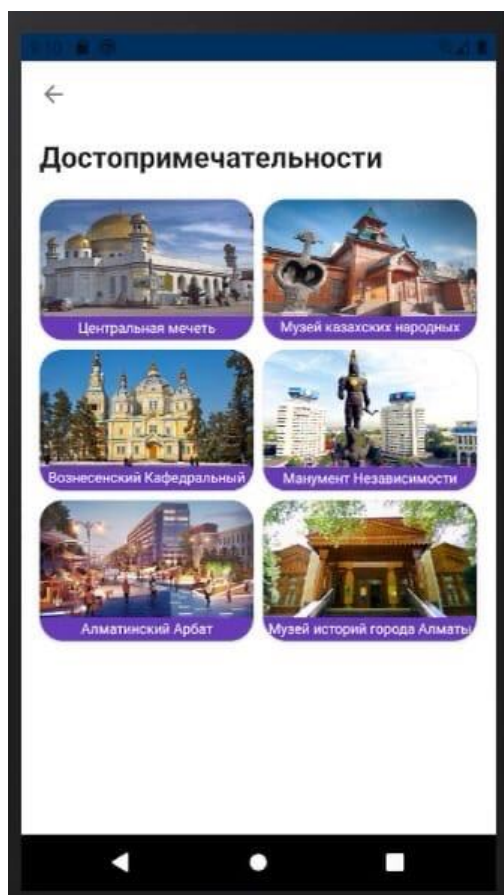


Рисунок 3.19 - Вкладка «Достопримечательности»

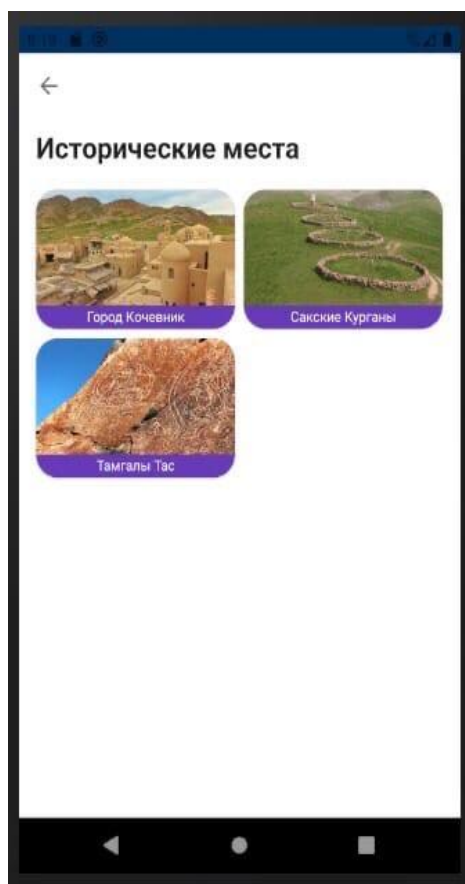


Рисунок 3.20 - Вкладка «Исторические места»

При выборе категории, будет доступна таблица подкатегорий с объектами, отсортированные по виду деятельности. Нажав на выбранное место перед нами выйдет информация по месту, адрес, телефон, часы работы и отзыв клиентов.



Рисунок 3.21 - Вкладка «Активный отдых»

При выборе подкатегории «Активный отдых», перед нами выйдет таблица с предлагаемыми нами места в городе Алматы. Если вы предпочитаете новые впечатления и эмоции, для компании друзей вкладка «Активный отдых» то что нужно вам. Мы предлагаем места в городе Алматы для ваших ярких и незабываемых выходных, провести время с пользой и получить адреналин. Вы сможете выбрать развлечение на свой вкус:

- побывать в батутном парке;
- посетить лыжный курорт Шымбулак;
- посетить высокогорный спортивный комплекс Медео;
- повидать один из символов г. Алматы парк “Кок-тобе”;
- для повышения адреналина, предлагаем выбрать профессиональный комплекс автотоспорта;
- воспользоваться сетью магазинов по аренде спортивного снаряжения и т.д.



Рисунок 3.22 - Информация по выбранному месту

Выбрав подкатегорию в меню выбора «Активный отдых», пользователь получить полную информацию по месту. Курорт уникален своим расположением - всего за 1 час вы можете добраться из центра города Алматы до его верхней точки, чтобы выпить кофе в самом высокогорном баре на территории СНГ. Здесь можно найти снег даже в 30-ти градусную июльскую жару. Летом средняя температура воздуха - +20 С, зимой около -7 С. И целых 300 солнечных дней в году! (рисунок 3.24).

В приложение Welcome to Kazakhstan можно загрузить информацию о других городах, вся информация будет храниться в одном месте и обновлять при необходимости. Имеет гибкую настройку, для дальнейшего развития приложения, для внешнего вида. В Play Market есть немало аналогов туристических приложений, но нет приложения - гид для Казахстана. Наше приложение ничем не отличается от аналогов, будет иметь больше положительных сторон в удобстве для пользователя. Кроме этого, удобство приложения в том, что, пользователь легко может забронировать номер в отеле и купить авиабилет. Приложение Welcome to Kazakhstan всегда пополняется

актуальной информацией и улучшается. При возникновении ошибок администратору отправляется отчет, и они работают над исправлением данной ошибки.

3.3 Тестирование

В процессе разработки приложения производилось поэтапное тестирование с целью выявления программных ошибок и несоответствий ТЗ (техническому заданию). Для этого нами были созданы эмуляторы смартфона и планшета с разными диагоналями экрана для разных версий Android. Тестируемый программный продукт последовательно запускался на этих эмуляторах, его поведение анализировалось, и при необходимости по результатам анализа вносились изменения в код [12].

1) Каждая активность была подвергнута юнит-тестированию с целью выявления ошибок, вызванных несоответствием ожидаемых и полученных параметров. Для этого для каждой активности был создан специальный юнит-класс, посылающий в активность различные верные и неверные параметры. При аномальном поведении активности или ее сбое, мною анализировалось поведение, и ошибка исправлялась.

2) в базу данных намеренно вносились недопустимые данные в соответствующие поля, которые могли быть неверно интерпретированы программой. Затем мною анализировалось поведение активности во время обработки недопустимых данных.

3) Приложение было запущено на устройствах, работающих под управлением разных версий Android с целью выявления особенностей работы приложения, запущенного в разных операционных системах.

4) После завершения цикла разработки, программный продукт тестировался на реальных устройствах. По результатам тестирования была добавлена виртуальная кнопка «Меню» для устройств, не имеющих аппаратных кнопок.

Возможные варианты развития программного приложения

Возможными путями развития приложения являются:

- интеграция приложения со сторонними базами данных;
- реализация двухкомпонентной базы данных (локальной базы данных и базы данных в облаке);
- реализация возможности подзагрузки пакетной информации;
- обновление мест интереса с использование удаленного сервера;
- реализация интерфейса на различных языках.

4 Технико- экономическое обоснование

Цель экономической части дипломной работы - при помощи всесторонней оценки, анализа и расчетов определить сложность разработки программного обеспечения, выполнить расчет затрат на разработку и оценить результат работы программного обеспечения.

В дипломном проекте я разрабатываю приложение –гид для туристов нашего города. В магазинах приложений для мобильных устройств, а именно: Google Play, Apple App Store.

4.1 Определение объема и трудоемкости разработки программного продукта

Таблица 4 – Распределение работ по этапам и видам и оценка их трудоемкости

№ п/п	Наименование работ	Длительность работы в марте 2020 года	
		в днях	в часах
1	Анализ и описание задач	4	5
2	Изучение и описание задачи	2	16
3	Основной этап разработки	14	112
4	Отладка программы	5	40
5	Ввод в действие	4	32
Итого		25	200

Таким образом, на разработку приложение было затрачено 25 дней в марте 2020 года, или 200 человеко-часа (при восьмичасовом рабочем дне).

4.2 Расчет затрат на разработку программного продукта

Расчет полных затрат на разработку проектного решения осуществляется по формуле:

$$C_{pi} = Z_{фот} + Z_{сзи} + M_i + P_{ci} + P_{mi} + P_{nki} + P_{zi} + P_{ni} \text{ где,} \quad (4.1)$$

$Z_{фот}$ - общий фонд оплаты труда разработчиков, тенге;

$Z_{сзи}$ - отчисления по социальному налогу, тенге;

M_i - затраты на материалы, тенге;

P_{ci} - затраты на специальные программные средства, необходимые для разработки проектного решения, тенге;

P_{mi} - затраты, связанные с эксплуатацией техники, тенге;

P_{nki} - затраты на научные командировки, тенге;

P_{zi} - прочие затраты, тенге;

P_{ni} - накладные расходы, тенге.

Размер фонда оплаты труда разработчика (Зфот) рассчитывается по формуле:

$$Зфот = Зoi + Зdi, \quad (4.2)$$

Где, Зoi - основная заработная плата, тенге;

Зdi - дополнительная заработная плата, тенге.

Время на разработку проекта составляет 25 дней. Месячная заработная плата руководителя – 220 000 тенге программиста – 160 000 тенге.

Заработная плата за один час рассчитаем по формуле:

$$D = \frac{ЗПм}{Др \times Чр}, \quad (4.3)$$

Где, ЗПм - ежемесячный размер заработной платы;

Др - количество рабочих дней в месяц (21 рабочих дня);

Чр - количество часов рабочего дня (при 8 часовом рабочем дне).

Заработная плата руководителя за один час составляет:

$$D = \frac{220000}{21 \times 8} = 1309 \frac{\text{тенге}}{\text{час}}$$

Заработная плата программиста за один час составляет:

$$D = \frac{160000}{21 \times 8} = 952 \frac{\text{тенге}}{\text{час}}$$

Таблица 4.1 - Затраты на оплату труда

Категория работника	Трудоемкость разработки ПП, чел.×час	Часовая ставка, тг/ч	Сумма, тг
Руководитель	45	1309	58 905
Программист	155	952	147 560
Итого			206 465

Дополнительная заработная плата составляет 10 % от основной заработной платы и рассчитывается по формуле:

$$Здоп = Зосн \times 10\%, \quad (4.4)$$

$$Здоп = 206\,465 \times 0,1 = 20\,646 \text{ тенге.}$$

Таким образом суммарный фонд оплаты труда составит ФОТ = 206 465 + 20 646 = 227 111 тенге.

Затраты на оплаты труда - 227 111 ₸

$$СО (\text{Социальные отчисления}) = 3.5\% \text{ от } (ЗП - ОПВ)$$

$$СО = (227\,111 - 22\,711,1) \times 0.035 = 7153,99 \text{ ₸}$$

$СН$ (Социальный налог) = 0.095% от (ЗП - ОПВ - ВОСМС) - СО
 $СН = (227\ 111 - 22\ 711,1 - 2271,11) \times 0.095 - 7153,99 = 12\ 048,24\ \text{т}$
 $ВОСМС$ (Отчисления на ОСМС) = 2% от (227 111) = 4 542,22 т

Таблица 4.2 - Общеустановленный режим (ОУР)

Уплаченные юридическим лицом налоги	Ставка налога	ФОТ	161 145
СО (Социальные отчисления)	3,5	(ЗП - ОПВ)*3,5%	7153,99
ВОСМСЮ (Отчисления на ВОСМСЮ)	2,0	ЗП*2%	4 542,22
СН (Социальный налог)	9,5	(ЗП - ОПВ - ВОСМС)*9,5% - СО	12 048,24
Итого			23 744,45

Определяем общую сумму затрат на различные материальные ресурсы (З_м), по формуле, которая представлена ниже:

$$Z_M = \sum_{i=1}^n P_i * C_i \quad (4.5)$$

Где, P_i - расход i -го вида материального ресурса, натуральные единицы;
 C_i - цена за единицу i -го вида материального ресурса, тг;
 i - вид материального ресурса;
 n - количество видов материальных ресурсов.

Таблица 4.3 - Затраты на материальные ресурсы

Наименование материального ресурса	Единица измерения	Количество израсходованного материала	Цена за единицу, тг.	Сумма, тг.
Блокнот(А4)	Штук	2	1300	2600
Карандаш	Штук	2	120	240
Ручки	Штук	2	120	240
Итого				3080

Таблица 4.4 - Расчет затрат на оборудование и ПО, необходимое для проекта

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Ноутбук	Lenovo Idea Pad 330	Штук	1	164 389	164 389
Компьютерная мышь беспроводная	Logitech M171	Штук	1	5300	5300
Подписка Play Market			1	10 668	10668
Итого					180 357

$$Z_{\text{общее}} = 164\,389 + 5300 + 10668 = 180\,357 \text{ тг.}$$

Общая сумма затрат на электроэнергию ($Z_э$) рассчитывается по формуле:

$$Z_э = \sum_{i=1}^n M_i * K_i * T_i * Ц, \quad (4.6)$$

Где, M_i - паспортная мощность i -го электрооборудования, кВт;

K_i - коэффициент использования мощности i -го электрооборудования (принимается $K_i=0.7, 0.9$);

T_i - время работы i -го оборудования за весь период разработки ПП ч;

$Ц$ - цена электроэнергии, тг/кВт×ч;

i - вид электрооборудования;

n - количество электрооборудования.

$$Z_{э \text{ общие}} = (0,6 \times 0,7 \times 200 \times 19,17) + (0,5 \times 0,7 \times 6 \times 19,17) = 1650,53 \text{ тенге}$$

Расходы по статье «Прочие затраты» на конкретное ПО включают затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы по формуле:

$$Pz_i = Z_{oi} \times Nпз / 100, \quad (4.7)$$

Где, $Nпз$ – норматив прочих затрат в целом по организации в (%), в дипломной работе нужно брать 20%.

$$Pz_i = 146496 \times 20 / 100 = 29\,299$$

Таблица 4.5 - Смета затрат на разработку программы

Статьи затрат	Сумма, тг
Зфот – общий фонд оплаты труда разработчиков, тенге;	227 111
Зсзi- отчисления по социальному налогу, тенге;	23 744
Мi- затраты на материалы, тенге;	3080
Рсi- затраты на специальные программные средства, необходимые для разработки проектного решения, тенге;	180 357
Зэ-затраты на электроэнергию	1650,53
Пзi-прочие затраты, тенге;	29 990
Итого	465 932

4.3 Сравнительный анализ эксплуатационных затрат до и после внедрения

Статьи затрат после внедрения программного обеспечения:

- основная и дополнительная заработная плата программиста с отчисляемыми налогами;
- износ ПЭВМ;
- накладные расходы.

Статьи затрат до внедрения:

- основная и дополнительная заработная плата программиста;
- расходуемые материалы;
- накладные расходы.

Таблица 4.6 - Расчет затрат после внедрения ПО

Сотрудник	Кол-во, чел	Заработная плата в месяц, тг	Заработная плата в год, тг	Социальное отчисление	Социальный налог	ВОСМС
Программист	1	160 000	1 920 000	60 480	102 038	38 400
Итого	1	160 000	1 920 000	60 480	102 038	38 400
К оплате						2 120 918

Износ ПЭВМ рассчитывается исходя из 25 % амортизационных отчислений за год. Занесем данные в таблицу 4.7

Таблица 4.7 - Амортизационные отчисления

Оборудование	Количество	Стоимость, тенге	Отчисления
Lenovo Idea Pad 330	1	164 389	41 097
Компьютерная мышь беспроводная	1	5300	1325
Итого	2	169 689	42 422

Общие накладные расходы рассчитываются по формуле:

$$З_{н.р.} = 2\,120\,918 * 0,2 = 424\,183 \text{ тенге,} \quad (4.8)$$

Итого, общие эксплуатационные расходы после внедрения программного продукта будут равны,

$$З = 2\,120\,918 + 42\,422 + 424\,183 = 2\,587\,523 \text{ тенге.}$$

Таблица 4.8 - Затраты на заработную плату

Сотрудник	Кол-во, чел	Заработная плата в месяц одного, тг	Заработная плата в год, тг	Социальное отчисление	Социальный налог	ВОСМС
Руководитель	1	220 000	2 640 000	83 160	140 302	52 800
Программист	1	160 000	1 920 000	60 480	102 038	38 400
Итого	2	380 000	4 560 000	143 640	242 340	91 200
К оплате						5 037 180

Износ ПЭВМ рассчитывается исходя из 25 % амортизационных отчислений за год. Занесем данные в таблицу 4.9

Таблица 4.9 - Амортизационные отчисления

Оборудование	Количество	Стоимость, тенге	Отчисления
Lenovo Idea Pad 330	1	164 389	41 097
Компьютерная мышь беспроводная	1	5300	1325
Итого	2	169 689	42 422

Таблица 4.10 - Затраты на расходуемые материалы

Наименование материального ресурса	Единица измерения	Количество израсходованного материала	Цена за единицу, тг
Блокнот(А4)	Штук	2	1300
Карандаш	Штук	2	120
Ручки	Штук	2	120
Итого			3080

Общие накладные расходы рассчитываются по формуле (4.9)

$$Z_{н.р.} = 5\,037\,180 * 0,2 = 1\,007\,436 \text{тенге,} \quad (4.9)$$

Итого, общие эксплуатационные расходы до внедрения программного продукта будут равны (4.10)

$$Z = 4\,639\,110 + 42\,422 + 3080 + 1\,007\,436 = 5\,692\,048 \text{тенге,} \quad (4.10)$$

Для наглядности сведем все данные в аналитическую таблицу 4.11

Таблица 4.11 - Годовые эксплуатационные затраты

Статьи	До внедрения ПП	После внедрения ПП
Годовая заработная плата	5 037 180	2 120 918
Расходуемые материалы	3080	-
Износ ПЭВМ	42 422	42 422
Накладные расходы	1 007 436	465 932
Итого	5 692 048	2 629 272

Годовая экономия денежных средств будет составлять:

$$5\,692\,048 - 2\,629\,272 = 3\,062\,776 \text{тенге,} \quad (4.11)$$

Необходимо определить экономический эффект от внедрения программного продукта.

Технико-экономические показатели позволяют определить целесообразность проведения разработки и ее внедрения, а также оценить реальную выгоду, как для разработчика системы, так и для ее пользователя.

Необходимо определить срок окупаемости программного продукта в месяцах по формуле:

$$P_{\text{окуп}} = 12 * \frac{Z_{\text{разработка}}}{\Delta_{\text{годовая}}}, \quad (4.12)$$

Где, $P_{\text{окуп}}$ – период окупаемости в месяцах;

$Z_{\text{разр.}}$ – затраты на разработку программы;

$\Delta_{\text{годовая}}$ – годовая экономия.

$$P_{\text{окуп}} = 12 \cdot \frac{465\,932}{3\,062\,776} = 12 \cdot 0,152 = 1,8 \text{ мес.}$$

Зная годовую экономию и годовые эксплуатационные затраты с внедрением программного продукта можно определить коэффициент эффективности ($K_{\text{эф.}}$) по формуле.

Расчетный коэффициент экономической эффективности капитальных вложений составляет:

$$E_p = \frac{\Delta_{\text{уг}}}{K}, \quad (4.13)$$

$$E_p = \frac{3\,062\,776}{465\,932} * 100\% = 657\% \text{ или } 6,54 \text{ тенге.}$$

4.4 Вывод по экономической части

Коэффициент эффективности показывает сколько экономии в тенге приходится на каждый тенге затрат. Следовательно, на каждый тенге затрат приходится 6,54 тенге экономии.

Сравнивая данные показатели видно, что внедрение программного продукта отлично отразится на прибыльности предприятия, поскольку коэффициент экономической эффективности превышает нормативной, поэтому разработка продукта является эффективной.

5 Безопасность жизнедеятельности

В данном дипломном проекте описывается разработка информационной системы для туристов города Алматы. Приложение разрабатывается специально для города Алматы в целях развития внутреннего туризма. В ходе разработки информационной системы необходимо обеспечить безопасность разработчикам и был произведен анализ потенциально опасных и вредных факторов в офисе, а также выполнены расчеты обеспечивающие комфортные условия труда.

Данный раздел дипломного проекта рассматривает следующие вопросы:

- оптимальные условия труда инженера-программиста;
- расчет искусственного освещения;
- расчет уровня шума.

5.1 Анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал

На оператора в процессе его рабочей деятельности воздействуют опасные (вызывающие травмы) и вредные (вызывающие заболевания) производственные факторы.

Вредный производственный фактор - производственный фактор, работа в определенных условиях приводящая к снижению работоспособности, а также приводит к заболеваниям.

Опасный производственный фактор - производственный фактор, воздействие на работающего, в определённых условиях, приводит к травме или другому внезапному ухудшению здоровья.

Вредный производственный фактор, в зависимости от интенсивности и длительности воздействия, может стать опасным. При работе с компьютером человек подвергается к воздействию опасных, так и вредных производственных факторов, в связи с этим работа с использованием видео дисплея (ВДТ) учитывают возможное негативное влияние этих факторов на -человеческий-организм.

Опасные и вредные производственные факторы по влиянию на организм человека подразделяются на следующие группы:

- физические;
- психофизиологические;
- химические;
- биологические.

При работе с компьютером инженер-программист подвергается воздействию опасных и вредных производственных факторов первых двух групп.

К физическим факторам, которые могут воздействовать на оператора,

относятся:

- опасный уровень напряжения в электрической цепи, замыкание которой может произойти через тело человека;
- повышенный уровень статического электричества
- повышенный уровень электромагнитных излучений (ВЧ, УВЧ, СВЧ);
- повышенный уровень ионизирующих излучений;
- недостаточная освещенность рабочей зоны;
- повышенная пульсация светового потока;
- повышенный уровень ультрафиолетовой, инфракрасной и ионизирующей радиации;
- шум и вибрация.

К психофизиологическим факторам относятся:

- физические перегрузки (статические, динамические);
- нервно-психические перегрузки, обусловленные умственным перенапряжением, перенапряжением анализаторов, монотонностью труда.

Анализ возможных опасных и вредных производственных факторов, воздействующих на оператора, делает необходимым более детальное их изучение с целью разработки мероприятий по устранению или уменьшению последствий негативного влияния на организм человека, а также разработки требований к организации безопасной работы оператора ЭВМ.

5.1.1 Характеристика условий труда и рабочего места инженера-программиста ЭВМ

Производственное помещение состоит из 1 комнаты, оборудована 2-мя рабочими местами. Параметры микроклимата производственного помещения описывается следующим образом: температура воздуха на рабочем месте составляет 20°C, относительная влажность - 52%. Освещенность комнат соответствует 180 лк, точность зрительных работ - высокая. Уровень шума – 49.5 дБ. Длительность сосредоточенного наблюдения состоит из 20% от продолжительности рабочего времени, продолжительность повторяющихся операций - 56 сек., выявлено 4 важных видов наблюдения.

5.2 Расчет интегральной оценки условий труда и определение тяжести труда

Определить категорию тяжести труда инженера-программиста. Для этого необходимо присвоить каждому соответствующий балл характеризующий каждый фактор труда. Температура на рабочем месте рассчитан на 1 балл, относительная влажность - 1, освещенность - 3, точность зрительных работы - 4, уровень шума - 2, процент зрительных работ- 1, длительность повторяющихся

операций - 2, количество важных объектов наблюдения -1. Далее надо рассмотреть факторов из 8 баллы которых превышают 1, выявили 4 ряда факторов: точность зрительных работ, освещение, шум, продолжительность повторяющихся действий. Расчет интегральной оценки тяжести труда необходимо предварительно выявить среднее значение баллов X, средний балл не включается в расчет высшего балла, присвоенного неблагоприятному фактору по формуле:

$$X = \frac{2+3+2}{3} = 2.33 \quad (5.2)$$

Произведем расчет интегральной оценки тяжести труда:

$$I_T = 10 \left(4 + 2.33 \times \frac{6-4}{6} \right) = 47.68 \quad (5.3)$$

Интегральная оценка тяжести труда составляет 47,68 что соответствует IV категории тяжести труда.

Расчет показателей утомляемости и работоспособности работников отдела
Показатель утомляемости Y в условных единицах составляет:

$$Y = \frac{47,68-15,6}{0,64} = 50,12 \quad (5.4)$$

Соответственно показатель работоспособности R равен:

$$R = 100 - 50,12 = 49,88 \quad (5.5)$$

Основываясь на проведенный нами анализ, были выявлены факторы которые надо изменить для улучшений условий труда, необходимо изменить температуру помещения и точность зрительных работ.

5.3 Расчет искусственного освещения

Расчет освещения постоянно рабочего места приводит к выбору систем освещения, необходимого определить количество светильников, их вид и место размещения. Опираясь на вышеозначенное, рассчитаем объем искусственного освещения.

Искусственное освещение выполняется двумя видами электрических источников света: накаливания ламп и люминесцентных ламп. В расчете мы используем люминесцентные лампы, которые по сравнению с лампами накаливания имеют значительные преимущества [24]:

- по световому составу они близки к естественному, дневному свету;
- имеют высокий КПД (сравнивая выделяют 1,5-2 больше света чем лампы накаливания);

- имеют высокую светоотдачу (в 3-4 раза выше чем у накапливающих ламп);
- длительный срок службы;
- отсутствует ослепляющий эффект, свет мягкий и не давящее на глаза.

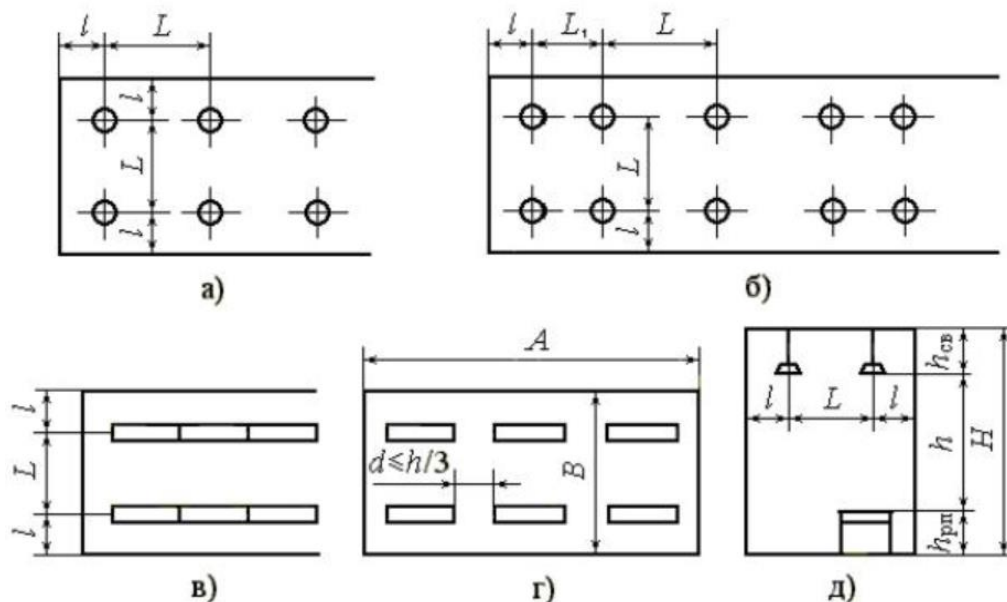


Рисунок 5.3.1 - Размещение светильников общего освещения в помещении

Расчет произведем для комнаты площадью 18 м², ширина – 6 м, высота – 3м. Будем использовать метод световых потоков. Определить число светильников и падающий световой поток по формуле :

$$F = \frac{E \cdot K \cdot S \cdot Z}{n} \quad (5.6)$$

где, F-расчет светового потока, Лм;

E-нормируемая минимальная освещенность, Лк (определить по таблице). Работа программиста, соответствуя с таблицей, относиться к точным работам, соответственно мин. освещение будет E = 300 Лк;

S-площадь освещения (в нашем случае S=18 м²);

Z-средняя освещенность к минимальной (принимается равным Z=1,1);

K-коэффициент запаса, учитывает снижение светового света при загрязнении светильника в процессе (значение зависит от вида помещения и производимых работ, в нашем случае K=1,5);

n = обозначает использовавшийся коэффициент (выводится между световым потоком и поверхность на которую падает свет, к числу потоков всех ламп и определяется в долях единицы; зависит от характеристик ламп, объемом

помещения, окраски стен и потолка, характеризуется коэф. отражения от стен (P_c) и потолка (P_n)), выше были указаны: $P_c=40\%$, $P_n=60\%$. Значение n определить по таблице коэф. использования светильников.

Вычислить индекс помещения по формуле:

$$I = \frac{S}{h(A+B)} \quad (5.7)$$

Где, S - площадь помещения, $S=18\text{м}^2$;

h - высота потолка, $h=2,92$ м;

A - ширина помещения, $A=3$ м;

B - длина помещения, $B=6$ м;

Подставив значение:

$$I = \frac{18}{2.92*(6+3)} = 0.68.$$

Учитывая индекс помещения I , в таблице находим $n=0,22$. Использовать все значения в формуле для расчета потока света F :

$$F = \frac{300*1.5*18*1.1}{0.22} = 40500\text{Лм}. \quad (5.8)$$

Выбираем для освещения люминесцентные лампы ЛБ 40-1 поток которых равен $F=4320\text{Лм}$. Рассчитаем по формуле:

$$N = \frac{F}{F_{\text{л}}}; \quad (5.9)$$

N -число необходимых ламп;

F - световой поток, $F_{\text{л}}$ -световой поток лампы.

$$N = \frac{40\ 500}{4320} = 9\text{шт.}$$

Всего для помещения нормируемой освещением 400 лк потребуется 9 люминесцентных ламп. При выборе светильных приборов надо использовать светильники типа ОД. В каждый светильный прибор входит две лампы.

5.4 Расчет уровня шума

Шум сказывается на качестве работы оказывая на человека вредные воздействия. Сотрудники, работающие под высоким и длительным шумом, испытывают раздражение, ослабление памяти, усиление утомляемости и боли в ушах. Такие признаки в системе организма человека сказывается на эмоциональном состоянии и приводит к нервозности. В результате повышенного шума можно отметить снижения концентрации внимания, повышение нервно-психического и состояния усталости, снижение речевой коммуникации. Все это приводит к понижению работоспособности инженера-программиста.

Продолжительное влияние высокого шума [выше 80дБ(А)] на слух приводит к его частичной потере [25].

Уровень шума на месте работы инженера-программиста не должен превышать 50 дБА, а в помещении обработки вычислительными машинами - 65дБА. Чтобы снизить уровень шума потолок и стены помещения, где стоят компьютеры, надо облицевать звукопоглощающими материалами.

В установленных помещениях вычислительной техники можно снизить установки специальных виброизоляторов.

Одним из неблагоприятных факторов производственной среды в ИВЦ является высокий уровень шума, создаваемый печатными устройствами, оборудованием для кондиционирования воздуха, вентиляторами систем охлаждения в самих ЭВМ.

Для решения вопросов о необходимости и целесообразности снижения шума необходимо знать уровни шума на рабочем месте оператора.

Уровень шума, возникающий от нескольких некогерентных источников, работающих одновременно, подсчитывается на основании принципа энергетического суммирования излучений отдельных источников [25]:

$$L_{\Sigma}=10 \lg \sum_{i=1}^{i=n} 10^{0.1L_i}, \quad (5.10)$$

Где, L_i – уровень звукового давления i -го источника шума;

n – количество источников шума.

Полученные результаты расчета сравнивается с допустимым значением уровня шума для данного рабочего места. Если результаты расчета выше допустимого значения уровня шума, то необходимы специальные меры по снижению шума. К ним относятся: облицовка стен и потолка зала звукопоглощающими материалами, снижение шума в источнике, правильная планировка оборудования и рациональная организация рабочего места оператора.

Уровни звукового давления источников шума, действующих на оператора на его рабочем месте представлены в таблице 5.

Таблица 5 - Уровень звукового давления различных источников

Источник шума	Уровень шума , дБ
Жесткий диск	40
Вентилятор	45
Монитор	17
Клавиатура	10
Принтер	45
Сканер	42

Обычно рабочее место оператора оснащено следующим оборудованием: винчестер в системном блоке, вентилятор(ы) систем охлаждения ПК, монитор, клавиатура, принтер и сканер.

Подставив значения уровня звукового давления для каждого вида оборудования в формулу (5.11), получим:

$$L_{\Sigma} = 10 \cdot \lg (10^4 + 10^{4,5} + 10^{1,7} + 10^1 + 10^{4,5} + 10^{4,2}) = 49,5 \text{ дБ} \quad (5.11)$$

Полученное значение не превышает допустимый уровень шума для рабочего места оператора, равный 65 дБ (ГОСТ 12.1.003-83). И если учесть, что вряд ли такие периферийные устройства как сканер и принтер будут использоваться одновременно, то эта цифра будет еще ниже. Кроме того, при работе принтера непосредственное присутствие оператора необязательно, т.к. принтер снабжен механизмом автоподачи листов.

5.5 Вывод по разделу безопасность жизнедеятельности

В результате сравнения ламп накаливания и люминесцентных ламп по техническим характеристикам соответствуют люминесцентные лампы. В расчете мы используем люминесцентные лампы, которые по сравнению с лампами накаливания имеют значительные преимущества. Вычислили индекс помещения по формуле и выявили нормируемое освещение для офисного помещения. Если результаты расчета выше допустимого значения уровня шума, то необходимы специальные меры по снижению шума. К ним относятся: облицовка стен и потолка зала звукопоглощающими материалами, снижение шума в источнике, правильная планировка оборудования и рациональная организация рабочего места оператора.

Заключение

В результате выполнения дипломного проекта была разработана информационная система для горожан и туристов города Алматы. Следовательно, была достигнута поставленная цель. Данная система будет служить информационным справочником для путешествий, предоставлять достоверную информацию, помогать туристам построить свой маршрут.

Были реализованы такие задачи как:

- изучение внутреннего туризма на территории РК;
- проведен подробный анализ существующих информационных систем для туристов города;
- определена структура данной программы;
- разработан удобный и понятный интерфейс;
- дана оценка экономической эффективности внедрения системы.

Успешность туристической компании, в современном мире, зависит от грамотно реализованной информационной системы. Работа была выполнена с использованием новых и актуальных технологий разработки приложений, как Java, Android Studio. Разработанная программа устойчиво выполняет все свои функции.

Таким образом, изучив развитие внутреннего туризма и проведя сравнительный анализ, было разработано многофункциональное приложение на операционной системе Android Studio и готовая к использованию:

- повысить уровень туризма;
- получить достоверную информацию
- бронировать номера в отеле;
- покупка авиабилетов;
- узнать о новых местах в городе.

Разработанная информационная система может быть внедрена в специальные туристические направления. Кроме того, программа пригодна для горожан, удобна в использовании не только для путешественников.

Список литературы

1 Об отраслевой программе развития перспективных направлений туристской индустрии Республики Казахстан на 2010 - 2014 годы // Казахстанская правда. - 2010. - 3 ноября.

2 Никитинский Е.С., Вуколов В.Н. Формирование индустрии туризма в Казахстане за годы государственной независимости // Агентство Республики Казахстан по туризму и спорту, Казахская академия спорта и туризма. Материалы Международной научно-практической конференции. - Астана, 2006.

3 stat.kz/DIGITAL/TURIZM/Pages/default.aspx — Официальный сайт агентства Республики Казахстан по статистике (раздел туризма и культуры).

4 What is Android? (англ.). Android Developers. Дата обращения 6 марта 2010. Архивировано 1 марта 2012 года.

5 Android Captures Record 85% Share of Global Smartphone Shipments in Q2 2014 (англ.), <http://www.strategyanalytics.com> (30 July 2014). Дата обращения 3 августа 2014.

6 Java 6 -server speed ÷ C++ GNU g++ speed | Computer Language Benchmarks Game (недоступная ссылка). Дата обращения 4 марта 2010. Архивировано 14 июня 2011 года.

7 Metz, Cade. Google pits C++ against Java, Scala, and Go (англ.). The Register (3 June 2011). Дата обращения 5 июня 2011. Архивировано 21 августа 2011 года.

8 Loop Recognition in C++/Java/Go/Scala (PDF, 0.3 МБ)

9 В Google провели сравнение производительности C++, Java, Go и Scala

10 Melendez, Steven Sometimes You're Just One Hop From Something Huge (May 27, 2014). Дата обращения 11 июня 2014

11 Farr, Christina Firebase's scalable backend makes it '10 times easier' to build apps (February 13, 2013). Дата обращения 12 июня 2014.

12 Marshall, Matt Firebase is building a Dropbox for developers (August 29, 2013). Дата обращения 12 июня 2014.

13 Metz, Cade How to Build Your Own Google Docs (Without Google) (04.09.13). Дата обращения 12 июня 2014.

14 Firepad - An open source collaborative code and text editor., Inc. Дата обращения 12 июня 2014.

15 Firechat - open source realtime chat built on Firebase (недоступная ссылка). Firebase, Inc. Дата обращения 12 июня 2014. Архивировано 5 декабря 2014 года.

16 Безопасность жизнедеятельности. /Под ред. Н.А. Белова - М.: Знание, 2000 - 364с.

17 Самгин Э.Б. Освещение рабочих мест. – М.: МИРЭА, 1989. – 186с.

18 Справочная книга для проектирования электрического освещения. / Под ред. Г.Б. Кнорринга. – Л.: Энергия, 1976.

19 Борьба с шумом на производстве: Справочник / Е.Я. Юдин, Л.А. Борисов; Под общ. ред. Е.Я. Юдина – М.: Машиностроение, 1985. – 400с., ил.

- 20 <https://inbusiness.kz/ru/news/v-kazahstane-podros-vnutrennij-turizm>
- 21 https://tourlib.net/statti_tourism/shilibekova.htm
- 22 <https://articlekz.com/article/8784>
- 23 <https://articlekz.com/article/8507>

Приложение А

Техническое задание

1 Общие сведения

1.1 Наименование системы

Полное наименование системы:

Разработка информационной системы «Турист - путеводитель»

1.2 Сроки начала и окончания работ

Дата начала: 17.02.2020

Дата окончания: 09.05.2020

2 Назначение и цели создания системы

2.1 Назначение системы

Обучающая система предназначена для обучения детей с расстройством аутистического спектра, развития коммуникативных навыков, речевой активности и познавательных способностей.

3 Рекомендации к разработке программы

Информационная система может быть разработана на язык Java с использованием СУБД FireBase.

4 Требование к внешнему виду системы

Обязательным условием является простота и понятливость интерфейса, не яркая, приятная цветовая гамма.

4 Технические требования

Специальных технических требований для ПК для работы приложения нет, кроме как Windows 7 или. Подойдёт любой ПК, на котором может работать Windows.

Приложение Б (листинг программы)

```
?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  package="com.example. tour">

  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
  <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />

  <application
    android:name=".TourApplication"
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme"
    android:usesCleartextTraffic="true"
    tools:ignore="AllowBackup,GoogleAppIndexingWarning,UnusedAttribute">

    <meta-data android:name="com.google.android.geo.API_KEY"
      android:value="@string/map_key"/>

    <activity
      android:name=".MainActivity"
      android:label="@string/app_name"
      android:theme="@style/AppTheme.Blue.NoActionBar" />
    <activity
      android:name=".SplashScreenActivity"
      android:theme="@style/AppTheme.NoActionBar">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>

```

```
</activity>
<activity
    android:name=".LanguageActivity"
    android:theme="@style/AppTheme.NoActionBar" />
<activity
```

Продолжение приложения Б

```
    android:name=".WebViewActivity"
    android:theme="@style/AppTheme.NoActionBar">
```

```
</activity>
<activity android:name=".ui.category.CategoryInfoActivity"/>
<activity
    android:name=".ui.category.CategoryDetailActivity"
    android:theme="@style/AppTheme.Blue.NoActionBar">
```

```
</activity>
<activity
    android:name=".ui.regions.RegionsDetailActivity"
    android:theme="@style/AppTheme.Blue.NoActionBar">
```

```
</activity>
</application>
```

```
</manifest>
```

```
package com.example.tour;
import android.view.View;
import androidx.annotation.CallSuper;
import androidx.annotation.UiThread;
import butterknife.Unbinder;
import butterknife.internal.DebouncingOnClickListener;
import butterknife.internal.Utils;
import java.lang.IllegalStateException;
import java.lang.Override;

public class LanguageActivity_ViewBinding implements Unbinder {
    private LanguageActivity target;
```

```
private View view7f080026;
```

```
private View view7f080027;
```

```
@UiThread
```

```
public LanguageActivity_ViewBinding(LanguageActivity target) {  
    this(target, target.getWindow().getDecorView());  
}
```

```
@UiThread
```

```
public LanguageActivity_ViewBinding(final LanguageActivity target, View source)  
{  
    this.target = target;
```

```
View view;
```

```
view = Utils.findRequiredView(source, R.id.btn_eng, "method 'onClickEng'");
```

Продолжение приложения Б

```
view.setOnClickListener(new DebouncingOnClickListener() {
```

```
    @Override
```

```
    public void doClick(View p0) {
```

```
        target.onClickEng();
```

```
    }
```

```
});
```

```
view = Utils.findRequiredView(source, R.id.btn_rus, "method 'onClickRus'");
```

```
view7f080027 = view;
```

```
view.setOnClickListener(new DebouncingOnClickListener() {
```

```
    @Override
```

```
    public void doClick(View p0) {
```

```
        target.onClickRus();
```

```
    }
```

```
});
```

```
}
```

```
@Override
```

```
@CallSuper
```

```
public void unbind() {
```

```
    if (target == null) throw new IllegalStateException("Bindings already cleared.");
```

```
    target = null;
```

```

        view7f080026.setOnClickListener(null);
        view7f080026 = null;
        view7f080027.setOnClickListener(null);
        view7f080027 = null;
    }
}

```

```

package com.example.tour.ui.category;

```

```

import android.view.View;
import android.widget.TextView;
import androidx.annotation.CallSuper;
import androidx.annotation.UiThread;
import androidx.appcompat.widget.Toolbar;
import androidx.recyclerview.widget.RecyclerView;
import butterknife.Unbinder;
import butterknife.internal.Utils;
import com.example.tour.R;
import java.lang.IllegalStateException;
import java.lang.Override;

```

```

public class CategoryDetailActivity_ViewBinding implements Unbinder {
    private CategoryDetailActivity

```

Продолжение приложения Б

```

    @UiThread

```

```

    public CategoryDetailActivity_ViewBinding(CategoryDetailActivity target) {
        this(target, target.getWindow().getDecorView());
    }

```

```

    @UiThread

```

```

    public CategoryDetailActivity_ViewBinding(CategoryDetailActivity target, View
source) {
        this.target = target;

        target.toolbar = Utils.findRequiredViewAsType(source, R.id.toolbar, "field
'toolbar'", Toolbar.class);
        target.recyclerView = Utils.findRequiredViewAsType(source, R.id.recyclerView,
"field 'recyclerView'", RecyclerView.class);
        target.category_name = Utils.findRequiredViewAsType(source,
R.id.category_name, "field 'category_name'", TextView.class);

```

```

}

@Override
@CallSuper
public void unbind() {
    CategoryDetailActivity target = this.target;
    if (target == null) throw new IllegalStateException("Bindings already cleared.");
    this.target = null;

    target.toolbar = null;
    target.recyclerView = null;
    target.category_name = null;
}
}

```

```

package com.example.tour;
public final class BuildConfig {
    public static final boolean DEBUG = Boolean.parseBoolean("true");
    public static final String APPLICATION_ID = "com.example.tour";
    public static final String BUILD_TYPE = "debug";
    public static final String FLAVOR = "";
    public static final int VERSION_CODE = 1;
    public static final String VERSION_NAME = "1.0";
}

```

```

package com.example.tour;

```

```

import android.content.Context;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

```

Продолжение приложения Б

```

public class AutoFitGridLayoutManager extends GridLayoutManager {
Продолжение приложения Б

```

```

    private int columnWidth;
    private boolean columnWidthChanged = true;

```

```

    public AutoFitGridLayoutManager(Context context, int columnWidth) {
        super(context, 1);
    }

```

```

        setColumnWidth(columnWidth);
    }

    public void setColumnWidth(int newColumnWidth) {
        if (newColumnWidth > 0 && newColumnWidth != columnWidth) {
            columnWidth = newColumnWidth;
            columnWidthChanged = true;
        }
    }

    @Override
    public void onLayoutChildren(RecyclerView.Recycler recycler,
RecyclerView.State state) {
        if (columnWidthChanged && columnWidth > 0) {
            int totalSpace;
            if (getOrientation() == VERTICAL) {
                totalSpace = getWidth() - getPaddingRight() - getPaddingLeft();
            } else {
                totalSpace = getHeight() - getPaddingTop() - getPaddingBottom();
            }
            int spanCount = Math.max(1, totalSpace / columnWidth);
            setSpanCount(spanCount);
            columnWidthChanged = false;
        }
        super.onLayoutChildren(recycler, state);
    }
}
package com.example.tour;

import android.content.Context;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

import butterknife.ButterKnife;
import butterknife.Unbinder;

abstract class BaseActivity extends AppCompatActivity {

    Продолжение приложения Б
    private Unbinder unbinder;

```



```
protected abstract int getLayoutResourceId();
```

```
@Override
```

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(getLayoutResourceId());  
    unbinder = ButterKnife.bind(this);  
}
```

```
@Override
```

```
protected void attachBaseContext(Context base) {  
    super.attachBaseContext(TourApplication.localeManager.setLocale(base));  
}
```

```
@Override
```

```
public void onDestroy() {  
    unbinder.unbind();  
    super.onDestroy();  
}
```

```
}
```

LanguageActivity

```
package com.example.tour;
```

```
import android.content.Context;
```

```
import android.content.Intent;
```

```
import android.os.Bundle;
```

```
import butterknife.ButterKnife;
```

```
import butterknife.OnClick;
```

```
public class LanguageActivity extends BaseActivity {
```

```
    public static Intent newIntent(Context packageContext) {  
        return new Intent(packageContext, LanguageActivity.class);  
    }  
}
```

```
@Override
```

```
protected int getLayoutResourceId() {  
    return R.layout.activity_language;  
}
```

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
//    setContentView(R.layout.activity_language);
    ButterKnife.bind(this);
}

@OnClick(R.id.btn_eng)
void onClickEng() {
    TourApplication.localeManager.setNewLocale(this, 1);
    startActivity(MainActivity.newIntent(LanguageActivity.this));
    finish();
}

@OnClick(R.id.btn_rus)
void onClickRus() {
    TourApplication.localeManager.setNewLocale(this, 0);
    startActivity(MainActivity.newIntent(LanguageActivity.this));
    finish();
}
}

```

Продолжение приложения Б

MainActivity

```

package com.example.tour;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import com.google.android.material.navigation.NavigationView;
import androidx.fragment.app.Fragment;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;

import com.example.tour.ui.category.CategoryFragment;

```

```

import com.example.tour.ui.mapviews.MapviewFragment;
import com.example.tour.ui.regions.RegionsFragment;

import butterknife.ButterKnife;

public class MainActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener {

    private Toolbar toolbar;

    public static Intent newIntent(Context packageContext) {
        return new Intent(packageContext, MainActivity.class);
        Продолжение приложения Б
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ButterKnife.bind(this);

        toolbar = findViewById(R.id.toolbar);
        toolbar.setTitle("");
        setSupportActionBar(toolbar);

        DrawerLayout drawer = findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
            this, drawer, toolbar, R.string.navigation_drawer_open,
            R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.syncState();

        NavigationView navigationView = findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);
        navigationView.setCheckedItem(R.id.nav_category);

        getSupportFragmentManager().beginTransaction().replace(R.id.content_frame,
            new CategoryFragment()).commit();
    }

    //Needed for locale change!

```

```

@Override
protected void attachBaseContext(Context base) {
    super.attachBaseContext(TourApplication.localeManager.setLocale(base));
}

@Override
public void onBackPressed() {
    DrawerLayout drawer = findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        new AlertDialog.Builder(this)
            .setMessage(getString(R.string.alert_close_app))
            .setCancelable(false)
            .setPositiveButton(getString(R.string.alert_positive), new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    MainActivity.super.onBackPressed();
                }
            })
            .setNegativeButton(getString(R.string.alert_negative), null)
            .show();
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    // if (id == R.id.action_settings) {

```

Продолжение приложения Б

```

//      return true;
//    }

    return super.onOptionsItemSelected(item);
}

@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {

    Fragment fragment = null;

    switch (item.getItemId()) {
        case R.id.nav_category:
            toolbar.setTitle("");
            fragment = new CategoryFragment();
            break;
        case R.id.nav_regions:
            toolbar.setTitle("");
            fragment = new RegionsFragment();
            break;
        case R.id.nav_map:
            toolbar.setTitle("");
            fragment = new MapviewFragment();
        case R.id.nav_facts:
            break;

        Продолжение приложения Б
        case R.id.nav_flight:
            Intent intFlight = new Intent(this, WebViewActivity.class);
            intFlight.putExtra("url", "http://www.aviasales.kz");
            startActivity(intFlight);
            break;
        case R.id.nav_hotel:
            Intent intHotel = new Intent(this, WebViewActivity.class);
            intHotel.putExtra("url", "http://www.booking.com");
            startActivity(intHotel);

            break;
        case R.id.nav_lang:
            startActivity(new Intent(this, LanguageActivity.class));
            finish();
            break;
    }
}

```

```

    }

    if (fragment != null) {

getSupportFragmentManager().beginTransaction().replace(R.id.content_frame,
fragment).commit();
    }

    DrawerLayout drawer = findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}
}

```

SplashScreenActivity

```

package com.example.tour;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Window;
import android.view.WindowManager;

public class SplashScreenActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(
            WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash_screen);

        Thread timerThread = new Thread() {
            public void run() {

                sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            } finally {
                startActivity(LanguageActivity.newIntent(SplashScreenActivity.this));
            }
        };
    }
}

```

Продолжение приложения Б

```

        finish();
    }
}
};
timerThread.start();
}

```

TourApplication

```

public class TourApplication extends Application implements
Application.ActivityLifecycleCallbacks {

```

```

    private SharedPreferences prefs;

```

```

    public static LocaleManager localeManager;

```

```

    @Override

```

```

    public void onCreate() {

```

```

        super.onCreate();

```

```

        registerActivityLifecycleCallbacks(this);

```

```

        prefs = getSharedPreferences(Key.SHARED_PREFS, MODE_PRIVATE);

```

```

    }

```

```

    @Override

```

```

    public void onActivityCreated(Activity activity, Bundle bundle) {

```

```

    }

```

```

    @Override

```

```

    public void onActivityCreated(Activity activity) {

```

Продолжение приложения Б

@Override

```
public void onResume(Activity activity) { }
```

@Override

```
public void onPause(Activity activity) {  
}
```

@Override

```
public void onStop(Activity activity) { }
```

@Override

```
public void onSaveInstanceState(Activity activity, Bundle bundle) {  
}
```

@Override

```
public void onDestroy(Activity activity) {  
}
```

@Override

```
protected void attachBaseContext(Context base) {  
    localeManager = new LocaleManager(base);  
    super.attachBaseContext(localeManager.setLocale(base));  
}
```

@Override

```
public void onConfigurationChanged(Configuration newConfig) {  
    super.onConfigurationChanged(newConfig);  
    localeManager.setLocale(this);  
}  
}
```

WebViewActivity

```
public class WebViewActivity extends AppCompatActivity {
```


Продолжение приложения Б

```
@SuppressWarnings("SetJavaScriptEnabled")
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.webview);
    Intent i = getIntent();
    String url = i.getStringExtra("url");
    WebView webView = findViewById(R.id.webview);
    webView.getSettings().setJavaScriptEnabled(true);
    webView.setWebViewClient(new WebViewClient());
    webView.loadUrl(url);
}
```