

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

Кафедра

вычислительной техники

«Вычислительная техника»

Специальность: 09.05.01 «Применение и эксплуатация
автоматизированных систем специального назначения»

Специализация: «Эксплуатация вычислительных машин, комплексов,
систем и сетей специального назначения»

ДИПЛОМНЫЙ ПРОЕКТ

на тему

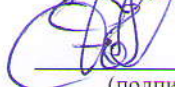
Система ивент-менеджмента

Студент



Тошев Евгений Николаевич
(ФИО полностью)

Руководитель



Федюнин Р.Н.
(фамилия, инициалы)

Нормоконтролёр



Кучин А.В.
(фамилия, инициалы)

Рецензент

(должность, место работы)



Васильков А.В.
(фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 10.06.2020 г. № 13)

Заведующий кафедрой



Митрохин М.А.
(фамилия, инициалы)

Работа защищена с отметкой Отлично (протокол заседания ГЭК от 19.06.2020 г. № 9)

Секретарь ГЭК



Гурин Е.И.
(фамилия, инициалы)

Пенза 2020

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

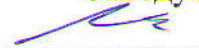
Вычислительной техники

Кафедра

Вычислительная техника

«Утверждаю»

Заведующий кафедрой

 Митрохин М.А.

«30» апреля 2020г.

**ЗАДАНИЕ
НА ДИПЛОМНЫЙ ПРОЕКТ**

1. Тошев Евгений Николаевич гр. 15В02 факультета вычислительной техники.

2. Тема работы: «Система ивент менеджмента».

Тема утверждена приказом ПГУ № 0335а/о от 30.04.2020 г.

3. Руководитель работы к.т.н., доцент каф. ВТ ПГУ Федюнин Р.Н.

4. Задание на работу (назначение разработки, исходные данные и т.п.).

Система предназначена для управления командной работой группы участников волонтерских или иных мероприятий.

Система состоит из следующих частей: мобильное приложение (модуль управления участником команды), Bluetooth-маяки (модуль слежения за заданиями группы), сервер управления маяками (модуль регистрации, добавления и удаления маяков и заданий в системе).

5 Перечень подлежащих разработке вопросов

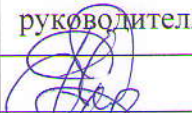
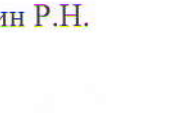
Анализ предметной области, выбор стека технологий и компонентов для реализации системы.

Разработка UML диаграмм для описания архитектуры приложения и потоков данных.

Разработка и отладка приложения.

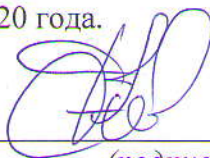
Реализация глав по БЖД и экономике.

6 Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области	17	22.03.2020	
2.	Обзор аналогов	15	06.04.2020	
3.	Анализ требований на разработку	15	19.04.2020	
4.	Разработка мобильного приложения	15	07.05.2020	
5.	Настройка системы	15	07.05.2020	
6.	Отладка приложения	11	08.05.2020	
7.	Оформление выпускной работы	12	10.06.2020	

Дата выдачи задания 30 «апреля» 2020 года.

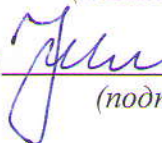
Руководитель дипломного проекта



Федюнин Р.Н.

(подпись, дата)

Задание к исполнению принял студент



30.04.2020

Тошев Е.Н.

(подпись, дата)

Работу к защите допустить

Декан факультета



Фионова Л.Р.

(подпись, дата)

РЕФЕРАТ

Дипломный проект содержит 120 листов, 31 рисунок, 15 таблиц, 7 источников.

ИВЕНТ МЕНЕДЖМЕНТ, ОС ANDROID, REST API, БАЗА ДАННЫХ, BLUETOOTH.

Цель работы – разработать масштабируемую, отказоустойчивую систему, позволяющую группе людей отслеживать расположение задач на карте местности и координировать свои действия при их выполнении.

Объектом разработки является программное приложение, позволяющее команде организовать участие в мероприятиях типа волонтерских движений и хакатонов.

В результате проведенной работы должен быть разработан проект, удовлетворяющий поставленной задаче, а именно, создан программный комплекс и разработаны соответствующие алгоритмы, реализующие возможность хранить, обрабатывать и доставлять до потребителя информацию о запланированных мероприятиях в виде сообщений.

Данное приложение рекомендуется использовать группе людей для выполнения поставленных задач на карте местности.

					ПГУ 1.090501.14.001 ПЗ				
Изм.	Лист	№ докум.№	Подпись	Дата					
Разраб.		Тошев Е.Н.			Система ивент менеджмента. Пояснительная записка	Лит.	Лист	Листов	
Провер.		Федюнин Р.Н.					2	120	
Н. Контр.		Кучин А.В.				ФВТ, каф. ВТ гр. 15В02			
Реценз.		Васильков А.В.							
Утверд.		Митрохин М.А.							

Содержание

Введение.....	5
1 Анализ предметной области	7
1.1 Сбор материала по теме	7
1.2 Выбор пути решения проблемы	10
1.3 Выбор стека технологий мобильной разработки.....	11
1.3.1 Стек мобильной разработки IOS	11
1.3.2 Стек универсальной мобильной разработки	11
1.3.3 Приложения на базе Exprelex.....	12
1.3.4 Приложения на базе облачных технологий	13
1.3.5 Стек мобильной разработки Android	14
1.4 Выбор способа хранения данных	14
1.5 Механизм идентификации в системе.....	20
1.6 Механизм реализации электронных маяков	21
1.7 Выводы по главе.....	22
2. Проектирование работы системы.....	23
2.1 Требования к функциональному назначению приложения.....	24
2.2 Разработка базы данных	25
2.2.1 Проектирование структуры данных.....	25
2.2.2 Логическое проектирование	26
2.2.3 Даталогическое проектирование	29
2.2.4 Инфологическая модель	34
2.3 Проектирование структуры программного обеспечения	36
2.4 Настройка beacons сервера.....	40
2.5 Выбор электронного маяка	43

2.6 Реализация работы с маяками в мобильном приложении	44
2.7 Проектирование пользовательского интерфейса.	49
3. Технология настройки среды реализации проекта	59
4. Техничко-экономическое обоснование разработки программы	68
4.1 Введение.....	68
4.2 Расчет затрат на разработку программного обеспечения.....	70
4.3 Расчет цены разработанной программы	72
4.4 Расчет капитальных вложений	73
4.5 Расчет эксплуатационных расходов.....	74
4.6 Определение расходов на интернет трафик	77
4.7 Заключение	78
5. Безопасность жизнедеятельности. Эргономика рабочего места программиста.....	79
5.1 Характеристика условий труда программиста	79
5.2 Эргономические требования к рабочему месту программиста	80
5.3 Освещенность рабочего места программиста.....	85
5.4 Расчет освещения рабочего места программиста.....	87
Заключение	91
Список используемых источников	92
ПРИЛОЖЕНИЕ А	93
ПРИЛОЖЕНИЕ Б.....	101
ПРИЛОЖЕНИЕ В	108
ПРИЛОЖЕНИЕ Г	116

Введение

В попытках сократить накладные расходы и транзакционные издержки, а также обеспечить наиболее полный объем информации для принятия решений организаторы мероприятий пытаются найти удобные платформы, сочетающие в себе функции планирования, организации, мониторинга мероприятия и выполнения специальных заданий участниками. Такое программное обеспечение смогло бы ускорить и облегчить процесс организации и проведения различных мероприятий, а также избавить работу организаторов от многооконного режима.

Раньше организаторы мероприятий могли использовать до семи различных инструментов при организации одного мероприятия. Организация стандартной конференции предполагала использование Wordpress для создания веб-страницы, Eventbrite для регистрации участников, Constant Contact для оптимизации работы с почтовым ящиком и стандартный Excel как дополнительный инструмент. Помимо данных приложений организатор использовал отдельные платформы для коммуникации с участниками. Каждое программное обеспечение поставлялось отдельным вендором, что создавало дополнительные транзакционные издержки.

Одновременная работа в нескольких программных окнах с разным интерфейсом не позволяет оптимизировать работу, а также соединить воедино информационные базы.

Дни многооконного режима прошли, поскольку целый ряд software-разработчиков пришли на рынок с готовыми решениями этой проблемы. Выбор софта зависит от индивидуальных требований заказчика и проектной специфики. К примеру, некоторые event-management платформы спроектированы специально для больших корпораций и масштабных мероприятий, которые несут в себе ограниченный функционал, однако позволяют работать с большими массивами данных. Именно поэтому исходить надо из проблем, которые испытывает event-менеджер при организации

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

мероприятий. И именно поэтому создание специализированного программного обеспечения для event-менеджмента всегда будет актуальным.

Так как большое внимание уделяется инструментам для организаторов мероприятий, но мало приложений для участников особенно командных мероприятий. Целью дипломного проекта будет создание приложения, которое поможет участнику и/или команде участников иметь полное представление о событиях на мероприятии с возможностью контроля членов и задач своей команды.

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

1 Анализ предметной области

В данной главе предлагается исследование решений для реализации проекта, стека технологий и описание бизнес-процессов, которые требуется реализовать в рамках данного проекта.

1.1 Сбор материала по теме

Анализ разрабатываемых систем начинается с нахождения проблемных зон в уже существующих приложениях данного типа и с описания функционала, который требуется реализовать в рамках приложения.

Профессиональные мобильные решения для event-специалистов и всех, кто организует b2b и b2c мероприятия – волонтерские движения, семинары, выставки, хакатоны имеют примерно один и тот же стандартный набор функций:

- полная программа события;
- описание контактов;
- схема площадки и/или описание мероприятия;
- push-уведомления;
- личный кабинет участников;
- статус участников;
- возможность кастомизации дизайна.

Все приложения в данном сегменте можно разделить на три вида:

- единое приложение с каталогом событий;
- конструктор приложений;
- смешанный тип.

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

Популярные мобильные приложения для пользователей и организаторов событий на 2019 год представлены на рисунке 1.1.

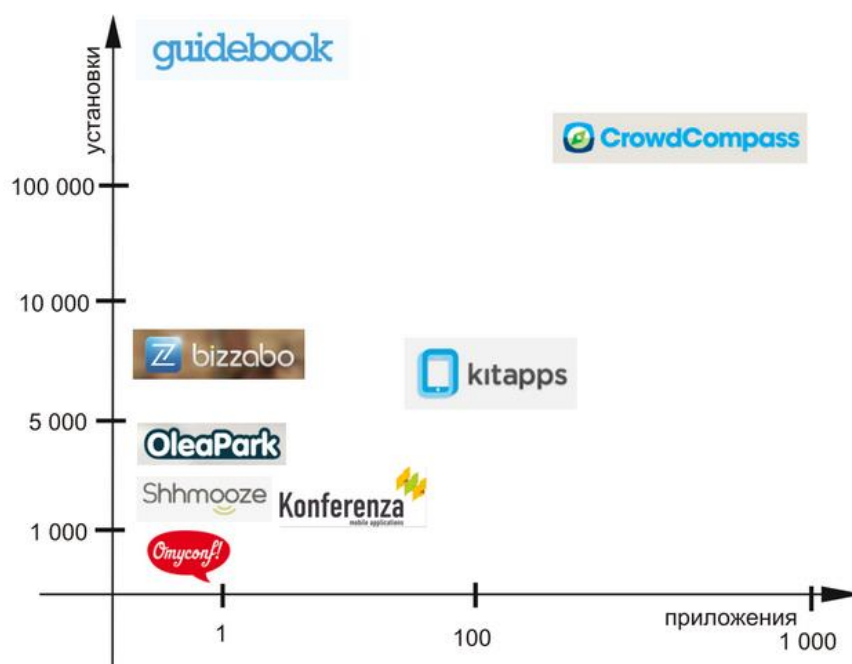


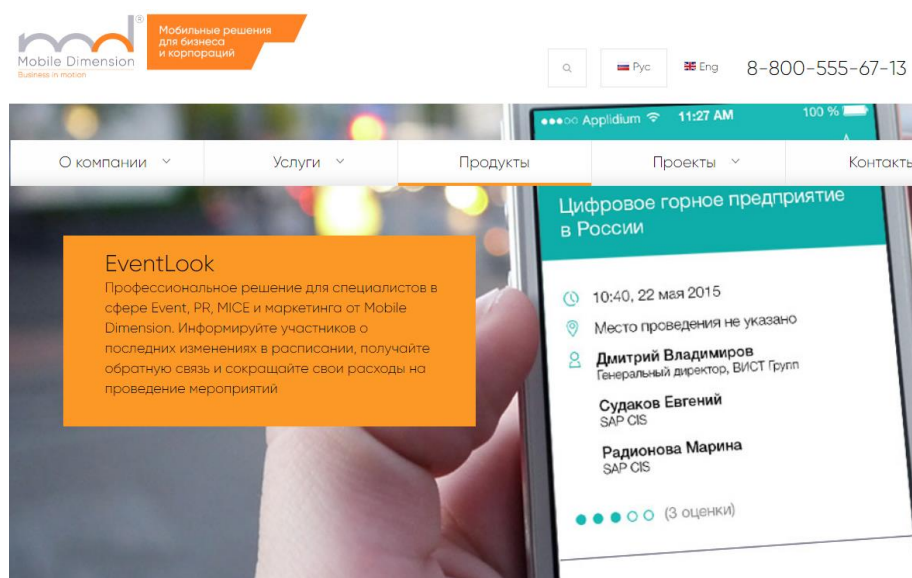
Рисунок 1.1 Популярные event приложения 2019 года

Сравним функционал популярных event приложений 2019 года и представим анализ в виде таблицы 1.1.

Таблица 1.1 Сравнительный анализ Event приложений

№	Функционал	Guidebook	Crowdcompass	Bizzabo	Omyconf
1	Программа	+	+	+	+
2	Составление расписания	+	+	+	+
3	Список участников	+	+	+	+
4	Список команд	+	+	+	+
5	Карта	+	+	+	-
6	Информация	+	+	+	-
7	Размещение в соцсети	+	-	-	-
8	Сообщения и вопросы	+	-	-	+

Также существуют конструкторы мобильных приложений позволяющие создать приложение под мероприятие, которое станет удобным источником информации для участников, и инструментом коммуникации для организаторов (рисунок 1.2).



Мобильное приложение для мероприятий и конференций

Рисунок 1.2 Конструктор event приложений

Недостаток существующих решений очевиден, большинство из них рассчитано на индивидуума, а не на команду участников, функционал приложений стандартен и быстро под потребности группы людей адаптирован не будет.

Отсюда делается вывод, что разрабатываемое приложение должно обладать двумя свойствами:

- обладать функционалом полезным, как персоне, так и команде и/или иной организованной группе участников;
- быть приложением свободно распространяемым, с открытым исходным кодом, для того чтобы любая команда участников event мероприятия могла доработать приложение под свои нужды.

Более того существующие решения зачастую не имеют возможности идентифицировать задачу и участника, для чего требуется использовать или

отдельное ПО, так называемые task manager или использовать универсальные решения, например MS Office.

1.2 Выбор пути решения проблемы

Для решения вышеописанных задач требуется определить функционал реализуемого приложения и выбрать стек технологий реализации приложения. Приложение реализуется в форм-факторе мобильного приложения.

Реализуемое приложение должно обладать следующими функциональными возможностями:

- приложение обеспечивает структурированное хранение информации о событиях, на которые пользователь зарегистрировался.
- пользователь может приглашать на мероприятия свою команду и отслеживать, какие члены команды посещают какие события и какую роль они там представляют;
- реестр событий - хранение основной информации о событиях, наименование, описание, даты проведения мероприятия, местоположение, ссылка на сайт и т. д.;
- управление статусом событий - способ различать события по приоритетам;
- участники - отслеживание членов команды, которые посещают события на мероприятие;
- свойства события - система позволяет пользователю описывать то или иное событие;
- авторизация - пользователи могут указывать роли в команде.
- отслеживание задачи и/или участника по карте на основании электронного маяка.

1.3 Выбор стека технологий мобильной разработки

В данном разделе будет описан выбор платформы реализации мобильных приложений. От данного выбора зависят инструменты реализации, сторонние библиотеки и облачные сервисы, с которыми мобильное приложение сможет работать.

1.3.1 Стек мобильной разработки IOS

В разработке мобильных приложений под IOS применяется SWIFT. Использование языка Objective-C становится плохим тоном для большинства разработчиков мобильных приложений.

Язык SWIFT хоть и имеет большие заимствования у своего прародителя Objective-C, но имеет и ряд ключевых преимуществ. К самому конкурентному преимуществу можно отнести поддержку со стороны Apple. Можно с уверенностью считать, что созданные мобильные приложения IOS на SWIFT будут поддерживаться значительно дольше, чем аналоги, в основу которых лег код Objective-C.

Развитие языка SWIFT в уже обозримом будущем, позволит разрабатывать приложения для ПК систем не только под управлением MacOS, но и Windows, Linux. Этот язык программирования имеет платформу для реализации приложений взаимодействия с пользователем (интерактив).

1.3.2 Стек универсальной мобильной разработки

При разработке мобильных приложений для бизнеса не редким требованием является скорость разработки с минимальным бюджетом.

В этом случае, на помощь приходят такие технологии разработки как PhoneGap и Xamarin. Обычно создание мобильных приложений на этих

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

технологиях применяется для пробных версий (прототипа или MVP), но могут так же быть применены и для полноценной работы.

1.3.3 Приложения на базе Exrelex

Большинство задач могут быть упрощены до простого взаимодействия с пользователем через последовательность сообщений и реакции на них. Это наиболее часто применимо к приложениям для бизнеса. Инновационный подход предлагает система связи - Exrelex. Суть подхода заключается в том, что уже имеется готовая серверная часть, в который может быть дополнен модуль бизнес-логики.

В качестве конечного приложения используется набор готовых протестированных модулей, которые доступны на всех мобильных и настольных операционных системах. Таким образом, создание приложений вообще не требуется, а достаточно только описать логику и настроить синхронизацию данных.

Приложения Exrelex обладают следующими возможностями:

- доставка сообщений с возможностью использования всех каналов связи (PUSH, Viber, Telegram, FB, VK, Skype, Email, SMS, Голосовые звонки);
- отчет прочтения (получения и прочтения);
- возможность выстраивания последовательности доставки (Работник-заместитель-руководитель);
- тонкая настройка сигнализации входящих сообщений (индикация, вибрация, мелодия, события);
- постановка задач и гео задач;
- корпоративный мессенджер;
- гибкая система контактов;
- возможность подписи сообщения (рукописная, биометрическая);
- функции навигации трекинга. Задачи основанные на геоданных.

1.3.4 Приложения на базе облачных технологий

В разработке приложений на базе облачных технологий применяются различные технологии в зависимости от поставленной задачи.

Для проектов с низкой нагрузкой, проектов, связанных с интернет-обслуживанием, преимущественно используются модули, написанные на PHP. Реже для этих целей применяется Perl/Python. Энтепрайз проекты строятся с применением технологий JAVA EE. Проекты с высоким уровнем нагрузки и уровнем отказоустойчивости - Erlang.

Хранение данных, быстрая индексация осуществляется за счет применения СУБД PostgreSQL. Встроенная логика, хранимые процедуры, триггеры и другие расширения позволяют построить консолидированное хранилище данных. Следует отметить, что PostgreSQL является свободным ПО и может функционировать как в среде сервера Linux, так и в Windows. Для малых интернет-проектов, более рационально использование БД MySQL.

Промежуточная «прослойка» между клиентом (мобильным приложением) и сервером (СУБД и/или облако) чаще всего служит протокол обмена (API). В большинстве проектов применяется технология REST API. Этот вид протокола является стандартом в обслуживании и поддерживается 95% разработчиков.

В качестве технологии передачи данных между слоями приложения будет использован REST API (JSON), но облачные технологии, возможно, будут использованы как вспомогательный ресурс. Основным стеком технологий для разработки приложения станут классические инструменты на базе паттерна MVC.

1.3.5 Стек мобильной разработки Android

Разработка Android-приложений это проверенная и надежная технология, к которой приходит каждый третий программист языка Java несмотря на то, что существует колоссальное число особенностей мобильной разработки. Работа с нативными функциями ОС Android позволяет создавать приложения Android, которые работают стабильно и быстро.

Из всех выше предложенных вариантов выбор остановлен на ОС Android, как системе максимально открытой для open source проектов. Под данную платформу реализовано множество библиотек и плагинов для подключения дополнительных функций. Более того платформа поддерживается и разрабатывается компанией Google, которая позволяет работы приложений с облачными и IoT решениями.

1.4 Выбор способа хранения данных

Сложно представить приложение, которое не использует базы данных (СУБД). Приложения обрабатывают, читают и записывают определенный набор данных.

Для решения различных задач разрабатывают большое количество СУБД (реляционные и NoSQL) и программ для работы с ними (MySQL, PostgreSQL, MongoDB, Redis и т.д.).

Если посмотреть DB-Engines Ranking, то можно увидеть, что в течение многих лет популярность open source баз данных растет.

Многие приложения используют несколько различных баз данных для того, чтобы задействовать их сильные стороны. Ни одна база данных не оптимизирована для всех возможных потребностей. С одной стороны, это хороший выбор, с другой нужно найти баланс, так как чем больше разных технологий задействовано в компании, тем сложнее их поддерживать.

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

Для того чтобы сузить круг используемых баз данных, можно определиться с основным операционным хранилищем и использовать вместе с ним какие-то дополнительные сервисы.

При выборе основного операционного хранилища возникает два основных варианта: с одной стороны — проверенные временем реляционные СУБД, с другой — различные не реляционные подходы к хранению данных.

Эти рассуждения и подводят к сравнению MySQL и MongoDB, которые, согласно DB-Engines Ranking, являются самыми популярными представителями реляционных и не реляционных баз данных соответственно.

К тому же компания MongoDB изначально фокусировалась на пользователях MySQL, поэтому у программистов есть опыт использования и выбор между этими двумя технологиями.

В данном разделе будет проведено сравнение производительности этих двух баз данных.

MongoDB - документно-ориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. Классифицирована как NoSQL, использует JSON-подобные документы и схему базы данных.

MySQL – свободная реляционная система управления базами данных.

Критерием сравнения будет скорость добавления, поиска, изменения и удаления записей из одной таблицы (рисунки 1.3-1.7).

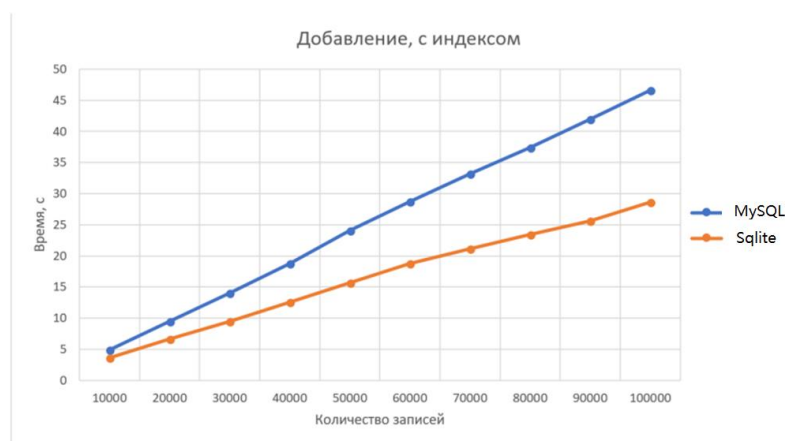


Рисунок 1.3 SQLite примерно на 63 процента быстрее MySQL

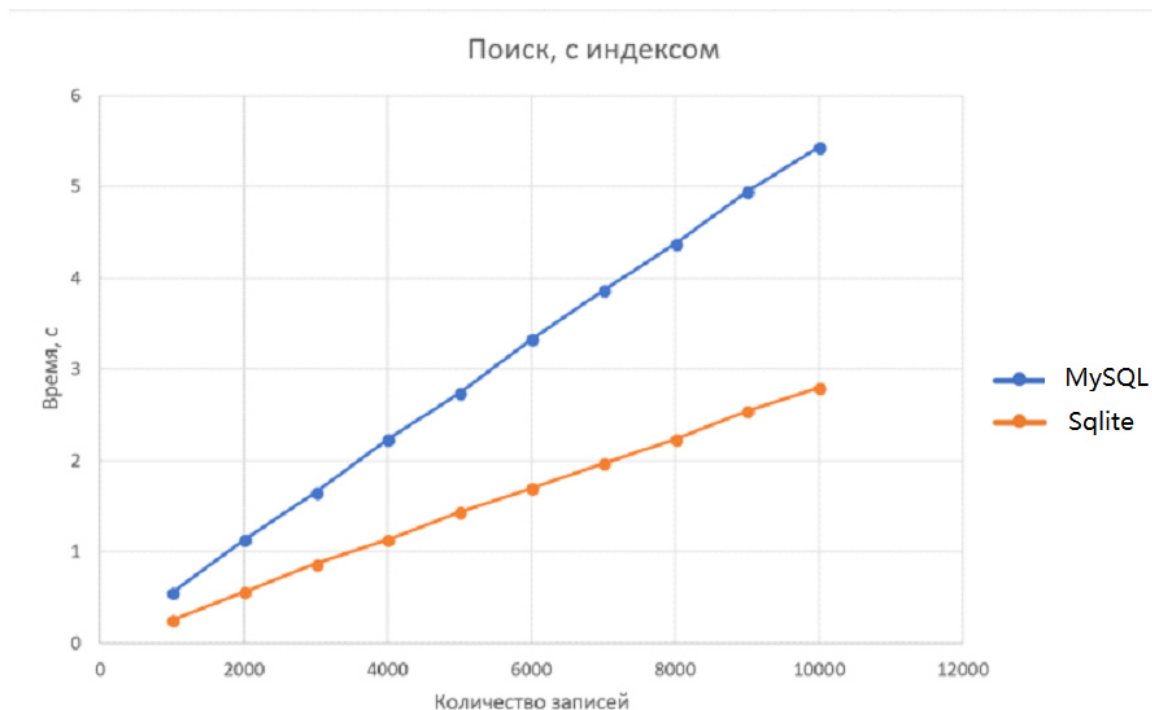


Рисунок 1.4 SQLite быстрее MySQL на 95 процентов

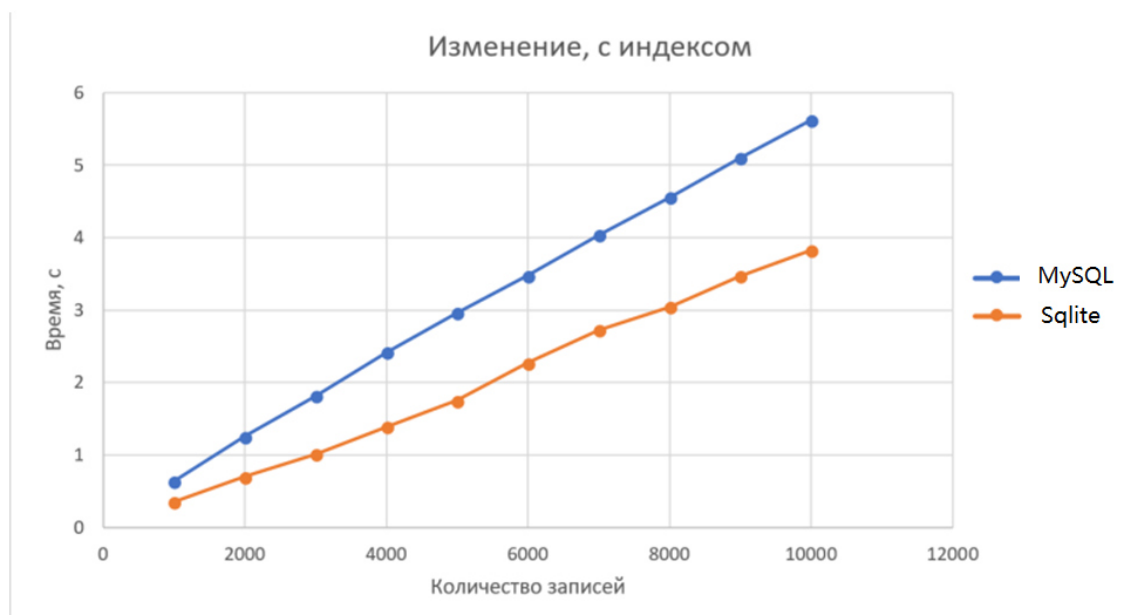


Рисунок 1.5 SQLite быстрее MySQL на 47 процентов

Таблица 1.2 Сравнительный анализ реляционных СУБД

№	Параметры	SQLite	MySQL
1	Описание	Свободная реляционная система управления базами данных	Свободная реляционная система управления базами данных
2	модель хранения данных	Реляционная база данных База данных типа Key/Value, документно- ориентированная база данных	Реляционная база данных База данных типа Key/Value, документно-ориентированная база данных
3	Документация	Имеется	Имеется
4	Лицензия	Открытое программное обеспечение, доступны коммерческие лицензии	Открытое программное обеспечение, доступны коммерческие лицензии
5	Облачное представление	Нет	Нет
6	Поддерживаемые операционные системы сервера	Linux, OS X, Solaris, Windows, FreeBSD	Linux, OS X, Solaris, Windows
7	Схема данных	Строгая	Свободная
8	Поддержка XML	Да	Нет
9	Поддержка языка SQL	Да	Нет

Продолжение таблицы 1.2

10	Поддерживаемое API и другие методы доступа	Проприетарное API, ADO.NET, JDBC, ODBC	Проприетарный протокол с использованием JSON
11	Поддерживаемые языки программирования	Ada, C, C#, C++, D, Delphi, Eiffel, Erlang, Haskell, Java, JavaScript (Node.js), Objective-C, OCaml, Perl, PHP, Python, Ruby, Scheme, Tcl	ActionScript, C, C#, C++, Clojure, ColdFusion, D, Dart, Delphi, Erlang, Go, Groovy, Haskell, Java, JavaScript, Lisp, Lua, MATLAB, Perl, PHP, PowerShell, Prolog, Python, R, Ruby, Scala, Smalltalk
12	Триггеры	Да	Нет
13	Методы репликации	Горизонтальное разбиение, шардинг с MySQL Cluster или MySQL Fabric	Шардинг
14	Внешние ключи	Master-Master, Master-Slave	Master-Slave
15	Внешние ключи	Да	Нет
16	Параллелизм	Да	Да
17	Контроль доступа пользователей	Нет групп или ролей	Назначение прав для отдельных пользователей и ролей

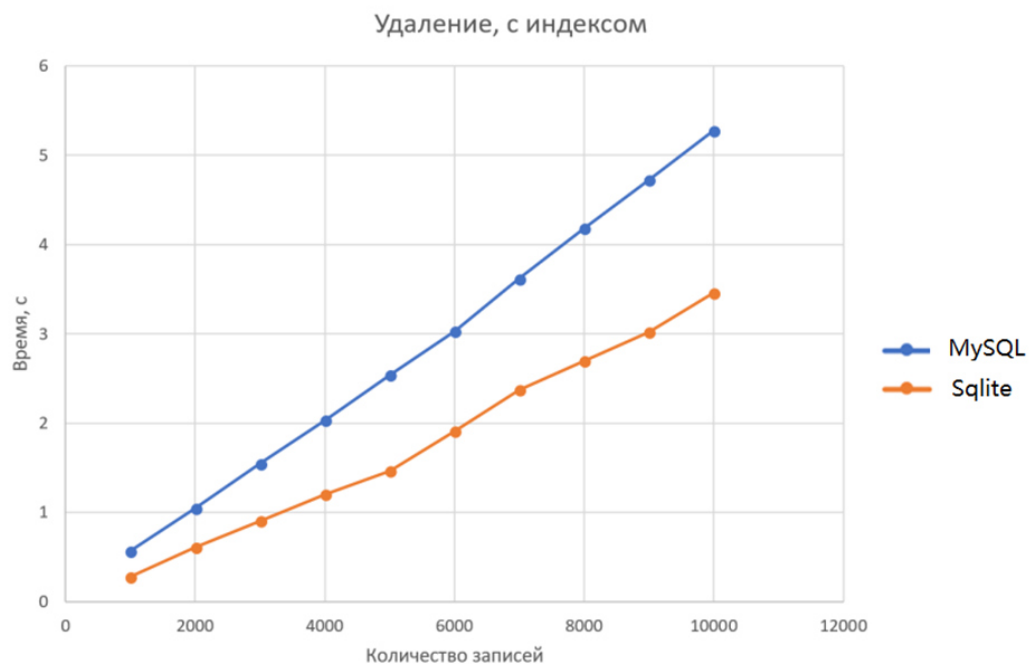


Рисунок 1.6 SQLite быстрее MySQL на 53 процентов

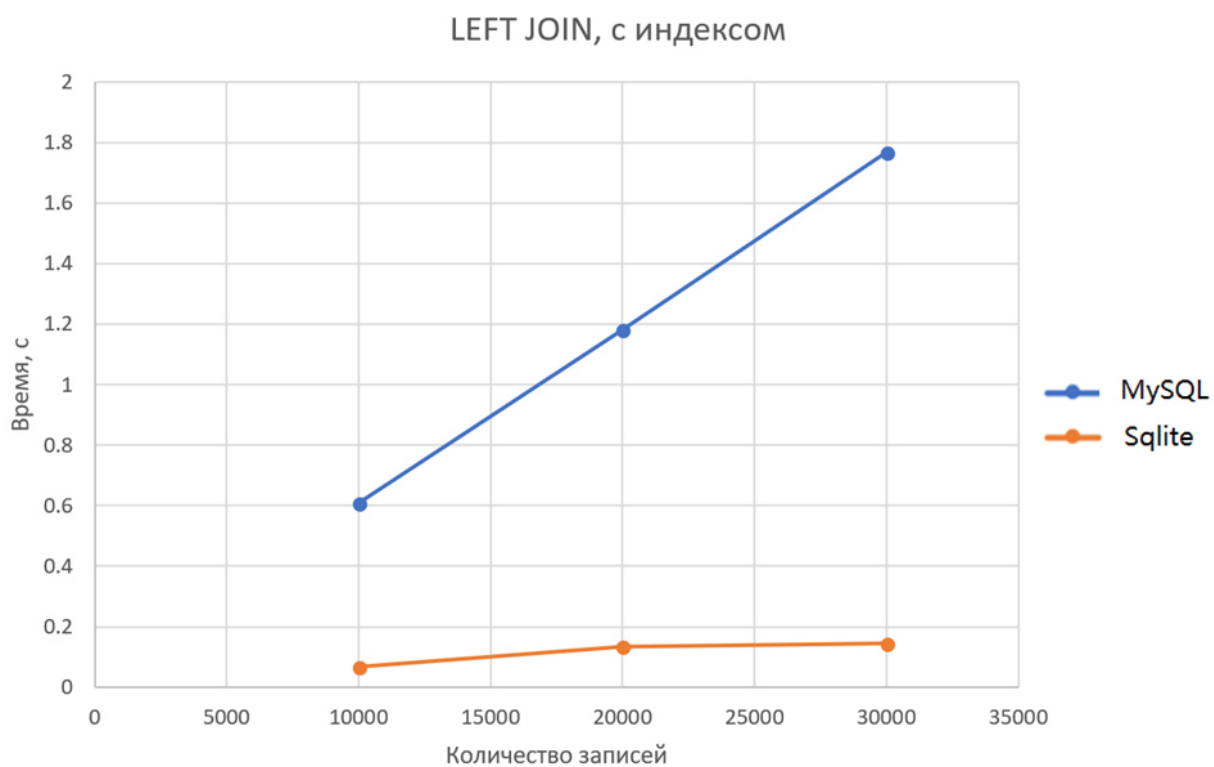


Рисунок 1.7 SQLite быстрее MySQL в 12 раз

В тестах, где выполнялись простые запросы данных из одной таблицы: SQLite оказалась быстрее при работе с таблицей, имеющей индекс, в то время как MySQL быстрее, когда индексы не использовались.

При этом разрыв между двумя СУБД был не более чем в два раза, а не на порядок, как в случае с LEFT JOIN.

Если же рассматривать работу с данными, имеющими некоторые зависимости друг от друга, то тут выбор очевиден — это SQLite.

Кроме того, SQLite в отличие от конкурента не поддерживает модель «клиент-сервер», она встроена в конечную программу. Это означает, что можно связать базу данных SQLite с приложением и получить доступ ко всем возможностям БД в своем приложении даже в режиме «оффлайн».

1.5 Механизм идентификации в системе

Аутентификация в приложении будет реализована с помощью протокола OAuth v. 2.0 (в проекте используется свободно распространяемая библиотека omniauth).

Основной особенностью приложения будет то, что приложение не будет хранить пароли. Если пользователь хочет войти в систему, он обращается к внешнему серверу авторизации, чтобы произвести процесс авторизации и/или аутентификации пользователя.

Пользователю будет предоставлен доступ к личной информации на внешнем сервере авторизации. Как только доступ предоставлен, приложение получает токен доступа с разрешением для пользователя. Затем приложение делает еще один запрос к серверу ресурсов (который в большинстве случаев совпадает с сервером авторизации), второй запрос предоставит личную информацию пользователя. Часть информации является уникальным идентификатором (UID) пользователя для данного сервера авторизации. Если у пользователя уже есть учетная запись, UID уже может быть найден в базе

данных, и пользователь проходит аутентификацию. Если такой UID не найден, учетная запись создается с использованием личной информации пользователя.

На данный момент omniauth поддерживает следующих провайдеров аутентификации: Twitter, Facebook и Google.

1.6 Механизм реализации электронных маяков

Так как одной из функций приложения должна быть связь мобильного телефона участника мероприятия с электронными маяками, которые позволяют выполнить задачу, поставленную перед командой.

Необходимо выбрать технологические решения, которые позволят реализовать данную техническую задачу.

Технология, которая позволяет выполнить данную задачу, называется Physical Web, это своеобразный мост между мобильным приложением, облачными сервисами и электронными маяками. В своей основе Physical Web является службой обнаружения. Умный объект передает соответствующий URL-адрес, который могут принимать любые устройства поблизости, например ваш смартфон и далее этот URL или содержит информацию, которую приложение обрабатывает или передается в облачный сервис для принятия решения.

В настоящий момент данную технологию реализуют два инструмента iBeacon и Eddystone это сервисы Apple и Google соответственно, использующие BLE (Bluetooth Low Energy) для локального позиционирования.

Так как для реализации приложения выбрана ОС Android, в систему будет интегрирован сервис Eddystone.

1.7 Выводы по главе

В результате исследований произведенных в первой главе сделаны следующие выводы:

- Выбранная тематика является актуальным направлением для бизнеса и социальных сообществ;
- Приложение для работы с ивентами будет реализовано на базе мобильного приложения (ОС Android);
- СУБД для хранения, обработки и структурирования данных будет SQLite;
- Процедуры идентификации и дополнительных функций будут реализованы с помощью свободно распространяемых библиотек, API.

2. Проектирование работы системы

Реализуемое мобильное приложение является частью системы, которая состоит из следующих модулей: мобильное приложение, облачные сервисы, beacons сервер.

Мобильное приложение позволяет пользователю управлять событиями и быть частью команды на мероприятии. Облачные сервисы предназначены для авторизации приложения и позиционирования пользователя в геолокации. Beacons сервер позволяет регистрировать и управлять электронными маяками, которые связаны с мобильным приложением заданием для пользователя и/или позволяют пользователю получить информацию необходимую для выполнения задания командой. На рисунке 2.1 представлена архитектура системы.

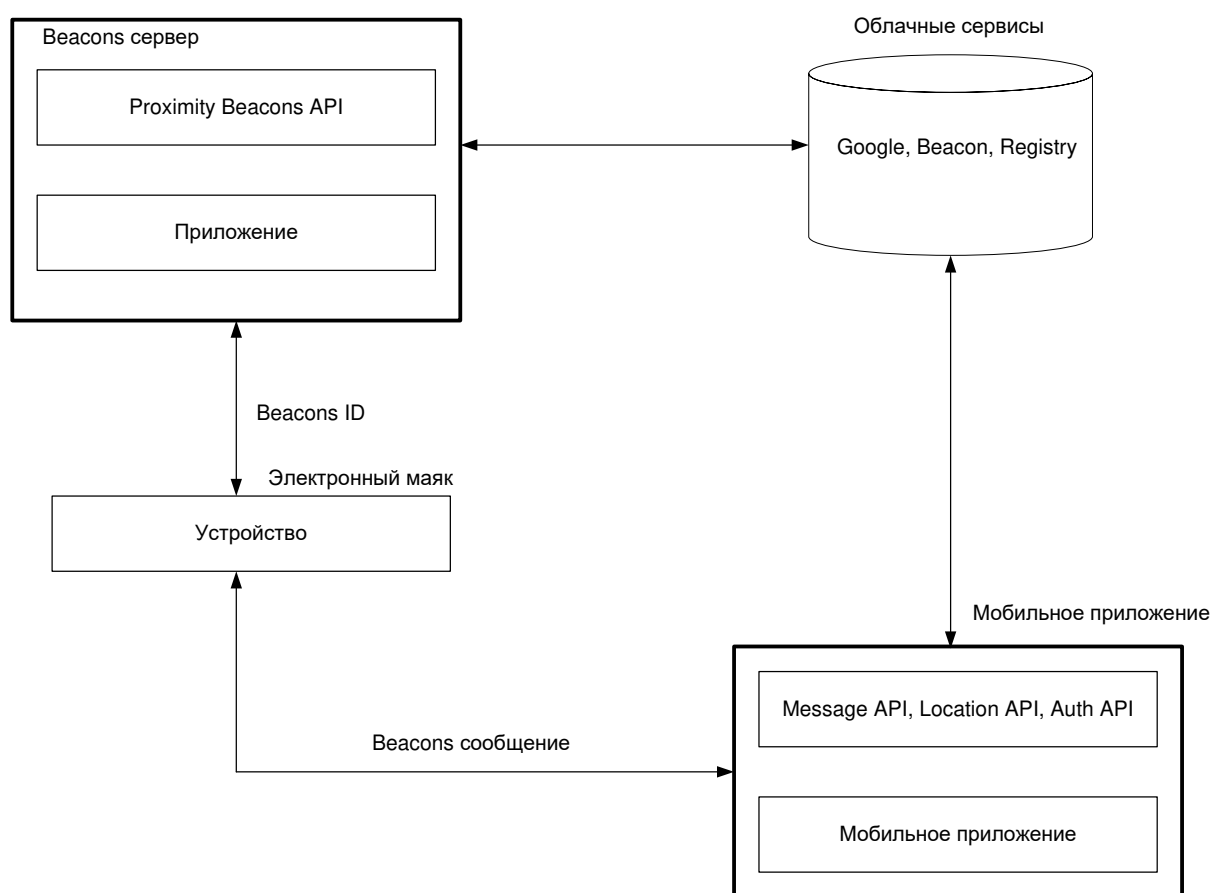


Рисунок 2.1 Архитектура системы

2.1 Требования к функциональному назначению приложения

Программа должна поддерживать работу разных типов пользователей. Возможности, которые должны быть реализованы в зависимости от пользователя, представлены в таблице 2.1.

Таблица 2.1 – Возможности пользователей

Тип пользователя	Возможности
Администратор	Приглашает членов команды. Координирует работу команды. Формирует персональные задачи.
Член команды	Координирует работу команды. Формирует персональные задачи.
Волонтер	Формирует персональные задачи.

В соответствии с основными функциями были разработаны диаграммы вариантов использования для каждого типа пользователей.

Главное требование к интерфейсу приложения это его простота. При работе с программой должна быть обеспечена минимальность затрат ресурсов со стороны пользователя. Основная рабочая форма должна включать в себя ссылки на необходимые страницы и рабочую область – область ввода данных. Должна быть предусмотрена возможность перед началом работы с приложением идентификации пользователя системы (проверки кода доступа). Далее должны быть доступны только те элементы интерфейса, которые реализуют функции, соответствующие правам доступа идентифицированного пользователя системы.

Этап разработки разбит на три части: проектирование базы данных, проектирование логики и проектирование интерфейса.

2.2 Разработка базы данных

2.2.1 Проектирование структуры данных

Процесс построения инфологической модели состоит из следующих шагов:

- определение сущностей;
- определение зависимостей между сущностями;
- задание первичных и альтернативных ключей;
- определение атрибутов сущностей.

Пользователь, работающий с программными средствами контроля успеваемости учащихся, будет добавлять информацию в базу, и работать с ней.

Далее в таблице 2.2 приведен обзор сущностей, где представлены все спроектированные сущности БД, заголовки столбцов означают следующее:

- название сущности – название выделенной в предметной области сущности;
- идентификация таблицы – физическое имя сущности;
- количество записей – предлагаемое количество записей сущности;
- описание – описание сущности.

Таблица 2.2 – Таблица сущностей

Название	Идентификатор	Описание
Пользователь	user	Информация о пользователях
Задачи	task	Информация о задачах
Мероприятия	event	Информация о мероприятиях, в которых группа может принять участие
Команда	team	Информация о команде

Продолжение таблицы 2.2

Роль в команде	attendee	Список ролей в команде: администратор, член команды, волонтер
Устройства	beacons	Список устройств доступный в рамках мероприятия
Право на использование устройства	tickets	Право пользователя воспользоваться устройством

2.2.2 Логическое проектирование

Логическое проектирование включает в себя следующие этапы:

- определение набора отношений исходя из концептуальной модели;
- проверка модели с помощью правил нормализации;
- проверка модели в отношении модели пользователей;
- создание полной атрибутивной диаграммы;
- определение требующейся поддержки целостности.

Анализ функциональных зависимостей в процессе логического проектирования для каждого отношения приведен ниже.

Отношение: Пользователь.

Схема отношения: Пользователи (идентификатор пользователя, имя участника, фамилия участника, e-mail участника).

Первичный ключ: идентификатор пользователя.

Функциональные зависимости:

Идентификатор пользователя → идентификатор команды, идентификатор события, идентификатор роли.

Отношение находится в 3НФ.

Отношение: Задачи.

Схема отношения: Задачи (идентификатор задачи, наименование задачи, описание задачи, исполнитель задачи, команда задачи).

Первичный ключ: идентификатор задачи.

Функциональные зависимости:

Идентификатор задачи → идентификатор пользователя, идентификатор мероприятия.

Отношение находится в 3НФ.

Отношение: Мероприятия.

Схема отношения: Мероприятия (идентификатор мероприятия, наименование мероприятия, описание мероприятия, начало мероприятия, окончание мероприятия, собственники мероприятия).

Первичный ключ: идентификатор мероприятия.

Функциональные зависимости:

Идентификатор мероприятия → идентификатор команды, идентификатор задачи, идентификатор пользователя.

Отношение находится в 3НФ.

Отношение: Команда.

Схема отношения: Команда (идентификатор команды, наименование команды).

Первичный ключ: идентификатор мероприятия.

Функциональные зависимости:

Идентификатор команды → идентификатор пользователя, идентификатор задачи, идентификатор события.

Отношение находится в 3НФ.

Отношение: Роль в команде.

Схема отношения: Роль в команде (идентификатор роли, наименование роли).

Первичный ключ: идентификатор мероприятия.

Функциональные зависимости:

Идентификатор роли в команде → идентификатор пользователя, идентификатор команды.

Отношение находится в ЗНФ.

Отношение: Устройства.

Схема отношения: Устройства (идентификатор устройства, ключ устройства в системе, идентификатор месторасположения устройства, номер устройства, координаты устройства).

Первичный ключ: идентификатор устройства.

Функциональные зависимости:

Идентификатор устройства → идентификатор права пользователя на использование устройства.

Отношение находится в ЗНФ.

Отношение: Право на использование устройства.

Схема отношения: Право на использование устройства (идентификатор, время использования, доступ к устройству).

Первичный ключ: идентификатор прав.

Функциональные зависимости:

Идентификатор прав → идентификатор устройства, идентификатор пользователя.

Отношение находится в ЗНФ.

2.2.3 Дatalogическое проектирование

На данном этапе разработки БД была сформирована физическая модель БД, изображенная в таблицах 2.3 - 2.9.

Таблица 2.3 – Отношение «Пользователь» («user»)

Имя столбца	Содержательное описание	Тип данных	Размерность	Возможность значения Null	Роль
iduser	Идентификатор пользователя	int	11	-	Primary Key
first_name	Имя	Varchar	45	-	Attribute
second_name	Фамилия	Varchar	45	-	Attribute
email	Почта	Varchar	45		Attribute
task	Задача	int	11		Foreign Key
event	Событие	int	11		Foreign Key
attendees	Роль в команде	int	11		Foreign Key
teams	Команда	int	11		Foreign Key
tickets	Устройство	int	11		Foreign Key

Таблица 2.4 – Отношение «Задачи» («task»)

Имя столбца	Содержательное описание	Тип данных	Размерность	Возможность значения Null	Роль
idtask	Идентификатор задачи	int	11	-	Primary Key
name	Наименование задачи	Varchar	45	-	Attribute
abstract	Описание	Varchar	45	-	Attribute
owner	Собственник задачи	int	11		Attribute
team	Команда	int	11		Attribute
evant_tasks	Ключ связи с мероприятием	int	11		Foreign Key

Таблица 2.5 – Отношение «Мероприятие» («task»)

Имя столбца	Содержательное описание	Тип данных	Размерность	Возможность значения Null	Роль
idevent	Идентификатор мероприятия	int	11	-	Primary Key

Продолжение таблицы 2.5

name	Наименование	Varchar	45	-	Attribute
description	Описание	Varchar	45	-	Attribute
begins_at	Начало	Data			Attribute
ends_at	Окончание	Data			Attribute
owner	организатор мероприятия	Varchar	45		Attribute
team	Команда	Varchar	45		Attribute
attendees	Роли	Varchar	45		Attribute
event_tasks	Типы мероприятий	int	11		Foreign Key
event_tasks	Типы задач	int	11		Foreign Key

Таблица 2.6 – Отношение «Команда» («team»)

Имя столбца	Содержательное описание	Тип данных	Размерность	Возможность значения Null	Роль
idteam	Идентификатор задачи	int	11	-	Primary Key

Продолжение таблицы 2.6

name	наименование задачи	Varchar	45	-	Attribute
events	Мероприятия	int	11		Foreign Key
tasks	Задачи	int	11		Foreign Key
users	Пользователи	int	11		Foreign Key
evant_types	Типы событий	int	11		Foreign Key
attendee_types	Роли пользователей	int	11		Foreign Key

Таблица 2.7 – Отношение «Роль в команде» («attendee»)

Имя столбца	Содержательное описание	Тип данных	Размерность	Возможность значения Null	Роль
idattendee	Идентификатор задачи	int	11	-	Primary Key
name	Наименование роли	Varchar	45	-	Attribute
user	Пользователь	int	11		Foreign Key

Таблица 2.8 – Отношение «Устройство» («beacons»)

Имя столбца	Содержательное описание	Тип данных	Размерность	Возможность значения Null	Роль
idbeacons	Идентификатор	int	11	-	Primary Key
UUID	Идентификатор устройства	Varchar	45	-	Attribute
namespaceID	Ключ на сервере	int	11		
instanseID	Код устройства	int	11		
coordX	Координаты устройства	double	11		
coordY	Координаты устройства	double	11		

Таблица 2.9 – Отношение «Право на использование устройства» («ticket»)

Имя столбца	Содержательное описание	Тип данных	Размерность	Возможность значения Null	Роль
idtickets	Идентификатор	int	11	-	Primary Key

Продолжение таблицы 2.9

time_at	Время использования	data		-	Attribute
time_to	Время использования	data		-	Attribute
completed	Доступность	int	11		Attribute

2.2.4 Инфологическая модель

Цель инфологического моделирования — обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Основными конструктивными элементами инфологических моделей являются сущности, связи между ними, идентификаторы (ключи) и свойства (атрибуты).

Инфологическая модель предметной области приведена на рисунке 2.2.

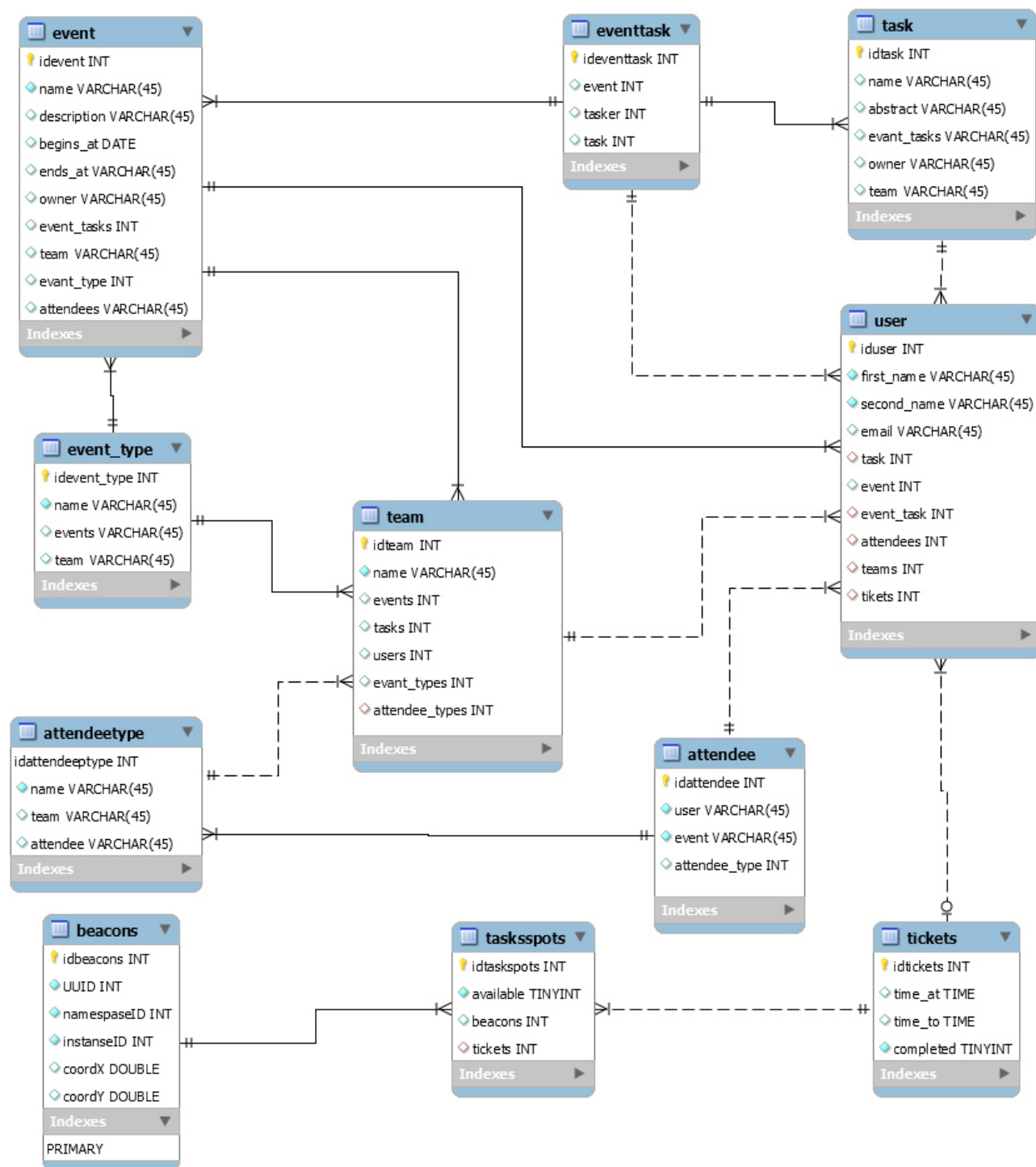


Рисунок 2.2 – Инфологическая модель базы данных

2.3 Проектирование структуры программного обеспечения

При проектировании программных средств для определения архитектуры разрабатываемой системы приведем диаграмму компонентов, изображенную на рисунке 2.3.



Рисунок 2.3 – Диаграмма компонентов приложения

На диаграмме изображено взаимодействие компонентов приложения.

Каждый экран пользовательского интерфейса представлен классом Activity является отдельной формой приложения. Приложение состоит из нескольких активностей и может переключаться между ними во время выполнения приложения.

Для отображения механизма взаимодействия подсистем программы, была построена диаграмма активностей, приведенная на рисунке 2.4.

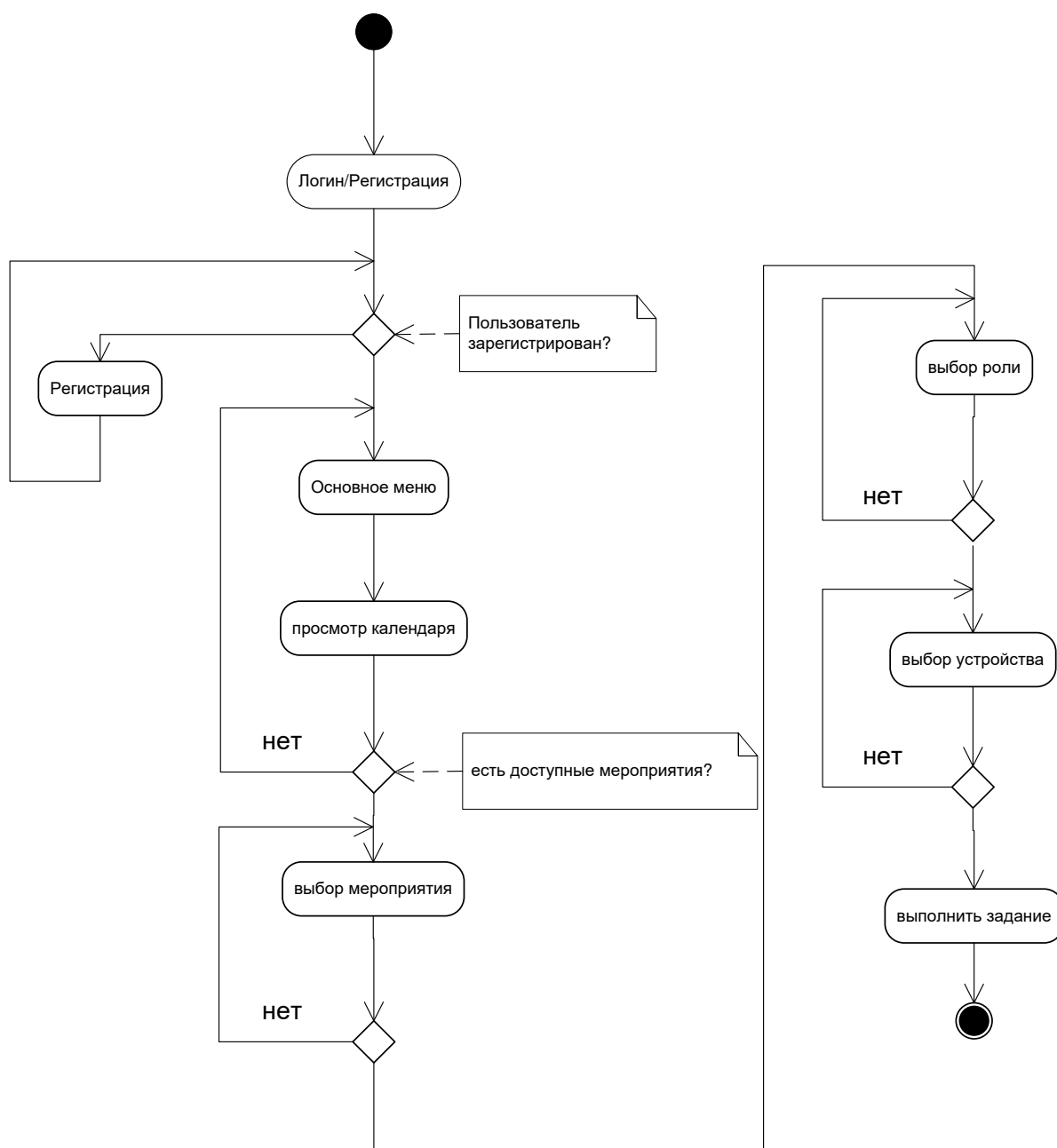


Рисунок 2.4 – Диаграмма компонентов активности

Как видно из диаграммы компонентов активности на рисунке 2.4, только авторизированный пользователь способен войти и использовать функционал приложения. При этом в зависимости от прав пользователя ему доступен определенный набор функций.

На примере диаграмм последовательности опишем работу приложения с точки зрения разных типов пользователей: администратора, пользователя и волонтера (рисунок 2.5 – 2.7).

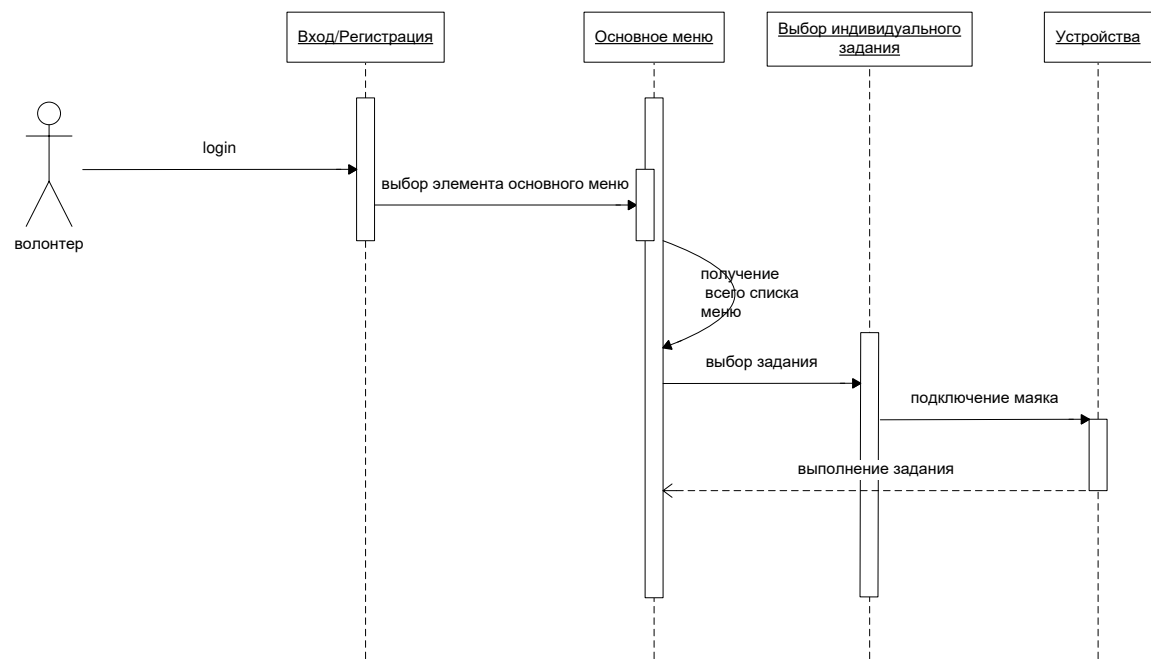


Рисунок 2.5 – Диаграмма последовательности «Волонтер»

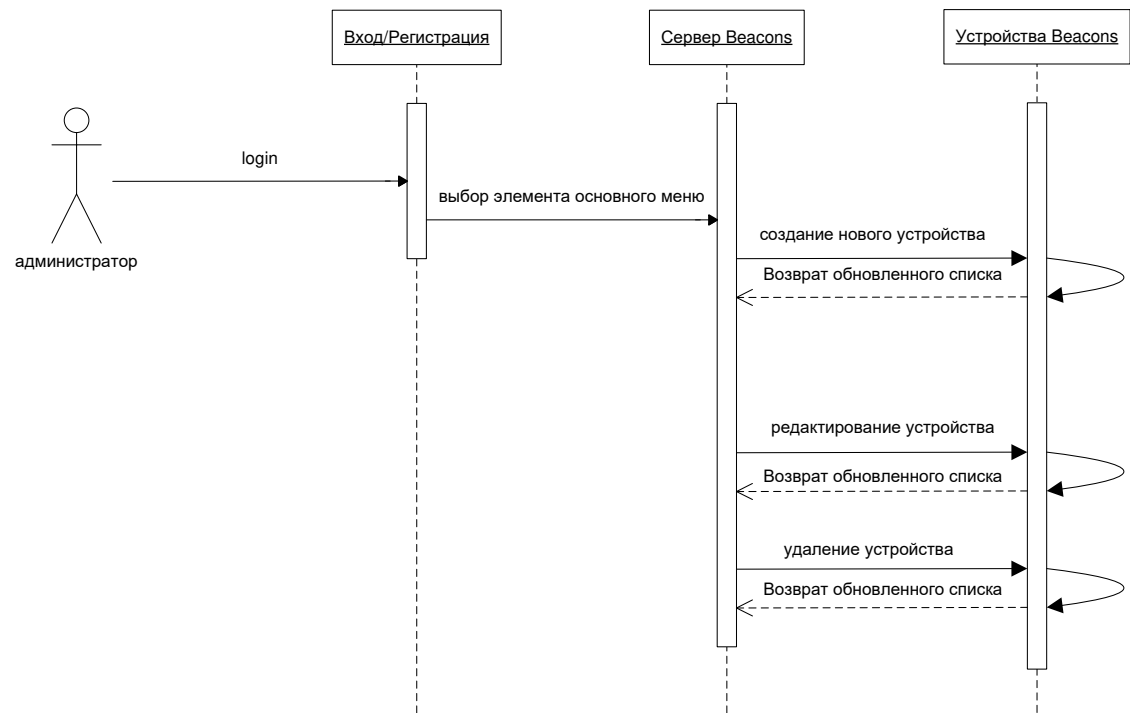


Рисунок 2.6 – Диаграмма последовательности «Администратор»

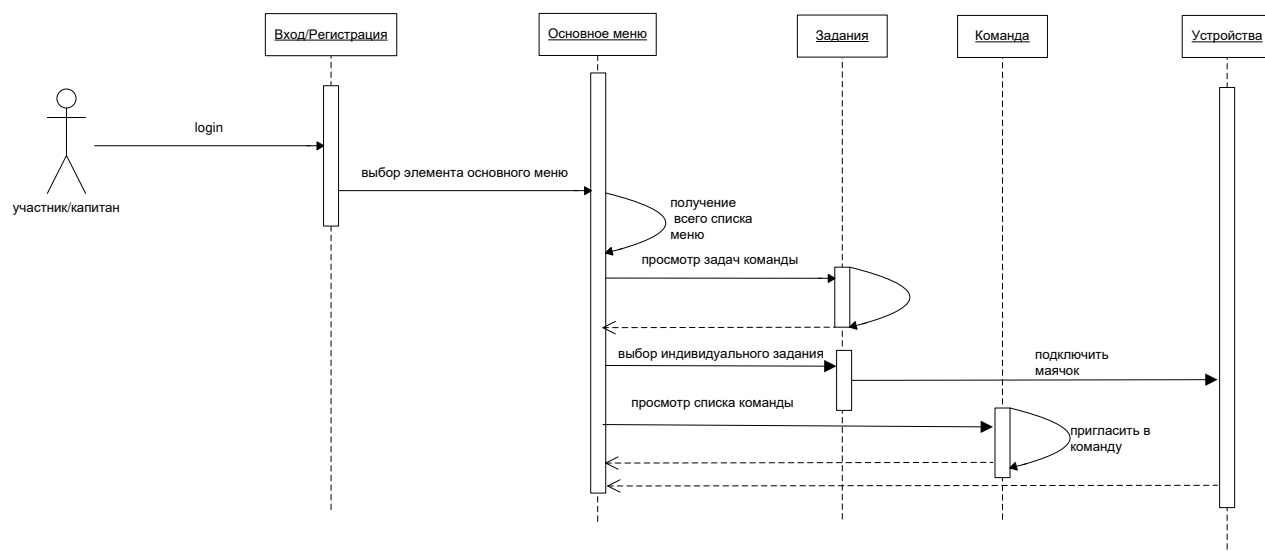


Рисунок 2.7 – Диаграмма последовательности «Пользователь»

Как видно из диаграмм выше (рисунки 2.5-2.7) волонтер способен просматривать мероприятия выбирать индивидуальные задания и исполнять их. При этом он также включает маяк для определения своего местоположения и для считывания с маяка информации о задании, которое он выбрал.

Администратор необходим как организатор мероприятия, его основная задача регистрация маяков в облачном сервере и привязка их к задаче, он также программирует маяки на показ задач и/или описания, которое можно считать приложением мобильного телефона.

Участник – является основным пользователем системы, он формирует команду, просматривает задачи, которые взяли на себя участники команды и также берет индивидуальную задачу на исполнение.

2.4 Настройка beacons сервера

Для администрирования маячков системе необходимо использовать Google Proximity Beacon API, с помощью которого можно регистрировать устройства в Google Beacons Registry. В нём хранится информация о зарегистрированных маячках. С клиентской стороны, для взаимодействия с устройствами предлагается использовать Nearby Messages API, Places API.

При работе маячки транслируют уникальный код Advertisment ID (Eddystone-UID), к которому можно привязывать вложения (Attachments).

После того как мобильное приложение обнаружило маячок оно делает запрос по его Advertisment ID в Beacons Registry, а ответом получаем это вложение с информацией.

Такой сценарий отлично подходит, когда необходимо сделать не публичное приложение для внутреннего использования.

В свою очередь Google Proximity Beacon API это облачный сервис, который через REST интерфейс позволяет управлять маячками, а точнее связанными с ними данными.

Proximity Beacon API позволяет: регистрировать устройства, присоединять вложения к маячкам, мониторить устройства, получать вложения от маячков.

Основой протокола Proximity Beacon API являются OAuth 2.0 для доступа к Google API, RESTfull Web сервисы, JSON представление данных.

Для работы с Proximity Beacon API необходимо выполнить 5 следующих шагов: создать Google Account, создать проект в консоли разработчика Google, активировать Proximity Beacon API в консоли разработчика Google, получить OAuth 2.0 client ID, получить API KEY.

Для чего нажмите Select a project и выберите проект, который будет использоваться. В левом, боковом меню нажмите на API Manager, затем найдите Proximity Beacon API и установите его статусу значение enabled.

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		40

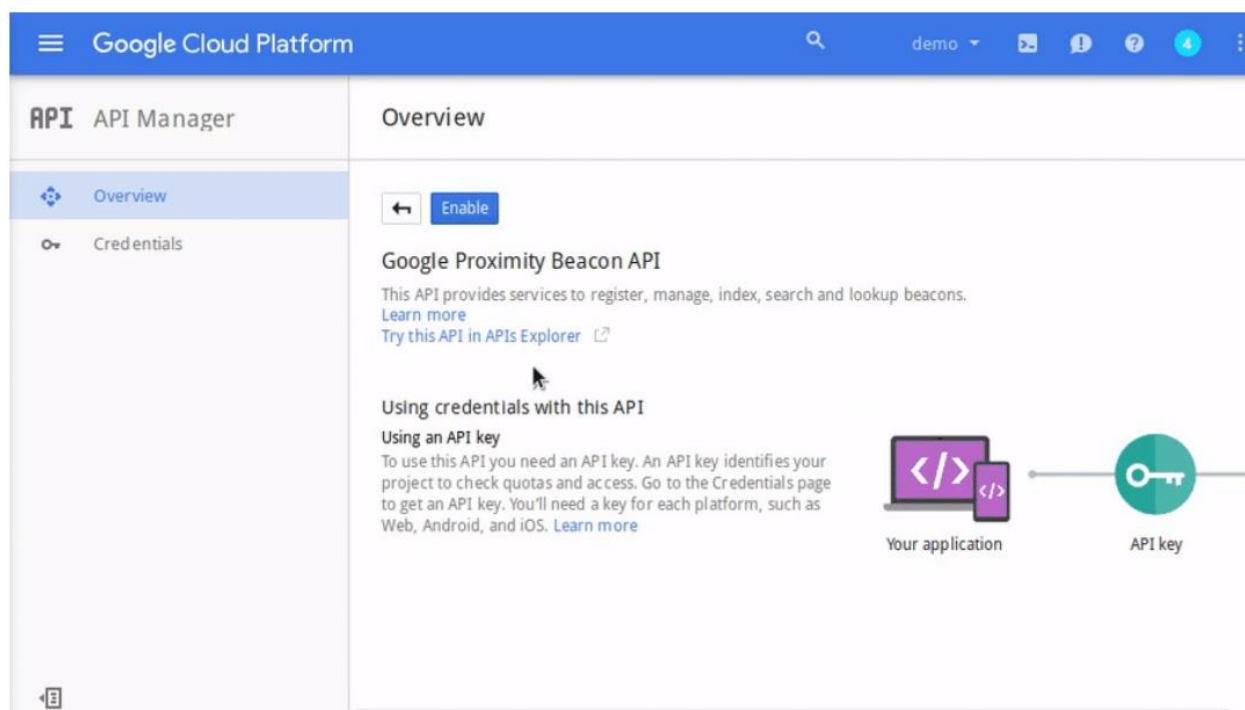


Рисунок 2.8 Регистрация системы в Beacons сервер Google.

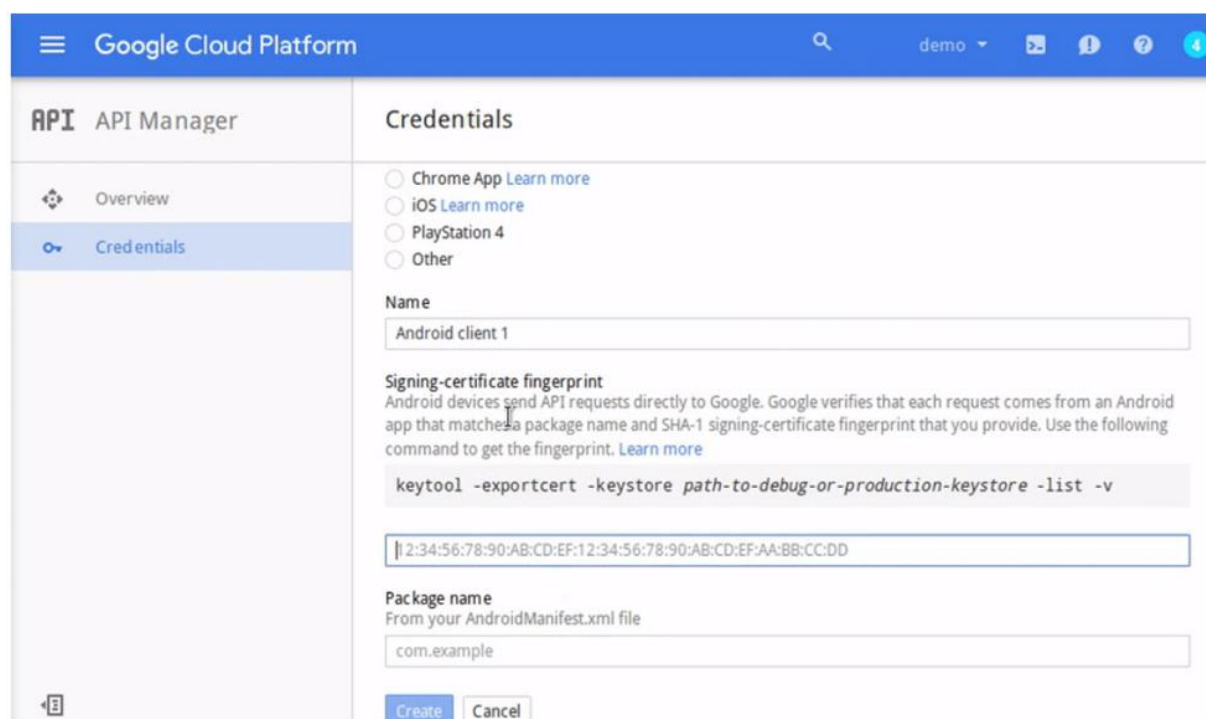


Рисунок 2.9 Регистрация маячка в Beacons сервер (Google).

Далее после получения уникального ключа маяка в системе к нему прикрепляются дополнительные данные, которые будут полезны мобильному приложению.

Запись в Google Beacons Registry будет содержать следующие метаданные: Advertised ID (обязательно), текущий статус (активный, не активный), широта и долгота (пара double значение в градусах), Google Places API Place ID, текстовое описание.

Если регистрация маячка прошла успешно можно проверить его работу в браузере. Для этого копируем выданный маяку URL и пробуем его запустить. В данном случае запрос:

<https://proximitybeacon.googleapis.com/v1beta1/beacons:register>

выдаст результат:

```
{"advertisedId": {  
  "type": "EDDYSTONE",  
  "id": "Fr4Z98nSoW0hgAAAAAAAAAg"  
},  
"status": "ACTIVE",  
"placeId": "ChIJTxaх6NoSkFQRWPvFXI1LypQ",  
"latLng": {  
  "latitude": "47.6693771",  
  "longitude": "-122.1966037"  
},  
"description": "Волонтерская программа Дети весны",  
}
```

Полученный результат показывает, что система облачный сервер Beacons настроен и может работать с устройством, которое зарегистрировано в профиле реализуемой системы. Примеры других REST запросов (см. Приложение Г).

2.5 Выбор электронного маяка

В качестве электронных маяков выбраны модели компании Estimote (рисунок 2.10), т.к. они в отличие от других производителей имеют режимы взаимодействия с внешними устройствами Eddystone, iBeacon, AltBeacon.

- Размер: 56 мм * 41 мм * 18 мм
- Работает с платформами IOS и Android.
- Встроенный пароль сопряжения
- Поддержка настройки собственной конфигурации, включая считывание и изменение UUID и других характеристик сопряжения.
- Настройка мощности
- Контекстные сообщения могут быть запрограммированы из личного кабинета Google.
- Поддержка обновления прошивки через OAD.
- Встроенный порт JTAG для прошивки нужных приложений и работы в качестве автономной системы.
- Руководство, SDK для программирования устройства.



Рисунок 2.10 Электронный маяк

Модель универсальная по применению и использованию стеками программирования. Легко настраивается, эти характеристики стали причиной выбора.

2.6 Реализация работы с маяками в мобильном приложении

Для обнаружения BLE Beacons в ОС Android необходимо подключить внешнюю библиотеку. В этом проекте реализован UID Eddystone и естественным выбором будет библиотека Android Beacon Library, разработанная AltBeacon, обеспечивающая полную поддержку всех фреймов Eddystone. в своей документации AltBeacon упоминает, что любое устройство с Android 4.3+ и набором микросхем Bluetooth Low Energy может обнаруживать маяки, используя данную библиотеку. Пакет `java org.altbeacon.beacon` содержит интерфейсы и классы, необходимые для реализации проекта:

Класс `BeaconConsumer` является интерфейсом для `Activity` и/или `Service`, который взаимодействует с маяками. Интерфейс используется в сочетании с `BeaconManager` и обеспечивает вызов, когда `BeaconService` готов к обнаружению маяка.

Класс `BeaconManager` используется для настройки взаимодействия с маяками. Класс `BeaconManager` включает в себя методы мониторинга и определения маяков. `BeaconManager` требует объект `BeaconConsumer` для исполнения своих методов.

Интерфейс `RangeNotifier` необходим для определения расстояния до маяков. `RangeNotifier` обеспечивает идентификацию и ранжирование видимых маяков. Он реализуется классом `BeaconManager` и вычисляет расстояние с помощью метода: `void didRangeBeaconsInRegion` (Коллекция маяков `<Beacon>`, регион `Region`).

Класс `Region` работает с критериями маяков. Объекты класса могут фильтровать маяки, которые желательно отслеживать или ранжировать, на основе стандарта `Beacon`, `UUID` и других параметров идентификатора.

`Beacon Manager` управляет методами для мониторинга и ранжирования маяков, поэтому требуется переменная типа `private`, которая будет в `Activity`, в котором происходит работа с BLE.

```
private BeaconManager beaconManager;
```

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		44

Вместе с Beacon Manager служба Bluetooth реализована посредством использования интерфейсов BeaconConsumer и RangeNotifier.

```
public class BLEDevice implements BeaconConsumer, RangeNotifier { ... }
```

Благодаря такому подходу служба Bluetooth может быть связана с любым Activity, используя методы, показанные в таблице 2.10.

Таблица 2.10 – Методы для работы с маяками.

Тип возвращаемого значения	Метод	Описание
Void	nBeaconServiceConnect()	Вызывается, когда Activity запущено и готово к отправке команд через BeaconManager.
Boolean	bindService()	Вызывается BeaconManager, чтобы получить контекст действия в программе и отправить команду Bluetooth.
Void	unBindService()	Вызывается BeaconManager, чтобы получить сообщение от Bluetooth.
Void	didRangeBeaconsInRegion()	Вызывается один раз в секунду для расчета расстояния от маяка до мобильного телефона.

Для отображения уникального кода маяка в приложении реализована следующая функция преобразования кода маяка в целочисленный идентификатор:

```
public int identifierToInt (Identifier id) {
```

```

int sum = 0;
String str = id. toString ();
int index = 13;
int p = 0;
while (index > 1 && str. charAt (index) != '0'){
    int pow = ( int ) Math .pow (16.0 ,( double ) p);
    if (str. charAt ( index ) > '0' &&
        str. charAt ( index ) <= '9'){
        sum += ((( int )str. charAt ( index )) -48)*pow;
    } else {
        sum += ((( int )str. charAt ( index )) -87)*pow;
    }
    index --;
    p++;
}
return sum;
}

```

Для работы на клиентской стороне с Bluetooth маячками, является Nearby Messages API. Nearby Messages API это API, которое реализует парадигму publish-subscribe и позволяет разным устройствам публиковать, и подписываться на сообщения, таким образом обмениваться данными.

Для получения сообщений от маячков необходимо на них подписаться, есть два способа как приложение может это сделать: в активном режиме приложения, в ответ на действия пользователя или события или в фоновом режиме, т.е. когда приложение неактивно. В данном приложении реализован первый способ.

Nearby Messages API требует разрешения пользователя на запросы publish() и subscribe(). Поэтому приложение должно проверять, на что пользователь дал согласие и если такое отсутствует, то вызывать диалог запроса разрешений.

Что бы реализовать запрос разрешения у пользователя во время исполнения вашего приложения, необходимо:

- присоединить обработчик result callback к вызовам publish() и subscribe();
- использовать Nearby.Messages.getPermissionStatus(), что бы проверить статус разрешения непосредственно перед вызовом publish() или subscribe().

```
private void handleSuccessfulNearbyResult (Status status) {  
  
    ListBLEDevice BLEDev = new ListBLEDevice;  
  
    Log.i(TAG, "Processing error, status = " + status);  
  
    if (mResolvingError) {  
  
        return; }  
  
    else if (status.hasResolution()) {  
  
        try { mResolvingError = true;  
  
            BLEDev = startResolutionForResult(getActivity(),  
  
                Constants.REQUEST_RESOLVE_ERROR);  
  
        } catch (IntentSender.SendIntentException e) {  
  
            mResolvingError = false;  
  
            Log.i(TAG, "Failed to resolve error status.", e);  
  
        }  
  
        } else {  
  
        if (status.getStatusCode() == CommonStatusCodes.NETWORK_ERROR) {  
  
            Toast.makeText(getActivity().getApplicationContext(),  
  
                "No connectivity, cannot proceed. Fix in 'Settings' and try again.",  
  
                Toast.LENGTH_LONG).show();  
  
        } else {
```

```

Toast.makeText(getActivity().getApplicationContext(), "Unsuccessful: " +
status.getStatusMessage(), Toast.LENGTH_LONG).show();

}

}

```

В результате работы метода приложение получает или список устройств или код об ошибке.

2.7 Проектирование пользовательского интерфейса.

Работа приложения начинается с открытия окна «Вход в систему» (рисунок 2.11), модулями которого являются элементы TextView, EditText, Button.

Вход и регистрация реализованы на базе протокола Auth и библиотеки Google Play Services для Android.

Рисунок 2.11 Вход в систему

Если пароль неверный или пользователь не зарегистрирован в системе ему требуется пройти процесс регистрации (рисунок 2.12 - 2.13).

EventManagment

Управляй своей командой

Имя

Email

Пароль

☒ Я согласен с [Правилами](#) и [Условиями](#)

Создать аккаунт

Уже есть аккаунт? [Войти](#)

Рисунок 2.12 Регистрация в системе

[< Вход](#) [Восстановление пароля](#)

Забыли пароль?

Для восстановления пароля введите свлй
адрес электронной почты

Email

Продолжить

Рисунок 2.13 Регистрация в системе

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		50

После входа в систему пользователь попадает в основное меню программы (рисунок 2.14), в котором он в зависимости от роли имеет доступ к пунктам меню.

EventManagment

ГЛАВНОЕ МЕНЮ

	Календарь событий	>
	Задачи	>
	Команда	>
	Устройства	>
	Профиль пользователя	>

Рисунок 2.14 Основное меню

Календарь событий (рисунок 2.15) позволяет выбрать событие для выполнения заданий по нему.

Рисунок 2.15 Календарь событий

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		51

Волонтерское движение Доброе сердце

Задачи

Забрать подарки из Дома Дружбы
г.Пенза, Проспект Мира, д. 43
☒

Забрать подарки из ООО ТДК
г.Пенза, Проспект Мира, д. 43
☐

Привести группу аниматоров
г.Пенза, Проспект Мира, д. 43
☐

Воспитательная встреча
г.Пенза, Проспект Мира, д. 43
☐

Команда

Администратор
Свободно
☐

Член команды
Свободно
☐

Волонтер 1
Свободно
☐

Волонтер 2
Свободно
☐

Рисунок 2.16 Задачи события

Волонтерское движение Доброе сердце

Задачи

Забрать подарки из Дома Дружбы
г.Пенза, Проспект Мира, д. 43
☒

Забрать подарки из ООО ТДК
г.Пенза, Проспект Мира, д. 43
☐

Привести группу аниматоров
г.Пенза, Проспект Мира, д. 43
☐

Воспитательная встреча
г.Пенза, Проспект Мира, д. 43
☐

Команда

Администратор
Свободно
☐

Член команды
Свободно
☐

Владимир У.
Волонтер 1
☒

Волонтер 2
Свободно
☐

Рисунок 2.17 Выбор задач и своего статуса

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		52

Как только событие определено, можно определить свой статус, выбрав членство в команде.

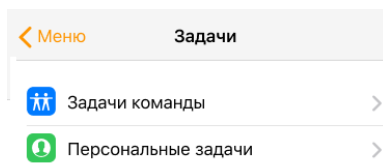
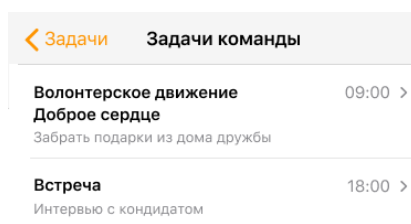


Рисунок 2.18 Просмотр задач

Если определен статус члена команды, то доступна опция список задач команды (рисунок 2.19), в противном случае можно просматривать только персональные задачи (рисунок 2.20), а также возможность посмотреть распределение задач.



Пригласить команду

Рисунок 2.19 Список задач

Забрать подарки из дома дружбы

Адрес

г.Пенза, Проспект Мира, д. 43

Валантёр 1 — Владимир У.

uteev.v@gmail.com

Рисунок 2.20 Персональная задача

Меню

Команда

Администратор — Евгений Т.

evg2d.v@gmail.com

Член команды — Артём С.

santa.v@gmail.com

Валантёр 1 — Владимир У.

uteev.v@gmail.com

Валантёр 2 — Игнат С.

infhdv@gmail.com

Рисунок 2.21 Список команды

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		54

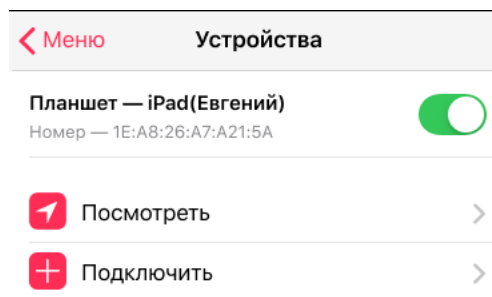


Рисунок 2.22 Доступные устройства

На карте (рисунок 2.23) показаны устройства, которые являются маяками для выполнения задачи.

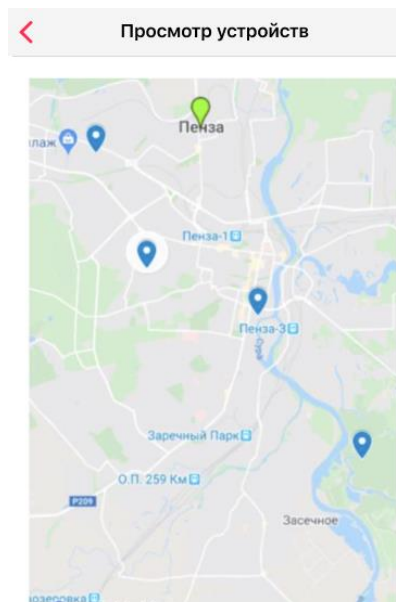


Рисунок 2.23 Карта устройств

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		55

← Меню

Устройства

Планшет — iPad(Евгений)

Номер — 1E:A8:26:A7:A21:5A

Посмотреть

Подключить

Рисунок 2.24 Выбор доступного устройства

←

Подключение устройства

Для подключение устройства его
необходимо отсканировать

Сканировать

Рисунок 2.25 Сообщение об ошибке

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		56

Доступные устройства дают возможность, члену команды считать информацию об устройстве, которое является маяком тех предметов и/или подарков, которые ему требуется взять, так как не всегда волонтерский персонал может визуально определить какие именно подарки требуется взять участнику мероприятия.

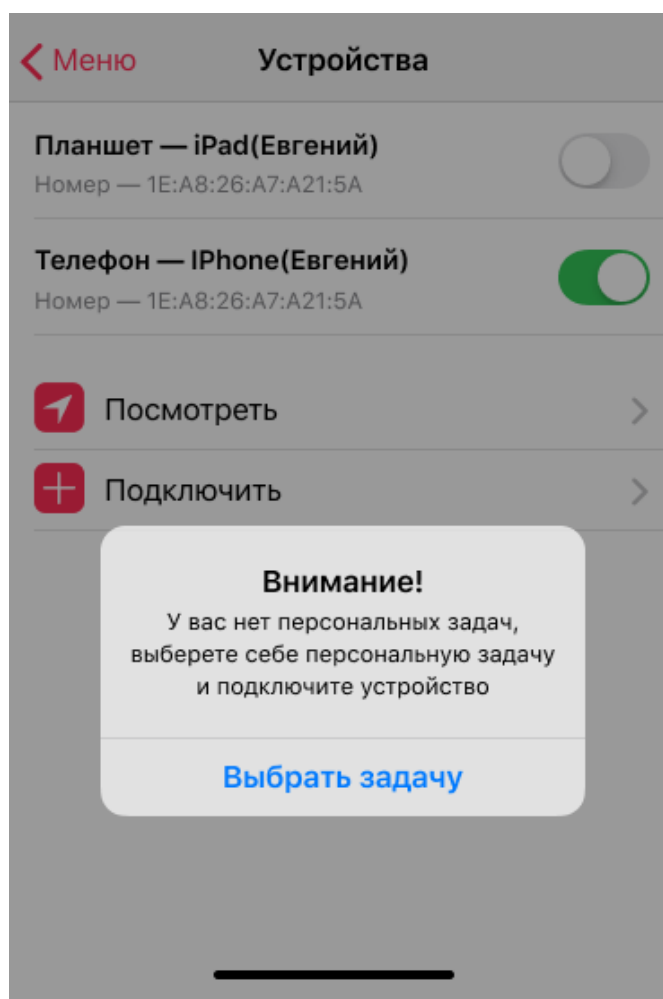


Рисунок 2.26 Сообщение об ошибке

Устройство даже если пользователь его видит, может быть доступно если есть персональное задание, связанное с этим устройством.

В профиле пользователя (рисунок 2.27) можно увидеть свой статус и задачи, которые доступны пользователю, также можно удалить и отказаться от выполнения задачи.

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		57



Владимир У.
utaev.v@gmail.com

Активность

Волонтер 1

Волонтерское движение Доброе сердце



Сменить пароль

Выйти из приложения

Рисунок 2.27 Профиль пользователя.

3. Технология настройки среды реализации проекта

Для того, чтобы писать программу нужна среда разработки. Google рекомендует использовать для этих целей Eclipse с плагином Android Development Tools (ADT).

Для реализации проекта требовались следующие инструменты: Java SDK – JDK, Android SDK, IDE Eclipse, ADT.

Т.к. разработка приложений ведется на Java, нам нужно скачать и установить соответствующее SDK, называемое еще JDK (если, конечно, оно уже не установлено).

Android SDK включает в себя инструменты, необходимые для разработки Android-приложений. создать каталог Android. Крайне желательно, чтобы путь к нему был коротким. Идеально - <имя диска>: \android (у меня это будет f:\android). И в него распакуем архив SDK, получим \android-sdk-windows.

IDE Eclipse среда разработки, где происходит разработка системы. По умолчанию Eclipse не особо годится для разработки Android-приложений. ADT - плагин, который настраивает среду разработки для использования Android SDK и добавляет возможность удобной разработки.

Запускаем Eclipse (\eclipse\eclipse.exe). При первом запуске он попросит указать ему рабочий каталог, где он будет хранить информацию о проектах. Предлагаю опять же не ходить далеко и создать каталог \workspace и указать этот каталог.

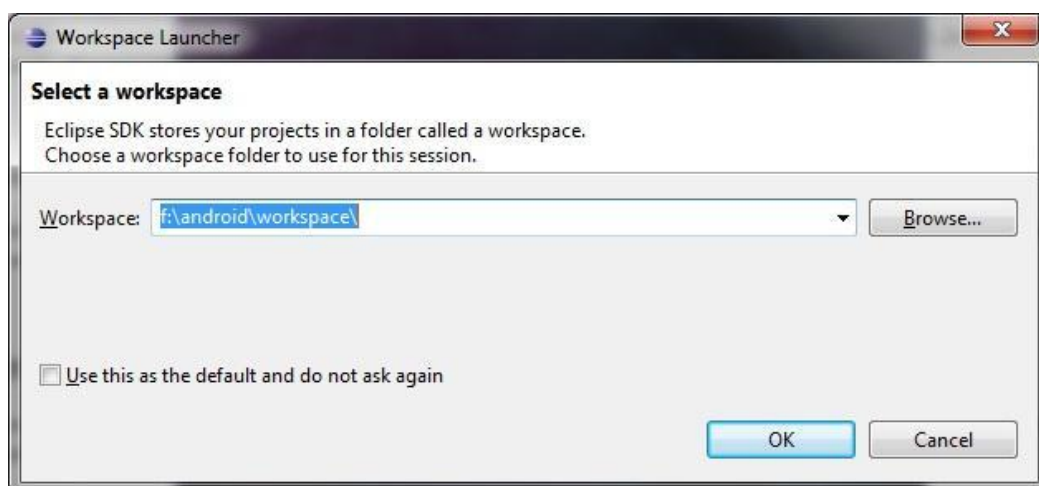


Рисунок 3.1 Запуск IDE Eclipse

Итак, Eclipse запущен. Скачаем ADT плагин. В меню запускаем Help> Install New Software...

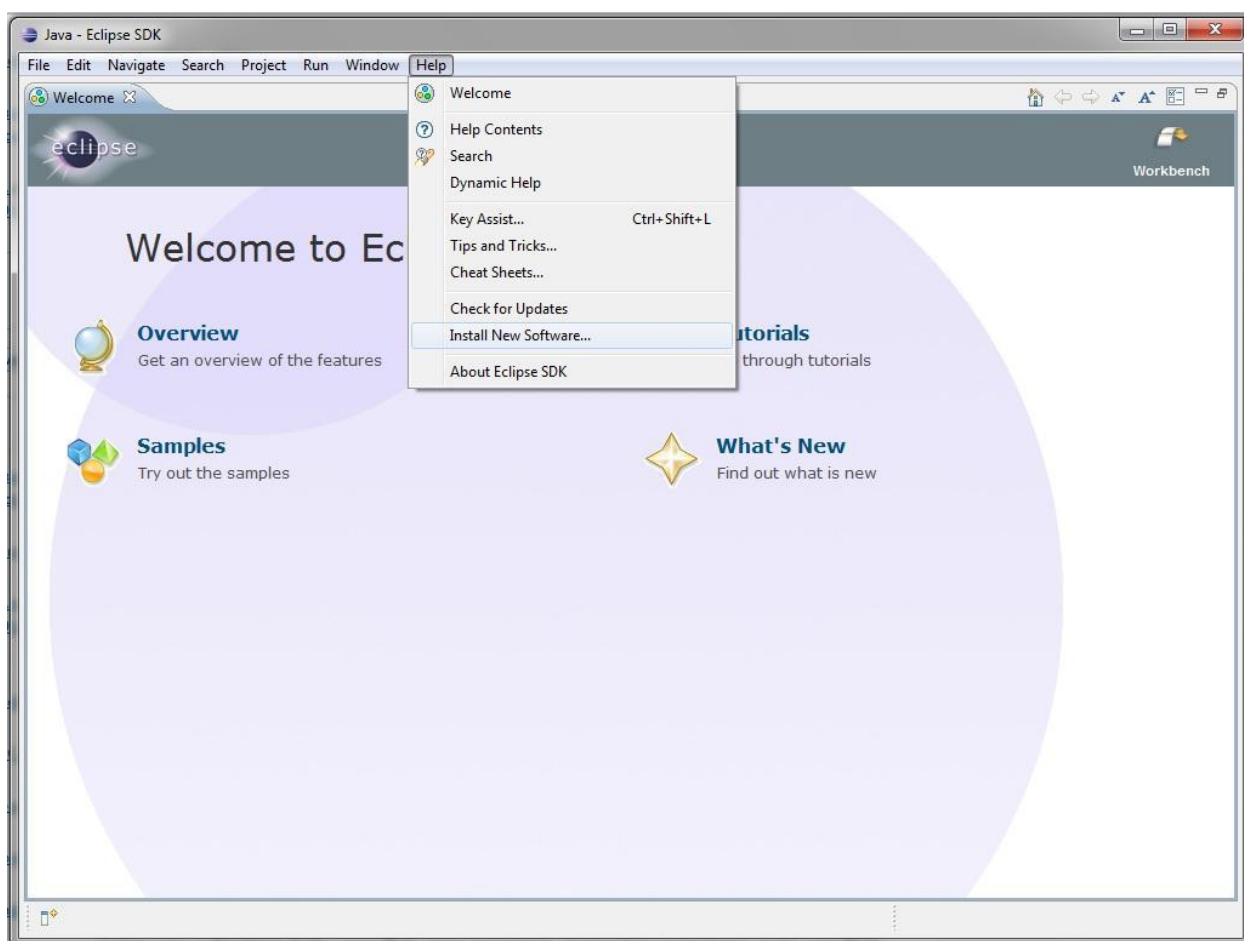


Рисунок 3.2 Основное меню IDE Eclipse

Жмем кнопку Add в правом верхнем углу основного меню IDE Eclipse

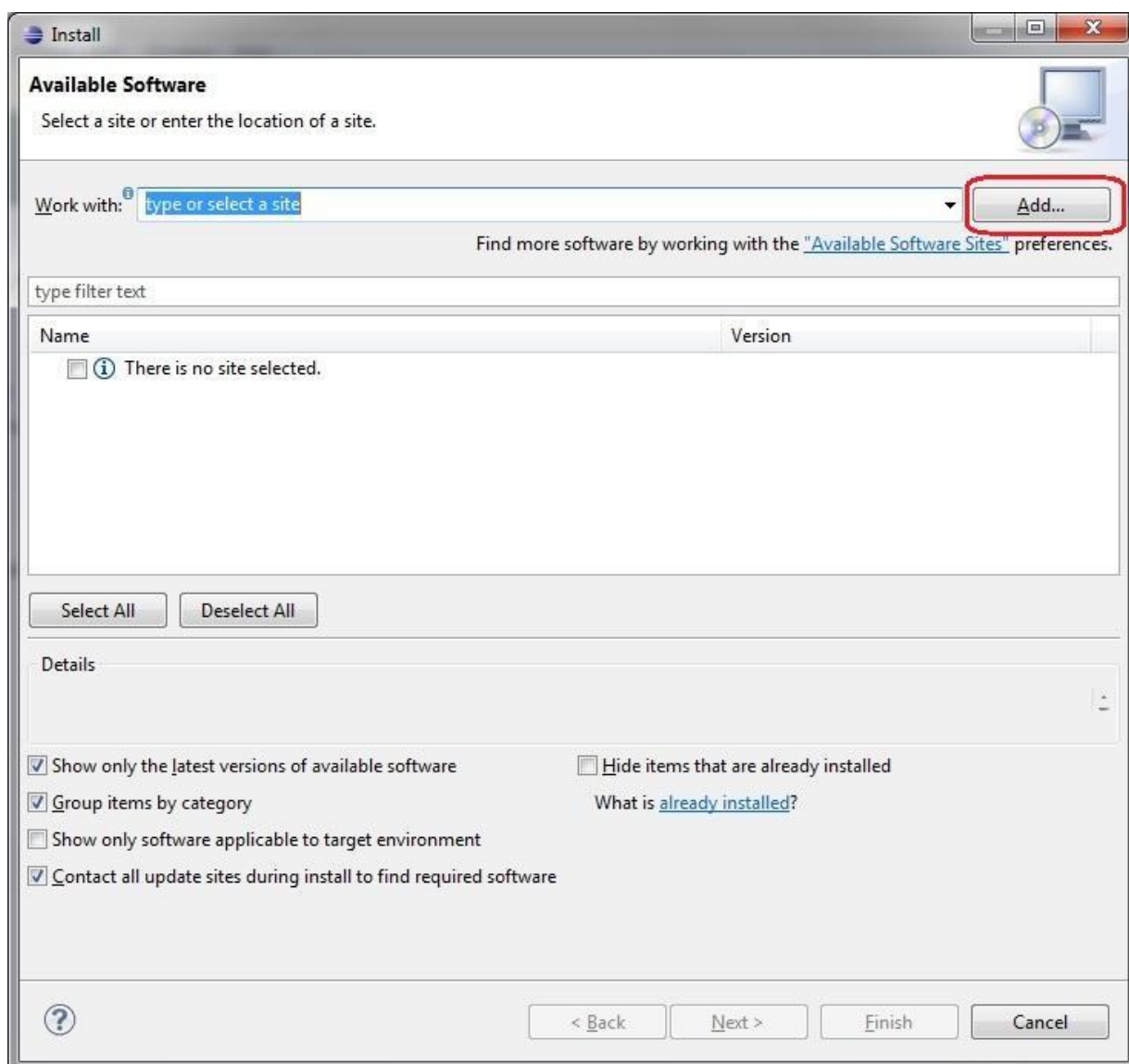


Рисунок 3.3 Добавить компонент

Вводим "ADT Plugin" в поле Name. URL адрес: <https://dl-ssl.google.com/android/eclipse/> - в поле Location.

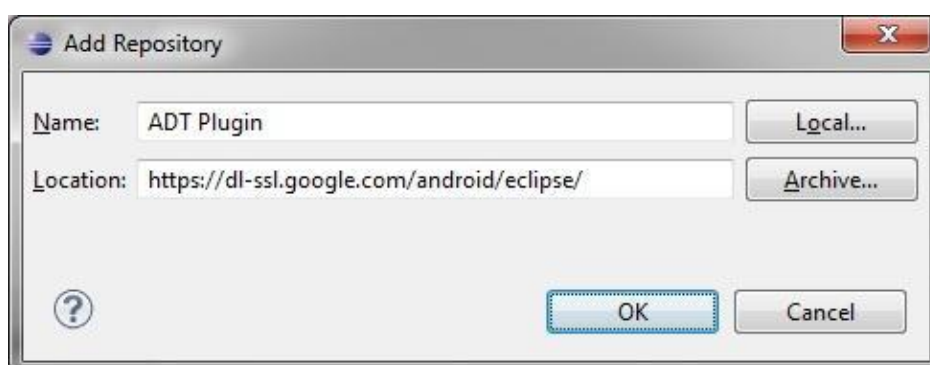


Рисунок 3.4 Настроить плагин

Ставим галку на Developer Tools и жмем Next. Жмем ОК и ждем, пока появится Developer Tools (Если возникают проблемы, используйте http вместо https).

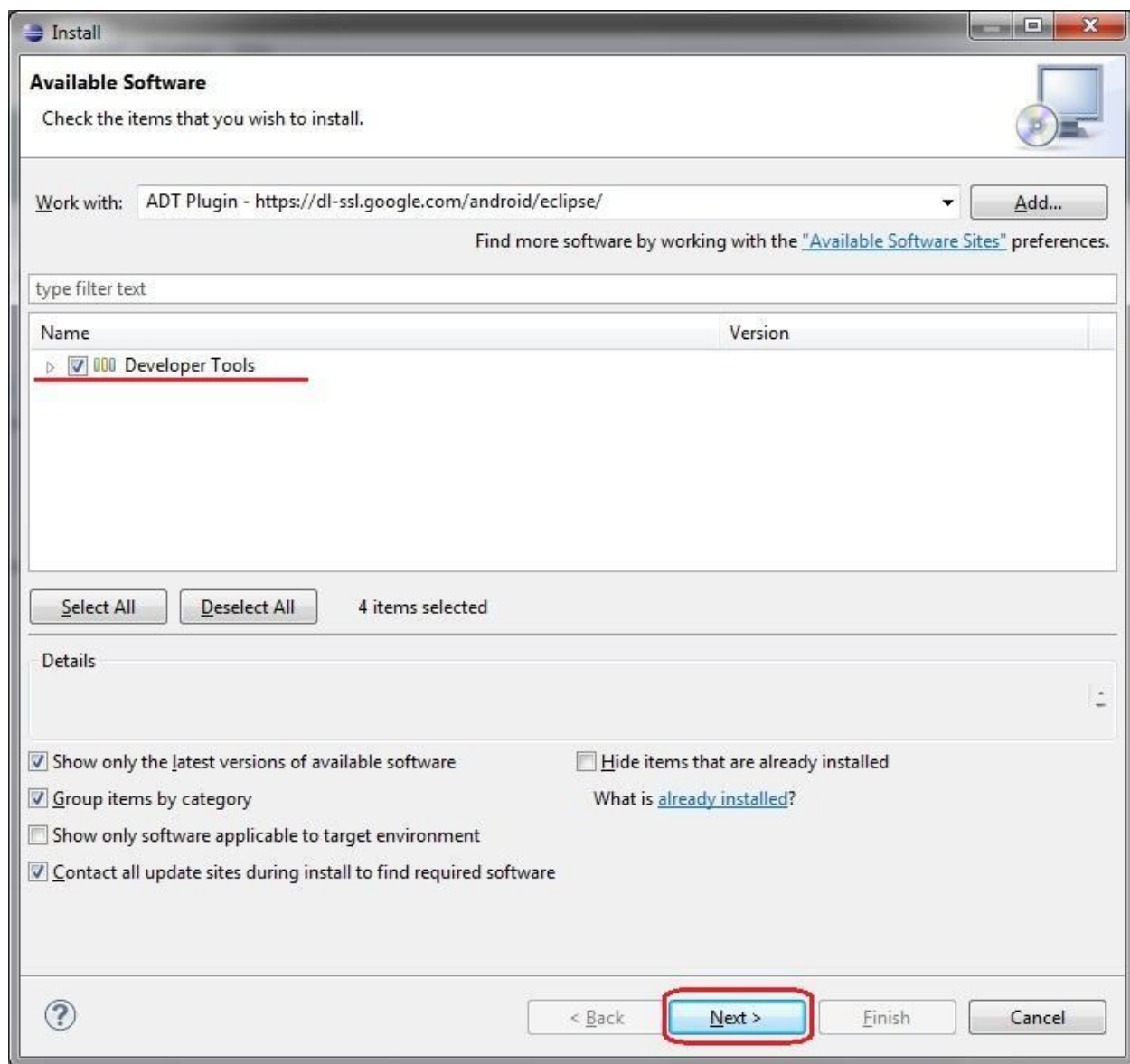


Рисунок 3.5 Ссылка на репозиторий плагина

Видим компоненты, которые будут установлены, жмем снова Next.

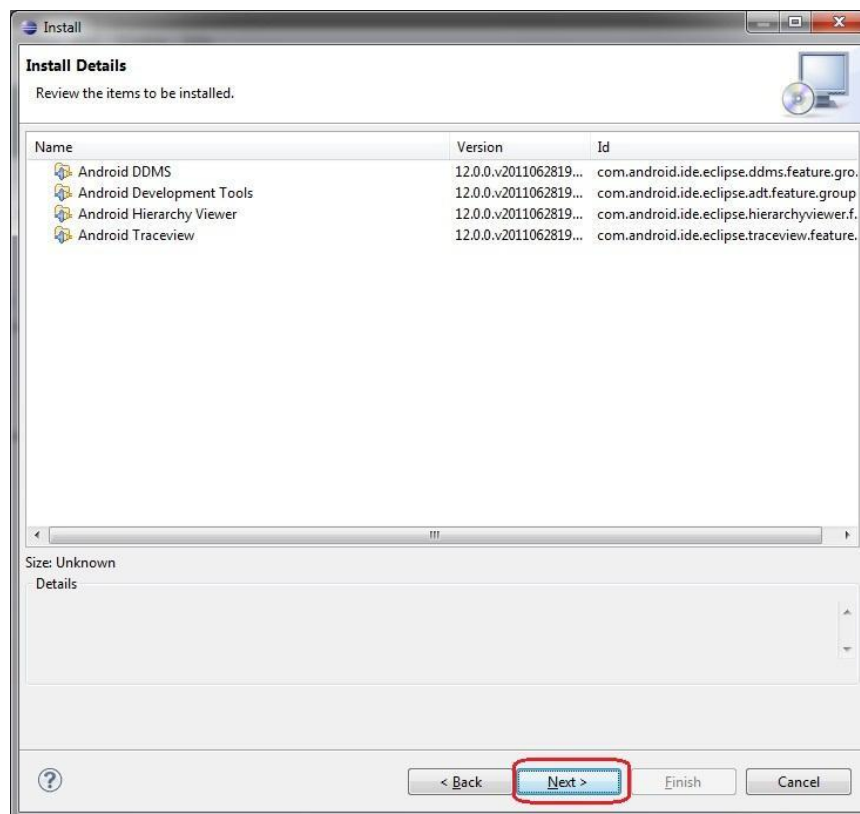


Рисунок 3.6 Доступные плагины репозитория.

Читаем и принимаем лицензионное соглашение и жмем Finish.

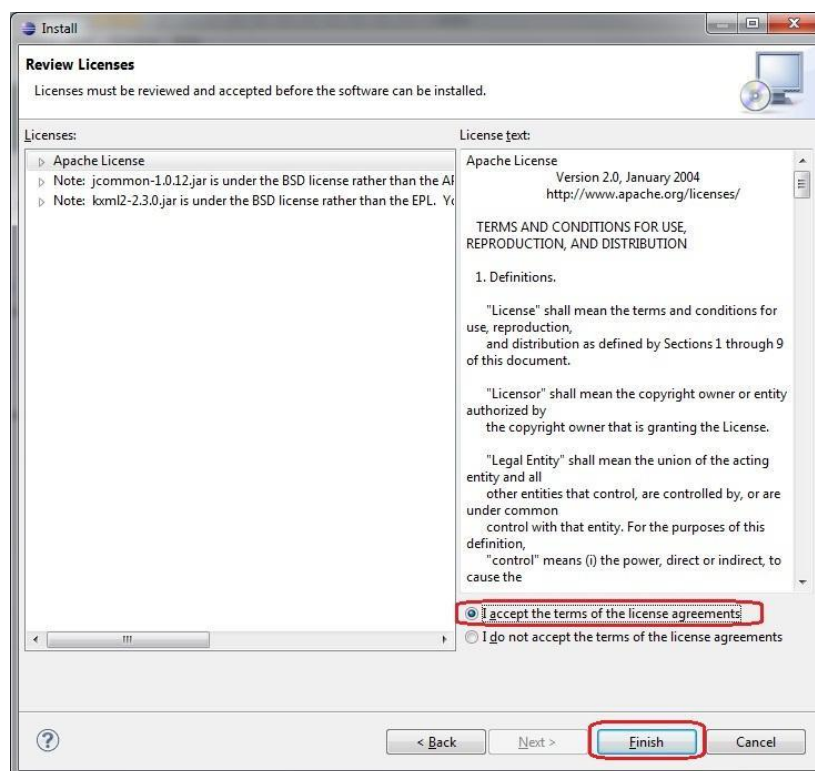


Рисунок 3.7 Доступные плагины репозитория.

Начинается загрузка компонентов. Если появится предупреждение Security warning о том, что the authenticity or validity of the software can't be established, жмите OK.

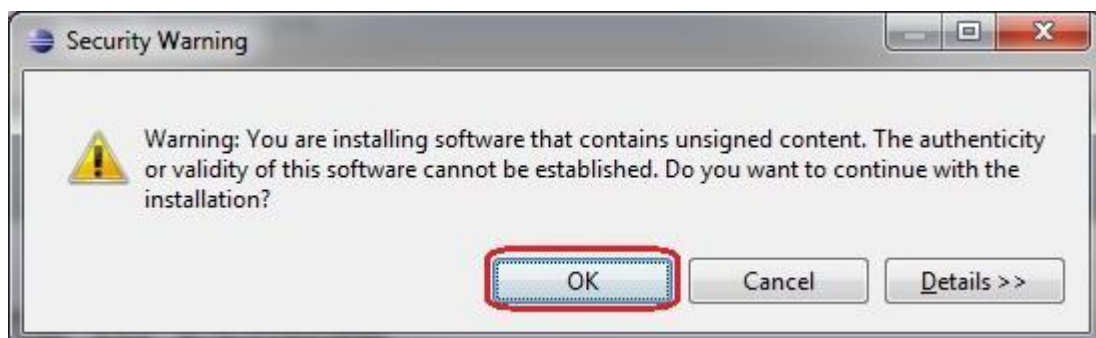


Рисунок 3.8 Предупреждение о безопасности компонентов.

После завершения надо перезапустить Eclipse - Restart Now.



Рисунок 3.9 Перезапуск IDE Eclipse.

После перезапуска Eclipse выдаст такой диалог (рисунок 3.10). Первый пункт нужен для тех, кто не скачал и не распаковал SDK на втором шаге данной технологической цепочки. Eclipse сможет сделать это сам. И дополнительно сразу же скачает последнюю Android-платформу. Также он предлагает скачать платформу версии 2.1, как наиболее поддерживаемую большинством устройств.



Рисунок 3.10 Android IDE Eclipse.

Выберите второй пункт - он позволяет указать, куда распакован SDK (каталог\android-sdk-windows). Жмем Next. И в появившемся окне выбираете, отправлять статистику в google или не отправлять. Скаченное SDK не содержит компонент и предлагают пройти в SDK Manager и срочно закатать. Жмем OK.

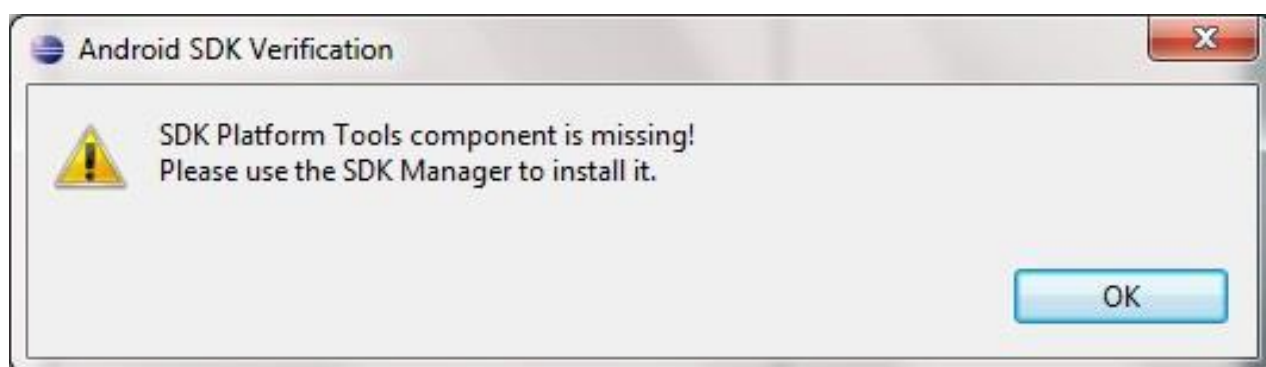


Рисунок 3.11 Окно предупреждения.

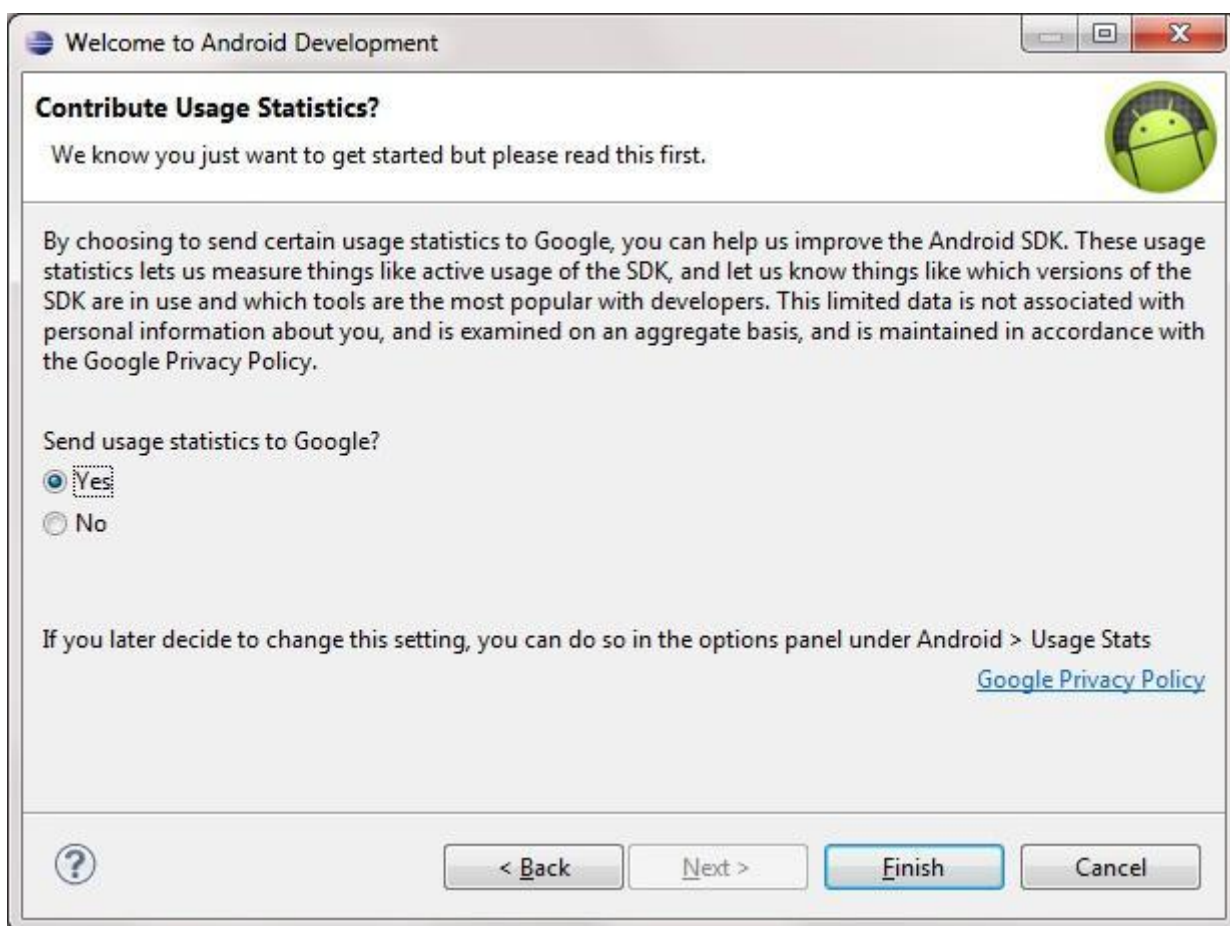


Рисунок 3.12 Статистика IDE Eclipse для google.

Запускаем на исполнение SDK Manager. Тут предлагается выбрать компоненты для загрузки через интернет. В правой части для каждого компонента указано, установлен он уже или нет. Обязательно нужно скачать не установленное содержимое папки Tools. Также в папке Extras (в конце списка) найдите пункт Android Support Library и отметьте его, если он есть. Для платформ версии 4.X выбираем SDK Platform (сама платформа) и ARM EABI v7a System Image (для работы эмулятора). Google APIs by Google Inc. - нужно если собираетесь работать с google-приложениями (Map, Navigation и пр.).

Когда все выбрали - жмем кнопку Install X packages справа снизу. В новом окне подтверждаем, что согласны все это скачать - Accept All.

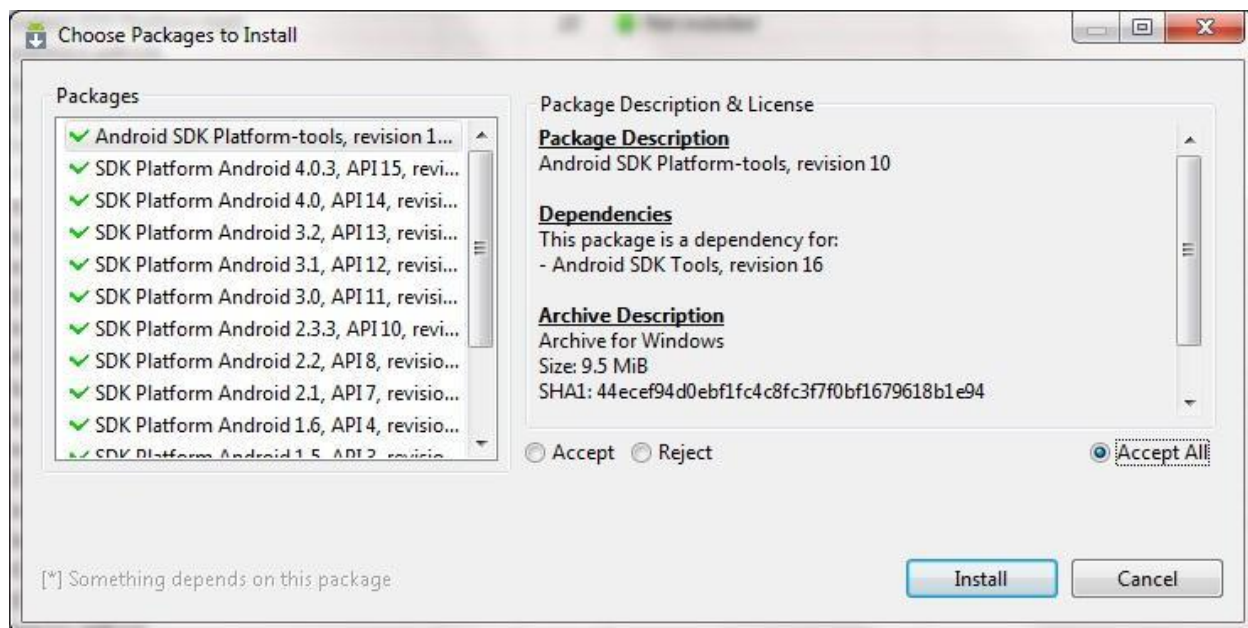


Рисунок 3.12 Компоненты SDK Manager.

После установки всех компонентов, среда осуществит перезапуск и будет готова к разработке приложения.

4. Технико-экономическое обоснование разработки программы

4.1 Введение

Целью технико-экономического обоснования разработки является количественное и качественное доказательство экономической целесообразности усовершенствования программы, а также определение организационно-экономических условий ее эффективного функционирования.

Эффективность программного модуля определяется его качеством и эффективностью процесса разработки и сопровождения. Качество программного изделия определяется тремя составляющими:

- с точки зрения специалиста-пользователя данного программного продукта;
- с позиции использования ресурсов и их оценки;
- по выполнению требований на программное изделие.

Программное изделие должно быть разработано так, чтобы оно выполняло свои функции без лишних затрат ресурсов (оперативной памяти ПК, машинного времени, пропускной способности каналов передачи данных и др. – на стадии функционирования; времени разработки и денежных ресурсов – на стадии использования программного изделия).

Содержание технико-экономического обоснования разработки программы заключается в следующем:

- расчет технико-экономических показателей и выбор базы сравнения;
- определение трудоемкости и стоимости программного обеспечения (ПО);
- расчет цены ПО;

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		68

- расчет капитальных и эксплуатационных затрат на разработку;
- определение показателей финансово-экономической эффективности.

Приведенные в данном разделе дипломной работы результирующие таблицы, перечни данных, позволяют сопоставить результаты разработки и затраты на нее, чтобы сделать вывод об эффективности проекта. Исходные данные для расчета экономических показателей приведены в таблице 5.1.

Таблица 5.1. Исходные данные для расчета экономических показателей

Обозначение	Наименование показателя	Единицы измерения	Значение показателя		
$C_{ЭВМ}$	Стоимость ЭВМ	тыс. руб.	20		
D_M	Среднее количество дней в месяце	дни	22		
ρ_n	Норматив рентабельности	—	0,24		
ω_d	Коэффициент, учитывающий дополнительную заработную плату разработчика программы	—	0,20		
ω_c	Коэффициент, учитывающий начисления органам социального страхования	—	0,268		
ω_n	Коэффициент, учитывающий накладные расходы организации	—	1,35		
q_I	Количество I-задач, решаемых потребителем	зад. год	60		
$t_{M.V.I}$	Время решения I-ой задачи разработанной программой	маш. час	1		
n_{II}	Количество организаций, которые приобретут данную программу	шт.	1г .	2г .	3г .

Продолжение таблицы 5.1

			8	10	15
$Z_{ЭЛ}$	Тариф за 1 кВт/час	руб.	3,44		
T_c	Срок службы разработанной программы	год	3		
$НДС$	Налог на добавленную стоимость	%	20		
T_p	Количество рабочих дней в году	дн	264		
$N_{см}$	Количество смен работы ЭВМ	—	1		
$t_{см}$	Продолжительность смены	ч	8		
α	Простои ЭВМ	%	5		
P	Мощность, потребляемая ЭВМ	кВт	0,4		
$N_{ср}$	Среднее количество ремонтов в год	—	2		
S_d	Стоимость деталей, заменяемых при ремонте	руб.	1300		

4.2 Расчет затрат на разработку программного обеспечения

Суммарные затраты на разработку программы рассчитываются по следующей формуле:

$$S_{РП} = S_{ЗП} + S_{НАК}, \quad (5.1)$$

где:

$S_{ЗП}$ – затраты по заработной плате инженера-программиста;

$S_{НАК}$ – накладные расходы.

Затраты по заработной плате инженера-программиста рассчитываются по формуле:

$$S_{ЗП} = ОЗП \cdot (1 + \omega_c) \cdot (1 + \omega_d) \cdot t_{pi}, \quad (5.2)$$

где:

$OЗП$ – основная заработная плата инженера-программиста за месяц (10000 руб.);

t_{pi} – время, необходимое для разработки программы программистом i -го разряда (чел.-мес.);

ω_d – коэффициент, учитывающий дополнительную заработную плату разработчика программы, в долях к сумме основной заработной платы;

ω_c – коэффициент, учитывающий начисления органам социального страхования на заработную плату разработчика программы, в долях к сумме основной заработной плате разработчика.

Программа разрабатывалась 30 дней, если учесть, что в одном месяце 22 рабочих дня, то:

$$t_{p12} = \frac{30}{22} = 1,36 \text{ (чел. –мес.)} \quad (5.3)$$

Таким образом, затраты по заработной плате инженера-программиста составят:

$$S_{ЗП} = 10000 \cdot (1+0,268) \cdot (1+0,20) \cdot 1,36 = 20693,76 \text{ (руб.)}$$

Накладные затраты рассчитываются с учетом ω_n – коэффициента, определяющего уровень накладных расходов организации по формуле:

$$S_{НАК} = OЗП \cdot \omega_n \cdot t_{p12}, \quad (5.4)$$

$$S_{НАК} = 10000 \cdot 1,35 \cdot 1,36 = 18360 \text{ (руб.)}$$

Таким образом, суммарные затраты на разработку программного обеспечения составляют:

$$S_{РП} = 20693,76 + 18360 = 39053,76 \text{ (руб.)}.$$

4.3 Расчет цены разработанной программы

Оптовая цена разработанной программы определяется по следующей формуле:

$$Z_{\Pi} = S_{\text{РП}} + \Pi, \quad (5.5)$$

где:

Z_{Π} – оптовая цена (цена разработчика) (руб.);

$S_{\text{РП}}$ – суммарные затраты на разработку программы (руб.);

Π – прибыль, рассчитанная по формуле:

$$\Pi = \rho_{\text{н}} \cdot S_{\text{РП}}, \quad (5.6)$$

где:

$\rho_{\text{н}}$ – норматив рентабельности, учитывающий прибыль организации, разрабатывающей данную программу в долях ко всем затратам данной организации на разработку программы.

Итак,

$$Z_{\Pi} = S_{\text{РП}} \cdot (1 + \rho_{\text{н}}), \quad (5.7)$$

$$Z_{\Pi} = 39053,76 \cdot (1 + 0,24) = 48426,7 \text{ (руб.)}$$

Розничная цена программы рассчитывается с учетом налога на добавленную стоимость ($\text{НДС} = 20\%$) по формуле:

$$Z_{\text{Пр}} = Z_{\Pi} + \text{НДС} = Z_{\Pi} + \frac{Z_{\Pi} \cdot 20}{100} = Z_{\Pi} \cdot (1 + 0,20), \quad (5.8)$$

$$Z_{\text{Пр}} = 48426,7 \cdot (1 + 0,20) = 58112,04 \text{ (руб.)}$$

Выручка от продаж при условии $n_{\text{п}}$ – количество организаций, желающих приобрести программу, составит:

$$B = Z_{\text{Пр}} \cdot n_{\text{п}}, \quad (5.9)$$

$$B_1 = 58112,04 \cdot 8 = 464896,32 \text{ (руб.)}$$

$$B_2 = 58112,04 \cdot 10 = 581120,40 \text{ (руб.)}$$

$$B_3 = 58112,04 \cdot 15 = 871680,60 \text{ (руб.)}$$

4.4 Расчет капитальных вложений

Капиталовложения, связанные с работой ЭВМ рассчитываются по формуле:

$$K_{\text{ЭВМ}} = C_{\text{ЭВМ}} + S_3, \quad (5.10)$$

где:

$C_{\text{ЭВМ}}$ – стоимость ЭВМ (руб.);

S_3 – стоимость запасных частей (руб.);

Итак, произведем расчет коэффициентов входящих в формулу расчета величины капиталовложений:

$$S_3 = 0,15 \cdot C_{\text{ЭВМ}}; \quad (5.11)$$

$$S_3 = 0,15 \cdot 20000 = 3000 \text{ (руб.)};$$

Капиталовложения в ЭВМ составляют:

$$K_{\text{ЭВМ}} = 20000 + 3000 = 23000 \text{ (руб.)}.$$

4.5 Расчет эксплуатационных расходов

Эксплуатационные расходы на ЭВМ рассчитываются по формуле:

$$E = (T_{\text{М.В.}} \cdot e_{\text{ч}}) + \frac{Z_{\text{П}}}{T_{\text{С}}}, \quad (5.12)$$

где:

$T_{\text{М.В.}}$ – машинное время для решения задач с помощью разработанной программы, (маш. час/год);

$e_{\text{ч}}$ – эксплуатационные расходы, приходящиеся на 1 час работы ЭВМ;

$Z_{\text{П}}$ – цена, по которой продается программа (руб.);

$T_{\text{С}}$ – срок службы программы (г).

Полезный фонд времени работы ЭВМ рассчитывается по формуле:

$$T_{\text{ПОЛ}} = T_{\text{ОБЩ}} \cdot t_{\text{СМ}} \cdot N_{\text{СМ}} \cdot \left(1 - \frac{\alpha}{100}\right), \quad (5.13)$$

где:

$T_{\text{ОБЩ}}$ – общий фонд времени работы ЭВМ (дни); $T_{\text{ОБЩ}} = T_{\text{Р}}$;

$N_{\text{СМ}}$ – количество смен работы ЭВМ;

$t_{\text{СМ}}$ – время одного рабочего дня (час);

α – простои ЭВМ (в % от общего фонда времени работы ЭВМ).

Полезный фонд времени работы ЭВМ получим:

$$T_{\text{ПОЛ}} = 264 \cdot 8 \cdot 1 \cdot \left(1 - \frac{5}{100}\right) = 2006,4 \text{ (Маш. час /год)}.$$

Машинное время для решения задач с помощью данной программы

рассчитывается по формуле:

$$T_{\text{М.В.}} = q_I \cdot t_{\text{М.В.И}}, \quad (5.14)$$

где:

q_I – количество I-задач, решаемых потребителем в год(шт.);

$t_{\text{М.В.И}}$ – время решения I-ой задачи, разработанной программой (Маш. час).

$$T_{\text{М.В.}} = 60 \cdot 1 = 60 \text{ (Маш. час / год)}.$$

Эксплуатационные расходы, приходящиеся на 1 час работы ЭВМ, оцениваются по формуле:

$$e_{\text{ч}} = \frac{(A_{\text{О}} + S_{\text{ЗП}} + S_{\text{ЭЛ}} + R_{\text{РМ}})}{T_{\text{ПОЛ}}}, \quad (5.15)$$

где:

$A_{\text{О}}$ – амортизационные отчисления (руб.);

$S_{\text{ЗП}}$ – затраты по заработной плате инженера в год (руб./год);

$S_{\text{ЭЛ}}$ – стоимость потребляемой энергии (руб.);

$R_{\text{РМ}}$ – затраты на ремонт ЭВМ (руб.);

$T_{\text{ПОЛ}}$ – полезный годовой фонд работы ЭВМ, (маш. час/год).

Амортизационные отчисления рассчитываются с учетом нормы амортизации ($a_{\text{н}} = 12,5 \%$);

$$A_{\text{О}} = a_{\text{н}} \cdot K_{\text{ЭВМ}} = 0,125 \cdot 23000 = 2875,0 \text{ (руб.)}. \quad (5.16)$$

Затраты по заработной плате инженера за год рассчитывается по формуле:

$$S_{ЗП} = (1 + \omega_c) \cdot (1 + \omega_d) \cdot ОЗП \cdot 12, \quad (5.17)$$

где:

ω_c – коэффициент, учитывающий начисления органам социального страхования на заработную плату разработчика программы, в долях к сумме основной заработной плате разработчика.

ω_d – коэффициент, учитывающий дополнительную заработную плату разработчика программы, в долях к сумме основной заработной платы;

$ОЗП$ – основная заработная плата инженера за месяц 8-го разряда (10000 руб.);

Рассчитываем годовые затраты по заработной плате и социальным отчислениям для инженера:

$$S_{ЗП} = (1 + 0,268) \cdot (1 + 0,20) \cdot 10000 \cdot 12 = 182592 \text{ (руб./ год)}$$

Стоимость потребляемой энергии оценивается по формуле:

$$S_{ЭЛ} = P \cdot T_{\text{ПОЛ}} \cdot Z_{ЭЛ}, \quad (5.18)$$

где: P – мощность, потребляемая ЭВМ (кВт);

$T_{\text{ПОЛ}}$ – полезный годовой фонд работы ЭВМ (маш. час/год);

$Z_{ЭЛ}$ – тариф за 1 кВт/час (руб. /кВт).

Итак, произведем расчет стоимости потребляемой энергии:

$$S_{ЭЛ} = 0,4 \cdot 2006,4 \cdot 3,44 = 2760,81 \text{ (руб.)}.$$

Затраты на ремонт ЭВМ вычисляются по формуле:

$$R_{\text{РМ}} = N_{\text{СР}} \cdot S_{\text{д}}, \quad (5.19)$$

где: $N_{\text{СР}}$ – среднее количество ремонтов в год;

S_d – стоимость деталей заменяемых при одном ремонте, в среднем.

$$R_{PM} = 2 \cdot 1300 = 2600 \text{ (руб.)}.$$

Произведем вычисление эксплуатационных расходов, приходящихся на 1 час работы ЭВМ:

$$e_{\text{ч}} = \frac{(2875 + 182592 + 2760,81 + 2600)}{2006,4} = 95,12 \text{ (руб./час)}.$$

Далее вычислим эксплуатационные расходы на ЭВМ:

$$E = (60 \cdot 95,12) + \frac{48426,7}{3} = 21849,43 \text{ (руб.)}.$$

4.6 Определение расходов на интернет трафик

Для облегчения работы программиста и повышения его производительности в проекте предусмотрено использование глобальной сети интернет.

В настоящее время на пензенском рынке интернет очень много провайдеров предлагающие свои услуги. Цены, предлагаемые основными провайдерами, сведем в таблицу 4.6.1, для анализа и экономически целесообразного выбора поставщика услуги по интернет трафику.

Таблица 5.2 Таблица интернет трафиков и провайдеров

Название провайдера	Предлагаемый тариф	Стоимость тарифа (руб.)	Скорость доступа (Мбит/с)
Ростелеком	Пакет-Для впечатлений	699	100
ПАО «МТС»	ФИТ Интернет	450	100
Дом.РУ	Скорость света	550	100

Т.к. вышеперечисленные провайдеры предлагают без лимитные интернет трафики с одинаковой скоростью доступа в сеть интернет, то экономически целесообразно выбрать трафик с самой низкой стоимостью. Выбираем ПАО "МТС", тариф ФИТ Интернет, с абонентской платой в месяц 450 рублей.

4.7 Заключение

Обобщенные технико-экономические показатели разработки программы сведены в таблицу 5.7.1.

Таблица 5.3 Итоговая таблица

Показатель	Значение
Капитальные вложения (руб.)	23000
Эксплуатационные расходы (руб.)	21849,43
Затраты на разработку ПО (руб.)	39053,76
Затраты на интернет трафик (руб.)	450
Затраты на заработную плату программисту (руб.)	20693,76

5. Безопасность жизнедеятельности. Эргономика рабочего места программиста.

5.1 Характеристика условий труда программиста

Технический прогресс глобально поменял условия производственной деятельности сотрудников умственного труда. Их труд стал более интенсивнее, требующим больших затрат умственной, эмоциональной и физической энергии. Это потребовало системного решения проблем эргономики, гигиены и организации труда, регламентации режимов труда и отдыха.

В настоящее время компьютеры широко применяется во всех областях деятельности человека. При работе с компьютером, человек подвергается воздействию ряда вредных электромагнитных полей (ВЧ, УВЧ и СВЧ), шума и вибрации, статического электричества и др.

Работа с компьютером характеризуется значительным умственным напряжением и нервно-эмоциональной нагрузкой операторов, высокой напряженностью зрительной работы и большой нагрузкой на мышцы рук при работе с клавиатурой. Большое значение имеет рациональная конструкция и расположение элементов рабочего места, что важно для поддержания оптимальной рабочей позы программиста.

В процессе работы с компьютером необходимо соблюдать правильный режим труда и отдыха. В противном случае у персонала отмечаются болезненное напряжение зрительного аппарата с появлением жалоб на неудовлетворенность работой, головные боли, раздражительность, нарушение сна, усталость и болезненные ощущения в глазах, в пояснице, в области шеи и руках.

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		79

Одним из критериев, по которому определяется «пригодность» рабочего места для постоянного пребывания сотрудников, является эргономические требования к рабочему месту и освещенность рабочего места программиста.

5.2 Эргономические требования к рабочему месту программиста

Проектирование рабочих мест, снабженных мониторами, относится к числу важных проблем эргономического проектирования в области вычислительной техники.

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

Эргономическими аспектами проектирования рабочих мест программиста, являются:

1. высота рабочей поверхности,
2. размеры пространства для ног,
3. требования к расположению документов на рабочем месте (наличие и размеры подставки для документов, возможность различного размещения документов, расстояние от глаз пользователя до экрана, документа, клавиатуры и т.д.),
4. характеристики рабочего кресла,
5. требования к поверхности рабочего стола,
6. регулируемость элементов рабочего места.

Главными элементами рабочего места программиста являются стол и кресло.

Основным рабочим положением является положение сидя.

Рабочая поза сидя вызывает минимальное утомление программиста.

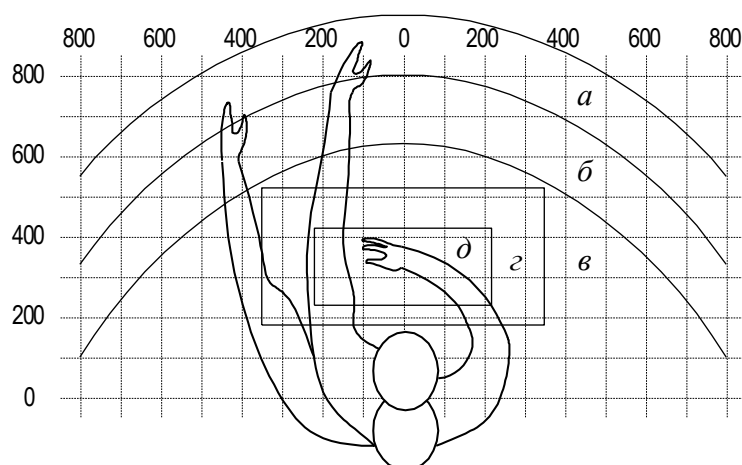
Рациональная планировка рабочего места предусматривает четкий порядок и постоянство размещения предметов, средств труда и документации.

То, что требуется для выполнения работ чаще, расположено в зоне легкой досягаемости рабочего пространства.

Моторное поле - пространство рабочего места, в котором могут осуществляться двигательные действия человека.

Максимальная зона досягаемости рук — это часть моторного поля рабочего места, ограниченного дугами, описываемыми максимально вытянутыми руками при движении их в плечевом суставе.

Оптимальная зона - часть моторного поля рабочего места, ограниченного дугами, описываемыми предплечьями при движении в локтевых суставах с опорой в точке локтя и с относительно неподвижным плечом.



- а - зона максимальной досягаемости;
- б - зона досягаемости пальцев при вытянутой руке;
- в - зона легкой досягаемости ладони;
- г - оптимальное пространство для грубой ручной работы;

Рисунок 6.1 Зоны досягаемости рук в горизонтальной

Оптимальное размещение предметов труда и документации в зонах досягаемости:

Монитор размещается в зоне а (в центре);

Системный блок размещается в предусмотренной нише стола;

Клавиатура - в зоне г/д;

«Мышь» - в зоне в справа;

Сканер в зоне, а/б (слева);

Принтер находится в зоне а (справа);

Документация необходимая при работе - в зоне легкой досягаемости ладони;

В выдвижных ящиках стола - литература, неиспользуемая постоянно.

На рис. 6.2 показан пример размещения основных и периферийных составляющих ПК на рабочем столе программиста.

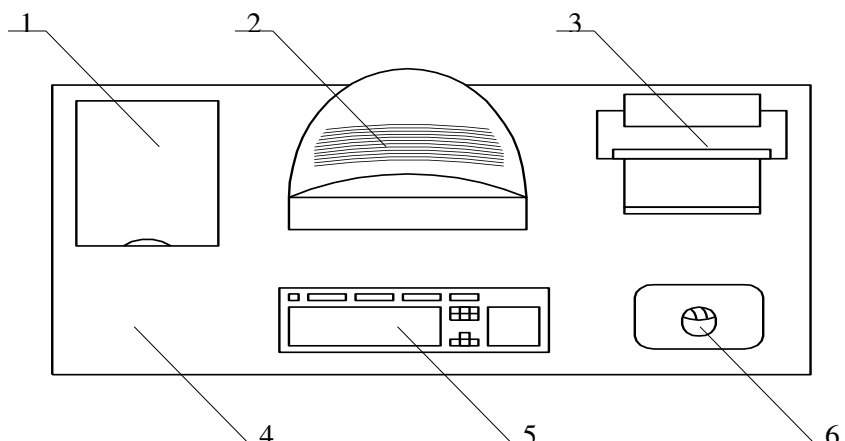


Рисунок 6.2 Размещение основных и периферийных составляющих ПК

1 – сканер, 2 – монитор, 3 – принтер, 4 – поверхность рабочего стола,
5 – клавиатура, 6 – «мышь».

Для комфортной работы стол должен удовлетворять следующим условиям:

1. Высота стола должна быть выбрана с учетом возможности сидеть свободно, в удобной позе, при необходимости опираясь на подлокотники;
2. Нижняя часть стола должна быть сконструирована так, чтобы программист мог удобно сидеть, не был вынужден поджимать ноги;
3. Поверхность стола должна обладать свойствами, исключающими появление бликов в поле зрения программиста;
4. Конструкция стола должна предусматривать наличие выдвижных ящиков (не менее 3 для хранения документации, канцелярских принадлежностей).
5. Высота рабочей поверхности рекомендуется в пределах 680-760 мм. Высота поверхности, на которую устанавливается клавиатура, должна быть около 650 мм.

Большое значение придается характеристикам рабочего кресла. Так, рекомендуемая высота сиденья над уровнем пола находится в пределах 420-550 мм. Поверхность сиденья мягкая, передний край закругленный, а угол наклона спинки - регулируемый.

Необходимо предусматривать при проектировании возможность различного размещения документов: сбоку от монитора, между монитором и клавиатурой и т.п. Кроме того, в случаях, когда монитор имеет низкое качество изображения, например заметны мелькания, расстояние от глаз до экрана делают больше (около 700 мм), чем расстояние от глаза до документа (300-450 мм). Вообще при высоком качестве изображения на видеотерминале расстояние от глаз пользователя до экрана, документа и клавиатуры может быть равным.

Положение экрана определяется:

1. расстоянием считывания (0,6...0,7 м);

2. углом считывания, направлением взгляда на 20° ниже горизонтали к центру экрана, причем экран перпендикулярен этому направлению. Должна также предусматриваться возможность регулирования экрана:
3. по высоте $+3$ см;
4. по наклону от -10° до $+20^\circ$ относительно вертикали;
5. в левом и правом направлениях.

Большое значение также придается правильной рабочей позе пользователя. При неудобной рабочей позе могут появиться боли в мышцах, суставах и сухожилиях. Требования к рабочей позе пользователя монитора следующие:

1. голова не должна быть наклонена более чем на 20° ,
2. плечи должны быть расслаблены,
3. локти - под углом $80^\circ \dots 100^\circ$,
4. предплечья и кисти рук - в горизонтальном положении.

Причина неправильной позы пользователей обусловлена следующими факторами: нет хорошей подставки для документов, клавиатура находится слишком высоко, а документы - низко, некуда положить руки и кисти, недостаточно пространство для ног.

В целях преодоления указанных недостатков даются общие рекомендации: лучше передвижная клавиатура; должны быть предусмотрены специальные приспособления для регулирования высоты стола, клавиатуры и монитора, а также подставка для рук.

Существенное значение для производительной и качественной работы на компьютере имеют размеры знаков, плотность их размещения, контраст и соотношение яркостей символов и фона экрана. Если расстояние от глаз оператора до экрана дисплея составляет $60 \dots 80$ см, то высота знака должна быть не менее 3 мм, оптимальное соотношение ширины и высоты знака составляет 3:4, а расстояние между знаками – $15 \dots 20\%$ их высоты.

Соотношение яркости фона экрана и символов - от 1:2 до 1:15.

Во время пользования компьютером медики советуют устанавливать монитор на расстоянии 50-60 см от глаз. Специалисты также считают, что верхняя часть видеодисплея должна быть на уровне глаз или чуть ниже. Когда человек смотрит прямо перед собой, его глаза открываются шире, чем когда он смотрит вниз. За счет этого площадь обзора значительно увеличивается, вызывая обезвоживание глаз. К тому же если экран установлен высоко, а глаза широко открыты, нарушается функция моргания. Это значит, что глаза не закрываются полностью, не омываются слезной жидкостью, не получают достаточного увлажнения, что приводит к их быстрой утомляемости.

Создание благоприятных условий труда и правильное эстетическое оформление рабочих мест на производстве имеет большое значение, как для облегчения труда, так и для повышения его привлекательности, положительно влияющей на производительность труда.

5.3 Освещенность рабочего места программиста

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

Недостаточность освещения приводит к напряжению зрения, ослабляет внимание, приводит к наступлению преждевременной утомленности.

Чрезмерно яркое освещение вызывает ослепление, раздражение в глазах.

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

Существует три вида освещения - естественное, искусственное и совмещенное (естественное и искусственное вместе).

Естественное освещение - освещение помещений дневным светом, проникающим через световые проемы в наружных ограждающих конструкциях помещений. Естественное освещение характеризуется тем, что меняется в широких пределах в зависимости от времени дня, времени года, характера области и ряда других факторов.

Искусственное освещение применяется при работе в темное время суток и днем, когда не удастся обеспечить нормированные значения коэффициента естественного освещения (пасмурная погода, короткий световой день). Освещение, при котором недостаточное по нормам естественное освещение дополняется искусственным, называется совмещенным освещением.

Искусственное освещение подразделяется на рабочее, аварийное, эвакуационное, охранное. Рабочее освещение, в свою очередь, может быть общим или комбинированным. Общее - освещение, при котором светильники размещаются в верхней зоне помещения равномерно или применительно к расположению оборудования. Комбинированное - освещение, при котором к общему добавляется местное освещение.

Согласно СНиП 23-05-2010 в помещениях вычислительных центров необходимо применить систему комбинированного освещения.

При выполнении работ требующих высокой зрительной точности (наименьший размер объекта различения 0,3...0,5 мм), величина коэффициента естественного освещения (КЕО) должна быть не ниже 1,5%, а при зрительной

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		86

работе средней точности (наименьший размер объекта различения 0,5...1,0 мм) КЕО должен быть не ниже 1,0%.

В качестве источников искусственного освещения обычно используются люминесцентные лампы типа ЛБ или ДРЛ, которые попарно объединяются в светильники, которые должны располагаться над рабочими поверхностями равномерно.

Требования к освещенности в помещениях, где установлены компьютеры, следующие: при выполнении зрительных работ высокой точности, общая освещенность должна составлять 300 лк, а комбинированная - 750 лк; аналогичные требования при выполнении работ средней точности - 200 и 300 лк соответственно.

Кроме того, все поле зрения должно быть освещено достаточно равномерно – это основное гигиеническое требование. Иными словами, степень освещения помещения и яркость экрана компьютера должны быть примерно одинаковыми, т.к. яркий свет в районе периферийного зрения значительно увеличивает напряженность глаз и, как следствие, приводит к их быстрой утомляемости.

5.4 Расчет освещения рабочего места программиста

Расчет освещенности рабочего места сводится к выбору системы освещения, определению необходимого числа светильников, их типа и размещения. Исходя из этого, рассчитаем параметры искусственного освещения.

Обычно искусственное освещение выполняется посредством электрических источников света двух видов: ламп накаливания и люминесцентных ламп. Будем использовать люминесцентные лампы, которые по сравнению с лампами накаливания имеют ряд существенных преимуществ:

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		87

1. по спектральному составу света они близки к дневному, естественному свету;
2. обладают более высоким КПД (в 1,5-2 раза выше, чем КПД ламп накаливания);
3. обладают повышенной светоотдачей (в 3-4 раза выше, чем у ламп накаливания);
4. более длительный срок службы.

Расчет освещения производится для комнаты площадью 15 м², ширина которой 5 м, высота - 3 м. Воспользуемся методом светового потока.

Для определения количества светильников определим световой поток, падающий на поверхность по формуле:

$$F = \frac{E \cdot K \cdot S \cdot Z}{n}, \text{ где}$$

F - рассчитываемый световой поток, Лм;

E - нормированная минимальная освещенность, Лк (определяется по таблице). Работу программиста, в соответствии с этой таблицей, можно отнести к разряду точных работ, следовательно, минимальная

освещенность будет $E = 300 \text{ Лк}$;

S - площадь освещаемого помещения (в нашем случае $S = 15 \text{ м}^2$);

Z - отношение средней освещенности к минимальной (обычно принимается равным 1,1...1,2, пусть $Z = 1,1$);

K - коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации (его значение зависит от типа помещения и характера проводимых в нем работ и в нашем случае $K = 1,5$);

n - коэффициент использования, (выражается отношением светового

потока, падающего на расчетную поверхность, к суммарному потоку всех ламп и исчисляется в долях единицы; зависит от характеристик светильника, размеров помещения, окраски стен и потолка, характеризующих коэффициентами отражения от стен (P_C) и потолка (P_{Π}), значение коэффициентов P_C и P_{Π} были указаны выше: $P_C = 40\%$, $P_{\Pi} = 60\%$. Значение n определим по таблице коэффициентов использования

различных светильников. Для этого вычислим индекс помещения по формуле:

$$I = \frac{S}{h(A+B)}, \text{ где}$$

S - площадь помещения, $S = 15 \text{ м}^2$;

h - расчетная высота подвеса, $h = 2.92 \text{ м}$;

A - ширина помещения, $A = 3 \text{ м}$;

B - длина помещения, $B = 5 \text{ м}$.

Подставив значения получаем:

$$I = \frac{15}{2,92 \cdot (3+5)} = 0,64$$

Зная индекс помещения I , по таблице находим $n = 0,22$

Подставим все значения в формулу для определения светового потока F :

$$F = \frac{300 \cdot 1,5 \cdot 15 \cdot 1,1}{0,22} = 33750 \text{ Лм}$$

Для освещения выбираем люминесцентные лампы типа ЛБ40-1, световой поток которых $F = 4320 \text{ Лк}$.

Рассчитаем необходимое количество ламп по формуле:

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		89

$$N = \frac{F}{F_{\text{л}}}$$

N - определяемое число ламп;

F - световой поток, $F = 33750$ Лм;

$F_{\text{л}}$ - световой поток лампы, $F_{\text{л}} = 4320$ Лм.

$$N = \frac{33750}{4320} = 8 \text{ шт.}$$

При выборе осветительных приборов используем светильники типа ОД.
Каждый светильник комплектуется двумя лампами.

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		90

Заключение

В ходе выполнения дипломного проекта была разработана программное обеспечение, позволяющее команде организовать участие в мероприятиях типа волонтерских движений и хакатонов. Приложение работает в экосистеме Google и позволяет участнику выполнять поставленные задачи благодаря возможности отслеживать ход задания с помощью электронных маяков.

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		91

Список используемых источников

1. Петров А., Даркин С. Программирование под Android. Издательство Питер, 2017.
2. Грин К. С., Пронин О. Н., Усов В. И. Введение в разработку приложений для мобильных платформ. Издательство ВВМ, 2019.
3. Дейтел П. Android для программистов: создаем приложения. Издательство Питер, 2016.
4. Голощапов А.Л. Google Android. Создание приложений для смартфонов и планшетных ПК. Издательство Питер 2017.
5. Мельникова О.М. Смартфоны на Android. Издательство Эксмо, 2018.
6. Герман М.В. Разработка мобильных приложений под Android. УЦ «Специалист» при МГТУ им. Н. Э. Баумана, 2019.
7. John Wiley & Sons. Reto Meier Professional Android 4 Application Development. Wrox, 2015

ПРИЛОЖЕНИЕ А

(обязательное)

ЛИСТИНГ ПРОГРАММЫ

					<i>ПГУ 1.090501.14.001 ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		93

MainActivity.java

```
package com.google.android.gms.nearby.messages.samples.hellobeacons;

import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.widget.RelativeLayout;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.nearby.Nearby;
import com.google.android.gms.nearby.messages.MessagesOptions;
import com.google.android.gms.nearby.messages.NearbyPermissions;
public class MainActivity extends AppCompatActivity {
    private static final String TAG = MainActivity.class.getSimpleName();
    private GoogleApiClient mGoogleApiClient;
    private RelativeLayout mContainer;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        mContainer = (RelativeLayout) findViewById(R.id.main_activity_container);
    }
    @Override
    protected void onResume() {
        super.onResume();
    }
    @Override
    protected void onPause() {
        super.onPause();
    }
    @Override
    protected void onSaveInstanceState(Bundle outState) {
        super.onSaveInstanceState(outState);
    }
}
```

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		94

BackgroundSubscribeIntentService.java

```
package com.google.android.gms.nearby.messages.samples.hellobeacons;

import android.app.IntentService;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.support.v4.app.NotificationCompat;
import android.text.TextUtils;
import com.google.android.gms.nearby.Nearby;
import com.google.android.gms.nearby.messages.Message;
import com.google.android.gms.nearby.messages.MessageListener;
import java.util.List;

public class BackgroundSubscribeIntentService extends IntentService {

    public BackgroundSubscribeIntentService() {
        super("BackgroundSubscribeIntentService");
    }

    @Override
    public void onCreate() {
        super.onCreate();
    }

    @Override
    protected void onHandleIntent(Intent intent) {
    }

}
```

StartupActivity.java

```
package io.upnext.beaconcontrol.app;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import io.upnext.beaconcontrol.app.config.BeaconControlConfiguration;

public class StartupActivity extends Activity {

    public static Intent getIntent(Context context) {
        return new Intent(context, StartupActivity.class);
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        BeaconControlConfiguration beaconControlConfiguration = new BeaconControlConfiguration(this);

        if (beaconControlConfiguration.isLoggedIn()) {
            startActivity(MainActivity.getIntent(this));
        }
    }
}
```

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		95

```

    } else {
        startActivity(LoginActivity.getIntent(this));
    }
}
}

```

LoginErrorDialog.java

```

package io.upnext.beaconcontrol.app;

import android.app.Dialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.app.DialogFragment;
import android.support.v7.app.AlertDialog;

public class LoginErrorDialogFragment extends DialogFragment {

    public static DialogFragment newInstance() {
        return new LoginErrorDialogFragment();
    }

    @NonNull
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        return new AlertDialog.Builder(getContext())
            .setTitle(R.string.login_screen_error_dialog_title)
            .setMessage(R.string.login_screen_error_dialog_message)
            .setPositiveButton(R.string.login_screen_error_dialog_positive_button, new DialogInterface.OnClickListener() {

                @Override
                public void onClick(DialogInterface dialogInterface, int i) {
                    ((LoginActivity) getActivity()).login();
                }
            })
            .setNegativeButton(R.string.login_screen_error_dialog_negative_button, new DialogInterface.OnClickListener() {

                @Override
                public void onClick(DialogInterface dialogInterface, int i) {
                    dismiss();
                }
            })
            .create();
    }
}

```

BeaconAction.java

```
package io.upnext.beaconcontrol.app.actions;

import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.Locale;
import io.upnext.beaconcontrol.app.R;
import io.upnext.beaconcontrol.app.actions.BeaconActionsAdapter.ActionViewHolder;

public class BeaconActionsAdapter extends RecyclerView.Adapter<ActionViewHolder> {

    private final List<BeaconAction> actions = new ArrayList<>();

    public void setActions(List<BeaconAction> newActions) {
        actions.clear();
        actions.addAll(newActions);
        notifyDataSetChanged();
    }

    @Override
    public ActionViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        return new ActionViewHolder(LayoutInflater.from(parent.getContext()).inflate(R.layout.beacon_action_list_item, parent, false));
    }

    @Override
    public void onBindViewHolder(ActionViewHolder holder, int position) {
        holder.populate(actions.get(position));
    }

    @Override
    public int getItemCount() {
        return actions.size();
    }

    @Override
    public long getItemId(int position) {
        return actions.get(position).timestamp;
    }

    public static class ActionViewHolder extends RecyclerView.ViewHolder {

        private static final DateFormat ACTION_DATE_FORMAT = new SimpleDateFormat("yyyy.MM.dd HH:mm:ss",
        Locale.getDefault());

        private TextView actionName;

        private TextView actionTimestamp;
```

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		97

```

public ActionViewHolder(View itemView) {
    super(itemView);
    actionName = (TextView) itemView.findViewById(R.id.action_name);
    actionTimestamp = (TextView) itemView.findViewById(R.id.action_timestamp);
}

public void populate(BeaconAction action) {
    actionName.setText(action.actionName);
    actionTimestamp.setText(ACTION_DATE_FORMAT.format(new Date(action.timestamp)));
}
}
}

```

ControlConfiguration.java

```

package io.upnext.beaconcontrol.app.config;
import android.content.Context;
import android.content.SharedPreferences;
public class BeaconControlConfiguration {
    private static final String SHARED_PREFERENCES_KEY = "io.upnext.beaconcontrol.sampleapp.settings.CONFIGURATION";
    private static final String PREFERENCE_KEY_CLIENT_ID = "client_id";
    private static final String PREFERENCE_KEY_CLIENT_SECRET = "client_secret";
    private static final String PREFERENCE_KEY_USER_ID = "user_id";
    private static final String DEFAULT_USER_ID = "sampleapp_user";
    private final SharedPreferences preferences;
    public BeaconControlConfiguration(Context context) {
        preferences = context.getSharedPreferences(SHARED_PREFERENCES_KEY, Context.MODE_PRIVATE);
    }
    public void setClientId(String clientId) {
        preferences.edit().putString(PREFERENCE_KEY_CLIENT_ID, clientId).apply();
    }
    public String getClientId() {
        if (!preferences.contains(PREFERENCE_KEY_CLIENT_ID)) {
            throw new IllegalStateException("Client id requested, but unavailable in preferences.");
        }
        return preferences.getString(PREFERENCE_KEY_CLIENT_ID, "");
    }
    public void setClientSecret(String clientSecret) {
        preferences.edit().putString(PREFERENCE_KEY_CLIENT_SECRET, clientSecret).apply();
    }
    public String getClientSecret() {
        if (!preferences.contains(PREFERENCE_KEY_CLIENT_SECRET)) {
            throw new IllegalStateException("Client secret requested, but unavailable in preferences.");
        }
        return preferences.getString(PREFERENCE_KEY_CLIENT_SECRET, "");
    }
}

```

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		98

```

public boolean isLoggedIn() {
    return preferences.contains(PREFERENCE_KEY_CLIENT_SECRET);
}

public String getUserId() {
    return preferences.getString(PREFERENCE_KEY_USER_ID, DEFAULT_USER_ID);
}

public void clear() {
    preferences.edit().clear().apply();
}
}

```

HttpListener.java

```

import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.provider.Settings;
import android.support.annotation.NonNull;
import android.support.v4.app.DialogFragment;
import android.support.v7.app.AlertDialog;

public class LocationPermissionExplainingDialogFragment extends DialogFragment {

    public static final String URL_SCHEME_PACKAGE = "package";

    public static DialogFragment newInstance() {
        return new LocationPermissionExplainingDialogFragment();
    }

    @NonNull
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        return new AlertDialog.Builder(getContext())
            .setTitle(R.string.dialog_location_permission_required_title)
            .setMessage(R.string.dialog_location_permission_required_message)
            .setPositiveButton(R.string.dialog_location_permission_required_button_positive, new OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int i) {
                    showApplicationSettings();
                }
            })
            .setNegativeButton(R.string.dialog_location_permission_required_button_negative, new OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int i) {
                    getActivity().finish();
                }
            })
            .create();
    }
}

```

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		99

```

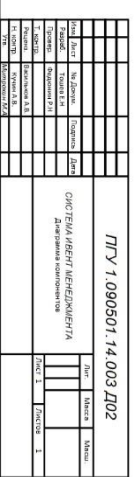
private void showApplicationSettings() {
    Intent intent = new Intent();
    intent.setAction(Settings.ACTION_APPLICATION_DETAILS_SETTINGS);
    Uri uri = Uri.fromParts(Uri.SHEME_PACKAGE, getContext().getPackageName(), null);
    intent.setData(uri);
    startActivity(intent);
}
}

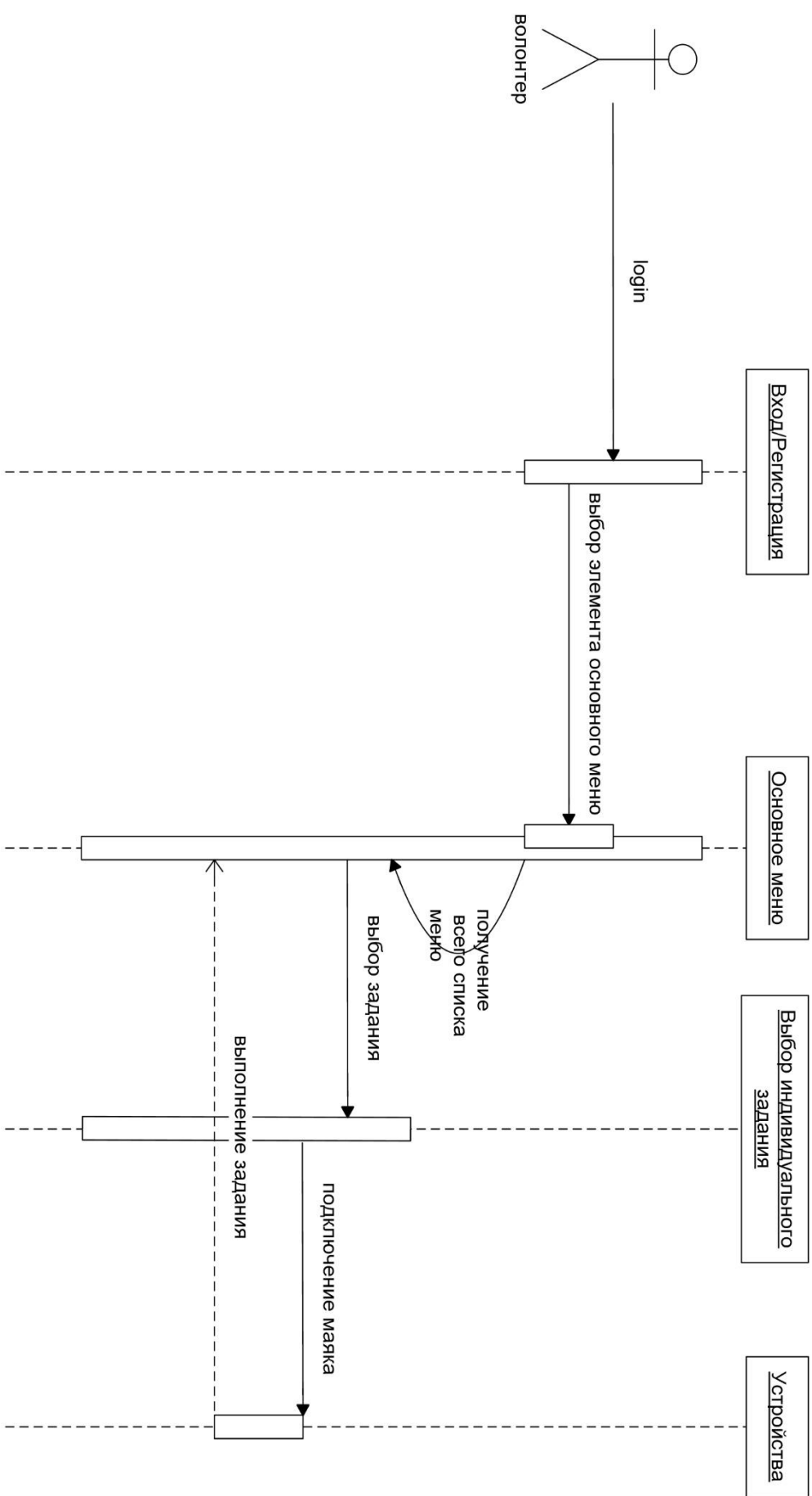
```

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		100

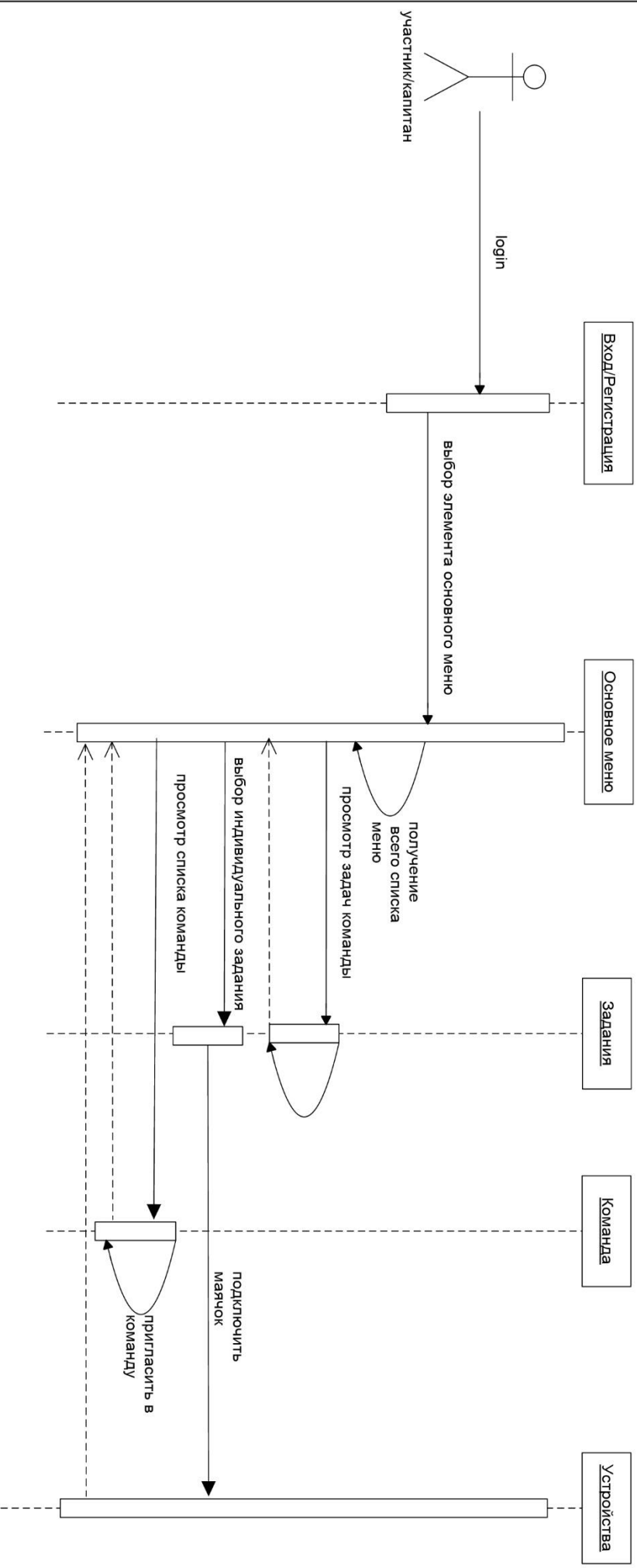
ПРИЛОЖЕНИЕ Б
(обязательное)
СХЕМЫ ПРОГРАММЫ

					ПГУ 1.090501.14.001 ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		101

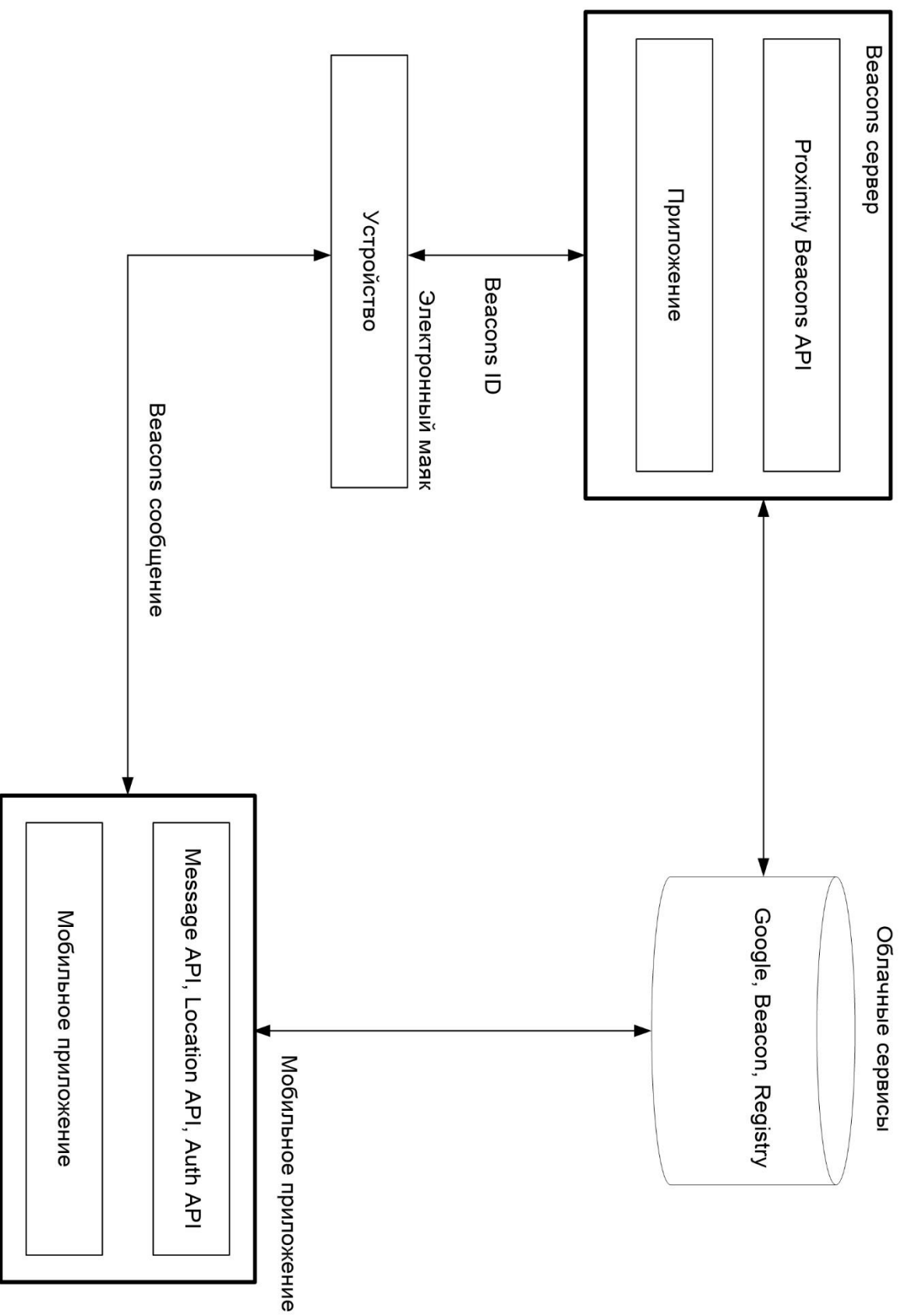




ПТУ 1.090501.14.004 Д03									
Фамилия	Имя	Отчество	Подпись	Дата	СИСТЕМА АВТОГРАФИЧЕСКОГО ДИСТАНЦИОННОГО ПОДКЛЮЧЕНИЯ				
Иванов	Иван	Иванович							
Петров	Петр	Петрович							
Сидоров	Сидор	Сидорович							
Тихонов	Тихон	Тихонович			Подпись				
Левченко	Левченко	Левченко							
Иванов	Иван	Иванович			Подпись				
Петров	Петр	Петрович							
Сидоров	Сидор	Сидорович			Подпись				
Тихонов	Тихон	Тихонович							
Левченко	Левченко	Левченко			Подпись				
Иванов	Иван	Иванович							



ПТУ 1.090501.14.005 Д03									
Фамилия	Имя	Отчество	Дата рождения	Пол	Возраст	Видеокарта	Память	Диск	Процессор
Иванов	Иван	Иванович	1990-01-01	М	28	16GB	8GB	1TB	Intel Core i7
Петров	Петр	Петрович	1991-02-02	М	27	16GB	8GB	1TB	Intel Core i7
Сидоров	Сидор	Сидорович	1992-03-03	М	26	16GB	8GB	1TB	Intel Core i7
Кузнецов	Кузнец	Кузнецович	1993-04-04	М	25	16GB	8GB	1TB	Intel Core i7
Левченко	Левченко	Левченко	1994-05-05	М	24	16GB	8GB	1TB	Intel Core i7
Смирнов	Смирнов	Смирнов	1995-06-06	М	23	16GB	8GB	1TB	Intel Core i7
Васильев	Васильев	Васильев	1996-07-07	М	22	16GB	8GB	1TB	Intel Core i7
Попов	Попов	Попов	1997-08-08	М	21	16GB	8GB	1TB	Intel Core i7
Морозов	Морозов	Морозов	1998-09-09	М	20	16GB	8GB	1TB	Intel Core i7
Иванов	Иванов	Иванов	1999-10-10	М	19	16GB	8GB	1TB	Intel Core i7
Петров	Петров	Петров	2000-11-11	М	18	16GB	8GB	1TB	Intel Core i7
Сидоров	Сидоров	Сидоров	2001-12-12	М	17	16GB	8GB	1TB	Intel Core i7
Кузнецов	Кузнецов	Кузнецов	2002-01-13	М	16	16GB	8GB	1TB	Intel Core i7
Левченко	Левченко	Левченко	2003-02-14	М	15	16GB	8GB	1TB	Intel Core i7
Смирнов	Смирнов	Смирнов	2004-03-15	М	14	16GB	8GB	1TB	Intel Core i7
Васильев	Васильев	Васильев	2005-04-16	М	13	16GB	8GB	1TB	Intel Core i7
Попов	Попов	Попов	2006-05-17	М	12	16GB	8GB	1TB	Intel Core i7
Морозов	Морозов	Морозов	2007-06-18	М	11	16GB	8GB	1TB	Intel Core i7
Иванов	Иванов	Иванов	2008-07-19	М	10	16GB	8GB	1TB	Intel Core i7
Петров	Петров	Петров	2009-08-20	М	9	16GB	8GB	1TB	Intel Core i7
Сидоров	Сидоров	Сидоров	2010-09-21	М	8	16GB	8GB	1TB	Intel Core i7
Кузнецов	Кузнецов	Кузнецов	2011-10-22	М	7	16GB	8GB	1TB	Intel Core i7
Левченко	Левченко	Левченко	2012-11-23	М	6	16GB	8GB	1TB	Intel Core i7
Смирнов	Смирнов	Смирнов	2013-12-24	М	5	16GB	8GB	1TB	Intel Core i7
Васильев	Васильев	Васильев	2014-01-25	М	4	16GB	8GB	1TB	Intel Core i7
Попов	Попов	Попов	2015-02-26	М	3	16GB	8GB	1TB	Intel Core i7
Морозов	Морозов	Морозов	2016-03-27	М	2	16GB	8GB	1TB	Intel Core i7
Иванов	Иванов	Иванов	2017-04-28	М	1	16GB	8GB	1TB	Intel Core i7
Петров	Петров	Петров	2018-05-29	М	0	16GB	8GB	1TB	Intel Core i7
Сидоров	Сидоров	Сидоров	2019-06-30	М	-1	16GB	8GB	1TB	Intel Core i7
Кузнецов	Кузнецов	Кузнецов	2020-07-31	М	-2	16GB	8GB	1TB	Intel Core i7
Левченко	Левченко	Левченко	2021-08-31	М	-3	16GB	8GB	1TB	Intel Core i7
Смирнов	Смирнов	Смирнов	2022-09-30	М	-4	16GB	8GB	1TB	Intel Core i7
Васильев	Васильев	Васильев	2023-10-31	М	-5	16GB	8GB	1TB	Intel Core i7
Попов	Попов	Попов	2024-11-30	М	-6	16GB	8GB	1TB	Intel Core i7
Морозов	Морозов	Морозов	2025-12-31	М	-7	16GB	8GB	1TB	Intel Core i7
Иванов	Иванов	Иванов	2026-01-31	М	-8	16GB	8GB	1TB	Intel Core i7
Петров	Петров	Петров	2027-02-28	М	-9	16GB	8GB	1TB	Intel Core i7
Сидоров	Сидоров	Сидоров	2028-03-27	М	-10	16GB	8GB	1TB	Intel Core i7
Кузнецов	Кузнецов	Кузнецов	2029-04-26	М	-11	16GB	8GB	1TB	Intel Core i7
Левченко	Левченко	Левченко	2030-05-25	М	-12	16GB	8GB	1TB	Intel Core i7
Смирнов	Смирнов	Смирнов	2031-06-24	М	-13	16GB	8GB	1TB	Intel Core i7
Васильев	Васильев	Васильев	2032-07-23	М	-14	16GB	8GB	1TB	Intel Core i7
Попов	Попов	Попов	2033-08-22	М	-15	16GB	8GB	1TB	Intel Core i7
Морозов	Морозов	Морозов	2034-09-21	М	-16	16GB	8GB	1TB	Intel Core i7
Иванов	Иванов	Иванов	2035-10-20	М	-17	16GB	8GB	1TB	Intel Core i7
Петров	Петров	Петров	2036-11-19	М	-18	16GB	8GB	1TB	Intel Core i7
Сидоров	Сидоров	Сидоров	2037-12-18	М	-19	16GB	8GB	1TB	Intel Core i7
Кузнецов	Кузнецов	Кузнецов	2038-01-17	М	-20	16GB	8GB	1TB	Intel Core i7
Левченко	Левченко	Левченко	2039-02-16	М	-21	16GB	8GB	1TB	Intel Core i7
Смирнов	Смирнов	Смирнов	2040-03-15	М	-22	16GB	8GB	1TB	Intel Core i7
Васильев	Васильев	Васильев	2041-04-14	М	-23	16GB	8GB	1TB	Intel Core i7
Попов	Попов	Попов	2042-05-13	М	-24	16GB	8GB	1TB	Intel Core i7
Морозов	Морозов	Морозов	2043-06-12	М	-25	16GB	8GB	1TB	Intel Core i7
Иванов	Иванов	Иванов	2044-07-11	М	-26	16GB	8GB	1TB	Intel Core i7
Петров	Петров	Петров	2045-08-10	М	-27	16GB	8GB	1TB	Intel Core i7
Сидоров	Сидоров	Сидоров	2046-09-09	М	-28	16GB	8GB	1TB	Intel Core i7
Кузнецов	Кузнецов	Кузнецов	2047-10-08	М	-29	16GB	8GB	1TB	Intel Core i7
Левченко	Левченко	Левченко	2048-11-07	М	-30	16GB	8GB	1TB	Intel Core i7
Смирнов	Смирнов	Смирнов	2049-12-06	М	-31	16GB	8GB	1TB	Intel Core i7
Васильев	Васильев	Васильев	2050-01-05	М	-32	16GB	8GB	1TB	Intel Core i7
Попов	Попов	Попов	2051-02-04	М	-33	16GB	8GB	1TB	Intel Core i7
Морозов	Морозов	Морозов	2052-03-03	М	-34	16GB	8GB	1TB	Intel Core i7
Иванов	Иванов	Иванов	2053-04-02	М	-35	16GB	8GB	1TB	Intel Core i7
Петров	Петров	Петров	2054-05-01	М	-36	16GB	8GB	1TB	Intel Core i7
Сидоров	Сидоров	Сидоров	2055-06-01	М	-37	16GB	8GB	1TB	Intel Core i7
Кузнецов	Кузнецов	Кузнецов	2056-07-01	М	-38	16GB	8GB	1TB	Intel Core i7
Левченко	Левченко	Левченко	2057-08-01	М	-39	16GB	8GB	1TB	Intel Core i7
Смирнов	Смирнов	Смирнов	2058-09-01	М	-40	16GB	8GB	1TB	Intel Core i7
Васильев	Васильев	Васильев	2059-10-01	М	-41	16GB	8GB	1TB	Intel Core i7
Попов	Попов	Попов	2060-11-01	М	-42	16GB	8GB	1TB	Intel Core i7
Морозов	Морозов	Морозов	2061-12-01	М	-43	16GB	8GB	1TB	Intel Core i7
Иванов	Иванов	Иванов	2062-01-01	М	-44	16GB	8GB	1TB	Intel Core i7
Петров	Петров	Петров	2063-02-01	М	-45	16GB	8GB	1TB	Intel Core i7
Сидоров	Сидоров	Сидоров	2064-03-01	М	-46	16GB	8GB	1TB	Intel Core i7
Кузнецов	Кузнецов	Кузнецов	2065-04-01	М	-47	16GB	8GB	1TB	Intel Core i7
Левченко	Левченко	Левченко	2066-05-01	М	-48	16GB	8GB	1TB	Intel Core i7
Смирнов	Смирнов	Смирнов	2067-06-01	М	-49	16GB	8GB	1TB	Intel Core i7
Васильев	Васильев	Васильев	2068-07-01	М	-50	16GB	8GB	1TB	Intel Core i7
Попов	Попов	Попов	2069-08-01	М	-51	16GB	8GB	1TB	Intel Core i7
Морозов	Морозов	Морозов	2070-09-01	М	-52	16GB	8GB	1TB	Intel Core i7
Иванов	Иванов	Иванов	2071-10-01	М	-53	16GB	8GB	1TB	Intel Core i7
Петров	Петров	Петров	2072-11-01	М	-54	16GB	8GB	1TB	Intel Core i7
Сидоров	Сидоров	Сидоров	2073-12-01	М	-55	16GB	8GB	1TB	Intel Core i7
Кузнецов	Кузнецов	Кузнецов	2074-01-01	М	-56	16GB	8GB	1TB	Intel Core i7
Левченко	Левченко	Левченко	2075-02-01	М	-57	16GB	8GB	1TB	Intel Core i7
Смирнов	Смирнов	Смирнов	2076-03-01	М	-58	16GB	8GB	1TB	Intel Core i7
Васильев	Васильев	Васильев	2077-04-01	М	-59	16GB	8GB	1TB	Intel Core i7
Попов	Попов	Попов	2078-05-01	М	-60	16GB	8GB	1TB	Intel Core i7
Морозов	Морозов	Морозов	2079-06-01	М	-61	16GB	8GB	1TB	Intel Core i7
Иванов	Иванов	Иванов	2080-07-01	М	-62	16GB	8GB	1TB	Intel Core i7
Петров	Петров	Петров	2081-08-01	М	-63	16GB	8GB	1TB	Intel Core i7
Сидоров	Сидоров	Сидоров	2082-09-01	М	-64	16GB	8GB	1TB	Intel Core i7
Кузнецов	Кузнецов	Кузнецов	2083-10-01	М	-65	16GB	8GB	1TB	Intel Core i7
Левченко	Левченко	Левченко	2084-11-01	М	-66	16GB	8GB	1TB	Intel Core i7
Смирнов	Смирнов	Смирнов	2085-12-01	М	-67	16GB	8GB	1TB	Intel Core i7
Васильев	Васильев	Васильев	2086-01-01	М	-68	16GB	8GB	1TB	Intel Core i7
Попов	Попов	Попов	2087-02-01	М	-69	16GB	8GB	1TB	Intel Core i7
Морозов	Морозов	Морозов	2088-03-01	М	-70	16GB	8GB	1TB	Intel Core i7
Иванов	Иванов	Иванов	2089-04-01	М	-71	16GB	8GB	1TB	Intel Core i7
Петров	Петров	Петров	2090-05-01	М	-72	16GB	8GB	1TB	Intel Core i7
Сидоров	Сидоров	Сидоров	2091-06-01	М	-73	16GB	8GB	1TB	Intel Core i7
Кузнецов	Кузнецов	Кузнецов	2092-07-01	М	-74	16GB	8GB	1TB	Intel Core i7
Левченко	Левченко	Левченко	2093-08-01	М	-75	16GB	8GB	1TB	Intel Core i7
Смирнов	Смирнов	Смирнов	2094-09-01	М	-76	16GB	8GB	1TB	Intel Core i7
Васильев	Васильев	Васильев	2095-10-01	М	-77	16GB	8GB	1TB	Intel Core i7
Попов	Попов	Попов	2096-11-01	М	-78	16GB	8GB	1TB	Intel Core i7
Морозов	Морозов	Морозов	2097-12-01	М	-79	16GB	8GB	1TB	Intel Core i7
Иванов	Иванов	Иванов	2098-01-01	М	-80	16GB	8GB	1TB	Intel Core i7
Петров	Петров	Петров	2099-02-01	М	-81	16GB	8GB	1TB	Intel Core i7
Сидоров	Сидоров	Сидоров	2100-03-01	М	-82	16GB	8GB	1TB	Intel Core i7
Кузнецов	Кузнецов	Кузнецов	2101-04-01	М	-83	16GB	8GB	1TB	Intel Core i7
Левченко	Левченко	Левченко	2102-05-01	М	-84	16GB	8GB	1TB	Intel Core i7
Смирнов	Смирнов	Смирнов	2103-06-01	М	-85	16GB	8GB	1TB	Intel Core i7
Васильев	Васильев	Васильев	2104-07-01	М	-86	16GB	8GB	1TB	Intel Core i7
Попов	Попов	Попов	2105-08-01	М	-87	16GB	8GB	1TB	Intel Core i7
Морозов	Морозов	Морозов	2106-09-01	М	-88	16GB	8GB	1TB	Intel Core i7
Иванов	Иванов	Иванов	2107-10-01	М	-89	16GB	8GB	1TB	Intel Core i7
Петров	Петров	Петров	2108-11-01	М	-90	16GB	8GB	1TB	Intel Core i7
Сидоров	Сидоров	Сидоров	2109-12-01	М	-91	16GB	8GB	1TB	Intel Core i7
Кузнецов	Кузнецов	Кузнецов	2110-01-01	М	-92	16GB	8GB	1TB	Intel Core i7
Левченко	Левченко	Левченко	2111-02-01	М	-93	16GB	8GB	1TB	Intel Core i7
Смирнов	Смирнов	Смирнов	2112-03-01	М	-94	16GB	8GB	1TB	Intel Core i7
Васильев	Васильев	Васильев	2113-04-01	М	-95	16GB	8GB	1TB	Intel Core i7
Попов	Попов	Попов	2114-05-01	М	-96	16GB	8GB	1TB	Intel Core i7
Морозов	Морозов	Морозов	2115-06-01	М	-97	16GB	8GB	1TB	Intel Core i7
Иванов	Иванов	Иванов	2116-07-01	М	-98	16GB	8GB	1TB	Intel Core i7
Петров	Петров	Петров	2117-08-01	М	-99	16GB	8GB	1TB	Intel Core i7
Сидоров	Сидоров	Сидоров	2118-09-01	М	-100	16GB	8GB	1TB	Intel Core i7
Кузнецов	Кузнецов	Кузнецов	2119-10-01	М	-101	16GB	8GB	1TB	Intel Core i7
Левченко	Левченко	Левченко	2120-11-01	М	-102	16GB	8GB	1TB	Intel Core i7
Смирнов	Смирнов	Смирнов	2121-12-01	М	-103	16GB	8GB	1TB	Intel Core i7
Васильев	Васильев	Васильев	2122-01-01	М	-104	16GB	8GB	1TB	Intel Core i7
Попов	Попов	Попов	2123-02-01	М	-105	16GB	8GB	1TB	Intel Core i7
Морозов	Морозов	Морозов	2124-03-01	М	-106	16GB	8GB	1TB	Intel Core i7
Иванов	Иванов	Иванов	2125-04-01	М	-107	16GB	8GB	1TB	Intel Core i7
Петров	Петров	Петров	2126-05-01	М	-108	16GB	8GB	1TB	Intel Core i7
Сидоров	Сидоров	Сидоров	2127-06-01	М	-109	16GB	8GB	1TB	Intel Core i7
Кузнецов	Кузнецов	Кузнецов	2128-07-01	М	-110	16GB	8GB	1TB	Intel Core i7
Левченко	Левченко	Левченко	2129-08-01	М	-111	16GB	8GB	1TB	Intel Core i7

[illegible]

ПРИЛОЖЕНИЕ В

(обязательное)

ПРЕЗЕНТАЦИЯ

					<i>ПГУ 1.090501.14.001 ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>108</i>

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ
КАФЕДРА «ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА»

Выпускная квалификационная работа на тему : «Система ивент менеджмента»

Выполнил:

студент группы 15ВО2
Тошев Е.Н

Руководитель ВКР:

к.т.н, доцент Федюнин Р.Н.

Пенза 2020

Цели и задачи

- ▶ Цель работы – разработать масштабируемую, отказоустойчивую систему ивент менеджмента, позволяющую группе людей отслеживать расположение задач на карте местности и координировать свои действия при их выполнении.
- ▶ Задачи:
 - ▶ Изучить предметную область;
 - ▶ Проанализировать предъявляемые требования;
 - ▶ Выбрать средства и технологии для разработки;
 - ▶ Разработать программные модули;
 - ▶ Протестировать и отладить программу.

Стек технологий

- ▶ Мобильное приложение разработано на базе ОС Android.
- ▶ Язык программирования Java.
- ▶ Среда разработки Eclipse.
- ▶ СУБД SQLite.
- ▶ Облачный сервер регистрации маяков
- ▶ Электронные BLE маяки

Электронный маяк



Электронный маяк компании Estimote

- ▶ В качестве электронного маяка выбрана модель компании Estimote, т.к. они в отличие от других производителей имеют режимы взаимодействия с внешними устройствами Eddystone, iBeacon, AltBeacon.
- ▶ Размер: 56 мм * 41 мм * 18 мм
- ▶ Работает с платформами IOS и Android.
- ▶ Встроенный пароль сопряжения.
- ▶ Поддержка настройки собственной конфигурации, включая считывание и изменение UUID и других характеристик сопряжения.
- ▶ Настройка мощности.
- ▶ Контекстные сообщения могут быть запрограммированы из личного кабинета Google.

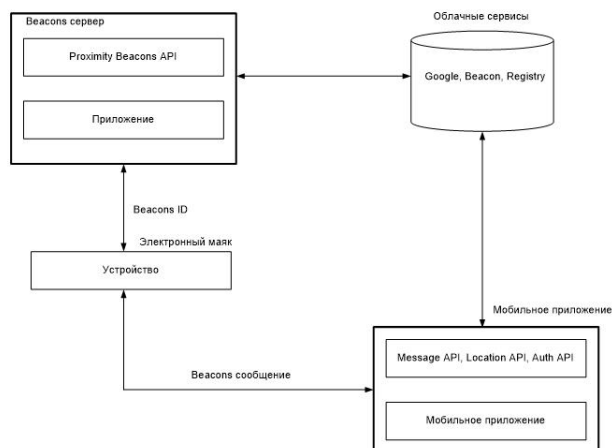
Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.14.001 ПЗ

Лист

110

Архитектура системы



Роли и типы пользователей

Тип пользователя	Возможности
Администратор	Приглашает членов команды. Координирует работу команды. Формирует персональные задачи.
Член команды	Координирует работу команды. Формирует персональные задачи.
Волонтер	Формирует персональные задачи.

Диаграмма последовательности администратор

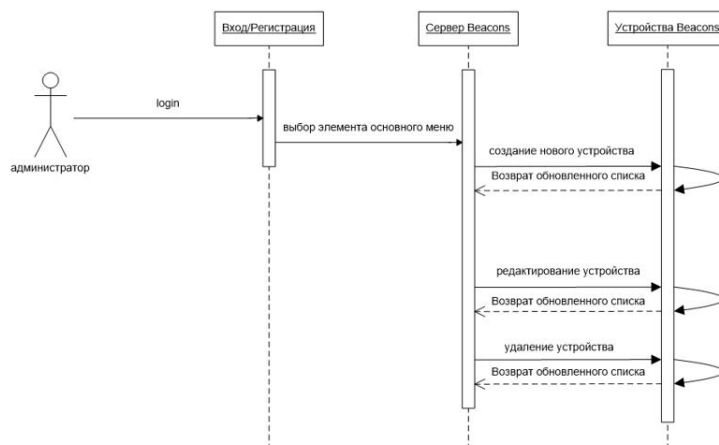
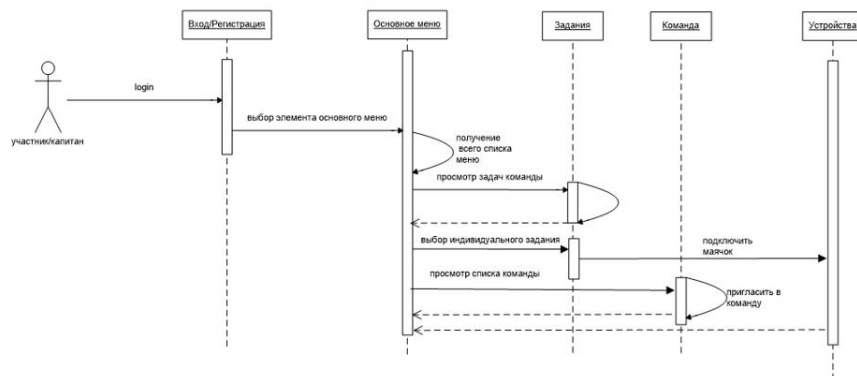


Диаграмма последовательности пользователь



14

100

Пользовательский интерфейс

EventManagment

Управляй своей командой

Логин

Пароль

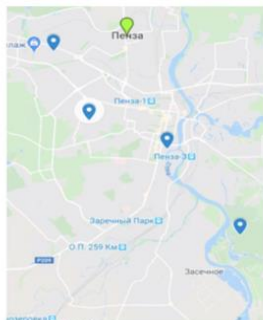
[Забил пароль?](#)

Войти

Нет аккаунта? [Создать аккаунт](#)

Вход в учетную запись

Просмотр устройств



Карта устройств

EventManagment

ГЛАВНОЕ МЕНЮ

- Календарь событий
- Задачи
- Команда
- Устройства
- Профиль пользователя

Главное меню

Меню Профиль пользователя



Владимир У.
uteev.v@gmail.com

Активность

Волонтер 1

Волонтерское движение Доброе сердце

Сменить пароль

[Выйти из приложения](#)

Профиль пользователя

Пользовательский интерфейс

Меню Команда

Администратор — Евгений Т.
evg2d.v@gmail.com

Член команды — Артём С.
santa.v@gmail.com

Волонтер 1 — Владимир У.
uteev.v@gmail.com

Волонтер 2 — Игнат С.
infhdv@gmail.com

Команда

Меню Задачи

Задачи команды

Персональные задачи

Задачи

Задачи Задачи команды

Волонтерское движение Доброе сердце 09:00 >
Забрать подарки из дома дружбы

Встреча 18:00 >
Интервью с кандидатом

Задачи команды

Волонтерское движение Доброе сердце

Забрать подарки из дома дружбы

Адрес
г.Пенза, Проспект Мира, д. 43

Волонтер 1 — Владимир У.
uteev.v@gmail.com

Персональная задача

Пригласить команду

Заключение

- ▶ В ходе выполнения дипломной работы было разработано мобильное приложение на базе ОС Android, обеспечивающее организацию участия в мероприятиях типа волонтерских движений.
- ▶ Разработка мобильного приложения велась на языке программирования Java.
- ▶ Среда разработки - свободная интегрированная среда разработки модульных кроссплатформенных приложений Eclipse.
- ▶ Система управления базами данных для хранения и обработки структуры данных выбрана SQLite.
- ▶ Процедура идентификации реализована при помощи свободно распространяемых библиотек API.

СПАСИБО ЗА ВНИМАНИЕ!

ПРИЛОЖЕНИЕ Г

REST ЗАПРОСЫ

					<i>ПГУ 1.090501.14.001 ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>116</i>

Работа с группой пользователей:

1.Получение информации о всех группах

[https://proximitybeacon.googleapis.com/v1beta1/beacons: UserGroups](https://proximitybeacon.googleapis.com/v1beta1/beacons:UserGroups)

В результате получим список всех групп:

```
{
  "UserGroups" : {
    [
      "Id" =1,
      "name": "Test1"
    ],
    [
      "Id" =2,
      "name": "Test2",
    ],
    [
      "Id" =100,
      "name": "TestUserGROUP"
    ]
  ]
}
```

2.Получение информации о конкретной группе

[https://proximitybeacon.googleapis.com/v1beta1/beacons: UserGroups<id>](https://proximitybeacon.googleapis.com/v1beta1/beacons:UserGroups<id>)

Вместо <id> необходимо вставить идентификатор обновляемой записи.

Если id = 100 в результате получим результат:

{

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		117

```
"name": "TestUserGROUP"
```

```
}
```

3.Получение информации о группе с пользователями

[https://proximitybeacon.googleapis.com/v1beta1/beacons: UserGroups<id>:Users](https://proximitybeacon.googleapis.com/v1beta1/beacons:UserGroups<id>:Users)

В результате получим список пользователей в группе:

```
{
```

```
"UserGroups" = id : {
```

```
[
```

```
  "Id" =1,
```

```
  "name": "Евгений",
```

```
  "secondname": "Тошев",
```

```
  "role": "Волонтер",
```

```
  "device": " Fr4Z98nSoW0hgAAAAAAAAAaAg",
```

```
],
```

```
[
```

```
  "Id" =2,
```

```
  "name": "Артем",
```

```
  "secondname": "тарасов",
```

```
  "role": "Волонтер",
```

```
  "device": " Fr4Z98nSoW0hgAAAAAAAAAaAg",
```

```
],
```

```
}
```

Работа с устройствами.

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		118

1.Получение информация о всех зарегистрированных устройствах:

<https://proximitybeacon.googleapis.com/v1beta1/beacons:register>

```
{ "advertisedId": {  
  
  "type": "EDDYSTONE",  
  
  "id": "Fr4Z98nSoW0hgAAAAAAAAAag"  
  
},  
  
"status": "ACTIVE",  
  
"placeId": "ChIJTxax6NoSkFQRWPvFXI1LypQ",  
  
  "latLng": {  
  
    "latitude": "47.6693771",  
  
    "longitude": "-122.1966037"  
  
  },  
  
  "description": "Волонтерская программа Дети весны",  
  
}
```

2.Получение информация о конкретном зарегистрированном устройстве:

<https://proximitybeacon.googleapis.com/v1beta1/beacons:register<id>>

```
{ "advertisedId": {  
  
  "type": "EDDYSTONE",  
  
  "id": "Fr4Z98nSoW0hgAAAAAAAAAag"  
  
},  
  
"status": "ACTIVE",  
  
"placeId": "ChIJTxax6NoSkFQRWPvFXI1LypQ",  
  
"latLng": {
```

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		119

"latitude": "47.6693771",

"longitude": "-122.1966037"

},

"description": "Волонтерская программа Дети весны",

}

					ПГУ 1.090501.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		120