

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Институт естественных и точных наук  
Факультет математики, механики и компьютерных технологий  
Кафедра прикладной математики и программирования  
Направление подготовки Программная инженерия

РАБОТА ПРОВЕРЕНА

\_\_\_\_\_ (И.О. Ф.)  
« \_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,  
доцент

\_\_\_\_\_/А.А.Замышляева  
« \_\_\_\_ » \_\_\_\_\_ 2018 г.

Разработка мобильного приложения для навигации по учебным корпусам  
ЮУрГУ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  
ЮУрГУ–09.03.04.2018.103.ПЗ ВКР

Руководитель работы, к.э.н., доцент

\_\_\_\_\_/ А.К. Демидов  
« \_\_\_\_ » \_\_\_\_\_ 2018 г.

Автор работы

Студент группы ЕТ-414

\_\_\_\_\_/ М.А. Свиридов  
« \_\_\_\_ » \_\_\_\_\_ 2018 г.

Нормоконтролер, к.т.н., доцент

\_\_\_\_\_/ Т.Ю.Оленчикова  
« \_\_\_\_ » \_\_\_\_\_ 2018 г.

Челябинск,  
2018

## АННОТАЦИЯ

Свиридов М.А. Разработка мобильного приложения для навигации по учебным корпусам ЮУрГУ. - Челябинск: ЮУрГУ, ЕТ-414, 42 с, 28 ил., 4 табл., библиогр. список – 20 наим, 3 прил.

Данная работа посвящена разработке мобильного приложения, позволяющего пользователям ориентироваться на территории учебных корпусов Южно-Уральского государственного Университета.

В работе выполнен обзор уже существующих решений подобного типа.

Разработана архитектура системы, включающая в себя диаграмму классов, а также создана диаграмма прецедентов и диаграмма активности. Спроектирована база данных для хранения данных необходимых для функционирования программы.

Программа реализована на языке программирования Java с использованием СУБД SQLite. В приложении приведен текст программы.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	7
1ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ НАВИГАЦИИ.....	8
1.1 Основные понятия предметной области.....	8
1.2 Обзор существующих решений.....	8
1.2.1 «2ГИС».....	8
1.2.2 «Виртуальный ЮУрГУ».....	11
1.2.3 Схема студгородка МФТИ.....	12
1.2.4 Схема расположения зданий Московского государственного университета.....	14
1.3 Постановка задачи.....	17
1.4 Требования к программе.....	17
1.5 Выводы по разделу.....	17
2РАЗРАБОТКА ПРИЛОЖЕНИЯ.....	18
2.1 Диаграмма прецедентов.....	18
2.2 Диаграмма классов.....	19
2.3 Диаграмма активности.....	20
2.4 Выводы по разделу.....	23
3РАЗРАБОТКА БАЗЫ ДАННЫХ.....	24
3.1 Построение ER-диаграммы.....	24
3.2 Структура связей реляционной базы данных.....	25
3.3 Выявление ограничений и правил поддержания целостности данных... ..	25
3.4 Выводы по разделу.....	28
4РЕАЛИЗАЦИЯ СИСТЕМЫ.....	29
4.1 Реализация алгоритмов.....	29
4.1.1 Общий алгоритм системы.....	29
4.1.2 Алгоритм обработки событий окна.....	29
4.1.3 Алгоритм обработки событий поля с информацией об объекте.....	31
4.1.4 Алгоритм обработки событий меню приложения.....	32

4.1.5 Алгоритм осуществления поиска .....	32
4.1.6 Алгоритм построения маршрута .....	33
4.2 Проверка работоспособности .....	35
4.3 Выводы по разделу.....	39
ЗАКЛЮЧЕНИЕ .....	40
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	41
ПРИЛОЖЕНИЕ 1 ТЕКСТ ПРОГРАММЫ .....	43
ПРИЛОЖЕНИЕ 2 .....	86
ПРИЛОЖЕНИЕ 3 .....	90

## ВВЕДЕНИЕ

Современное общество всё больше подвергается влиянию различных технологий, которые проникают во все сферы человеческой деятельности. Сейчас уже практически невозможно представить себе человека, который бы ни разу не столкнулся с необходимостью использовать их. Не говоря уже о том, что нынешнее общество становится сложно представить без мобильных устройств будь-то телефон, смартфон или другие коммуникаторы.

Мобильные устройства оказывают колоссальное влияние на развитие общества и современных технологий в целом, так как благодаря ним любой человек может хранить под рукой важную информацию и использовать различный функционал, упрощающий жизнь, в любое время и месте.

С появлением все новых и новых мобильных устройств, а также разнообразием платформ, на которых они разрабатываются, активно развивается бизнес по разработке мобильных приложений различного назначения. Люди чаще всего используют их для развлечений (различные игры), выхода в интернет, общения (социальные сети, сообщения, звонки, мессенджеры), ведения бизнеса и проведения различных рекламных кампаний.

В текущий период времени разнообразие мобильных устройств настолько велико, что возникает проблема выбора подо что конкретно стоит разрабатывать приложение.

Наиболее популярными на данный момент можно считать платформы Android и iOS. Доля каждой из них на мировом рынке является наиболее значимой. Но все же самой часто используемой платформой за последнее время является Android, так как большинство современных мобильных устройств основаны именно на ней.

Данная дипломная работа посвящена разработке мобильного приложения-навигатора по учебным корпусам Южно-Уральского государственного университета.

ЮУрГУ является одним из крупнейших университетов России. В его стенах на текущий момент учится около 30 тыс. студентов, а студенческий городок университета – крупнейший и один из лучших комплексов. Из этого возникает актуальность создания программы подобного рода, так как количество корпусов и различных мест вуза являются значительными, из-за чего большинству абитуриентов и студентов данного вуза, которые обучаются заочно и не часто находятся в стенах университета или только поступили и не успели разобраться со структурой и расположением, может быть сложно ориентироваться на этой территории. Также разработка данного программного продукта позволит ознакомиться с созданием мобильных приложений и получить знания в этой области.

Данная программа должна облегчить поиск необходимого места, аудитории или помещения как внутри зданий студенческого городка, так и снаружи.

# 1 ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ НАВИГАЦИИ

## 1.1 Основные понятия предметной области

Мобильное приложение - программа, написанная под ту или иную платформу, обладающая определенным функционалом для выполнения заранее заданных действий.

Платформа – операционная система, на которой основано мобильное устройство.

Операционная система – комплекс взаимосвязанных программ, предназначенных для управления ресурсами устройства и организации взаимодействия с пользователем.

Приложение-навигатор – приложение, осуществляющее навигацию на различных территориях путем указания конкретного маршрута между двумя точками на карте, описанием различных мест и отображением текущего местоположения пользователя.

«Оффлайн» режим - режим, не предполагающий наличие постоянного подключения к интернету.

«Онлайн» режим – режим, для работы которого необходимо наличие интернета.

## 1.2 Обзор существующих решений

### 1.2.1 «2ГИС»

«2ГИС» – приложение-навигатор по различным городам России и также ряду других стран. Реализовано для платформ iOS, Android, Windows Phone, Symbian, Windows Mobile.

Основное окно для работы с приложением изображено на Рисунке 1.1

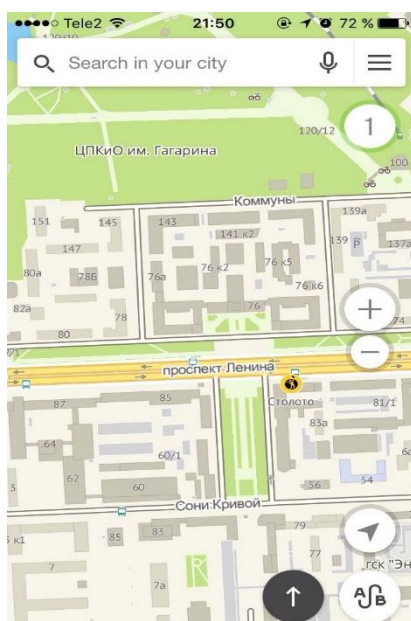


Рисунок 1.1 – «Основное окно»

На нем пользователю предоставляется возможность изучения в «оффлайн» режиме карты города, поиск собственного местоположения и необходимой ему точки на карте, а также построение маршрута между двумя отмеченными точками.

Каждая выделенная пользователем точка имеет собственное описание как показано на Рисунке 1.2

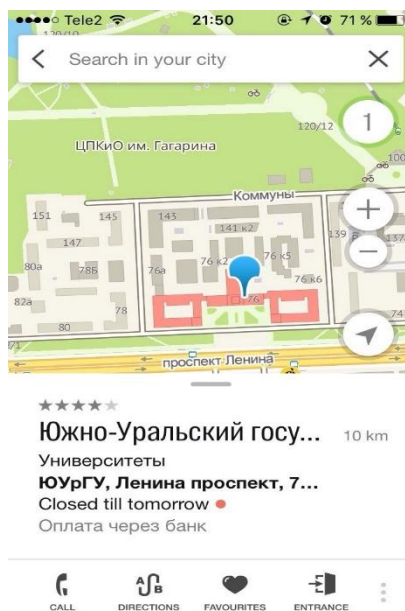


Рисунок 1.2 – «Описание выделенной точки»

Построение маршрута осуществляется как путем выделения конкретных точек на карте, так и указанием точного адреса в специальных полях. Пример на Рисунке 1.3

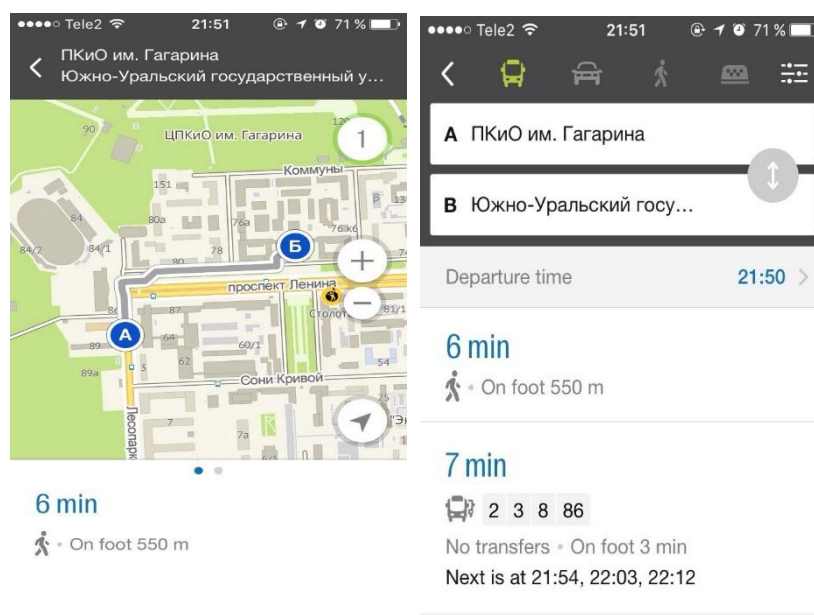


Рисунок 1.3 – «Построение маршрута»: а) указание точек маршрута; б) дополнительная информация по этому маршруту

Также данная система имеет возможность отображения внутреннего устройства некоторых общественных мест. Пример на Рисунке 1.4

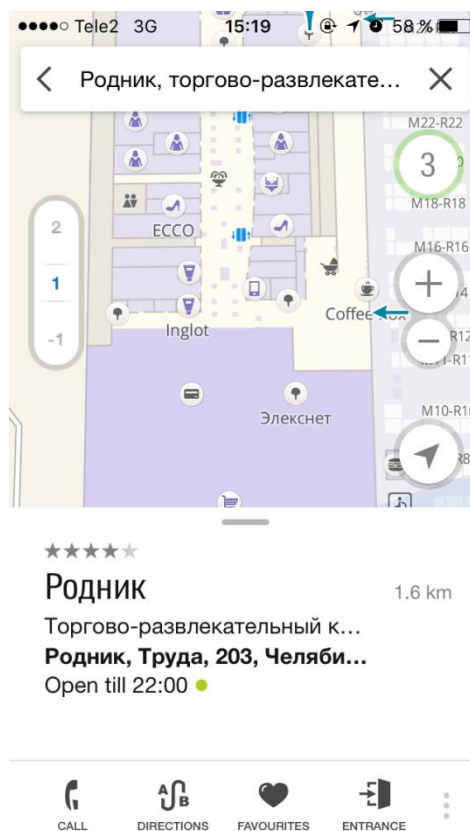


Рисунок 1.4 – «Отображение внутреннего устройства ТРК Родник»

На данном рисунке отображен вид здания ТРК Родник изнутри. Пользователь также может взаимодействовать объектами внутри, как и с объектами основной карты.

Достоинства приложения «2ГИС»:

- 1) возможность работы с картами в режиме «оффлайн»;
- 2) постоянная поддержка актуальности карты;
- 3) приложение бесплатно.

Недостатки приложения «2ГИС»:

- 1) иногда встречается неточность данных на карте, вызванная тем, что приходится работать с очень большим объемом информации;
- 2) отсутствие локальных карт, позволяющих ориентироваться по многим общественно значимым местам, как например студенческий городок ЮУрГУ.



## 1.2.2 «Виртуальный ЮУрГУ»

«Виртуальный ЮУрГУ» - интернет ресурс, на котором находятся панорамные туры интерьеров, позволяющие пользователю «пройтись» по значимым объектам университета.

Основная страница для работы с «Виртуальным ЮУрГУ» представлена на Рисунке 1.5



Рисунок 1.5 – «Основная страница»

Данный интернет ресурс позволяет выбрать и просмотреть панорамные изображения интерьеров университета. Пользователь может приближать, отдалять и вращать камеру для более детального рассмотрения местности, а также «пройтись» по выбранному объекту. Пример изображен на Рисунке 1.6



Рисунок 1.6 – «Пример работы системы»

Достоинства «Виртуального ЮУрГУ»:

- 1) пользователь может увидеть реальное изображения интерьеров университета в мельчайших подробностях;
- 2) позволяет сориентироваться в университете, основываясь на панорамных изображениях необходимых объектов ВУЗа.

Недостатки:

- 1) в данной версии отсутствует множество объектов университета;
- 2) пользователь должен ориентироваться только исходя из панорамных изображений без каких-либо указателей;
- 3) пользователь не может проложить маршрут до необходимой ему точки.

### 1.2.3 Схема студгородка МФТИ

На сайте Московского физико-технического института пользователь может ознакомиться со схемой, на которой расположены все корпуса данного учебного заведения. Она представлена на Рисунке 1.7

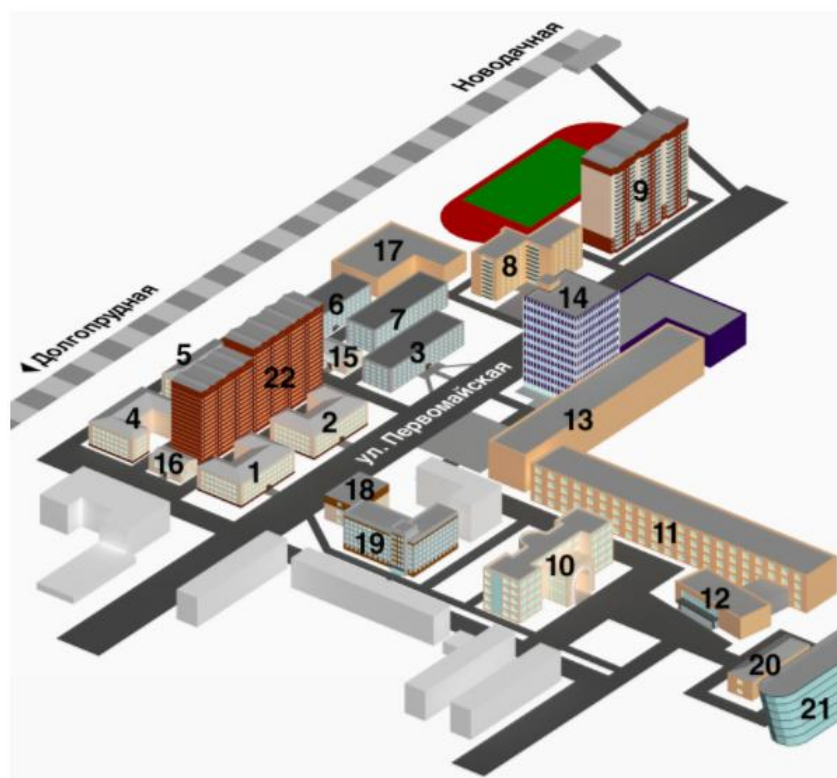


Рисунок 1.7 – «Схема студгородка МФТИ»

После данной схемы расположены расшифровки всех обозначений зданий на данной схеме. Они приведены на рисунке 1.8

1	-	Общеквирта №1
2	-	Общеквирта №2
3	-	Общеквирта №3
4	-	Общеквирта №4
5	-	Поликлиника, профилакторий
6	-	Общеквирта №6
7	-	Общеквирта №7
8	-	Общеквирта №8
9	-	<b>Общеквирта №9</b>
10	-	Лабораторный корпус
11	-	<b>Главный корпус (ГК)</b>
12	-	Концертный зал (КЗ)
13	-	<b>Новый корпус (НК)</b>
14	-	Корпус прикладной математики (КПМ)
15	-	Спортивный корпус №2
16	-	Спортивный корпус №1, клуб выпускников
17	-	Спортивный корпус (бассейн)
18	-	Студенческая столовая
19	-	Аудиторный корпус (АК)
20	-	Радиотехнический корпус
21	-	Биофармакластер МФТИ
22	-	Общеквирта №10 и №11

Рисунок 1.8 – «Расшифровка обозначений схемы студгородка МФТИ»

Достоинства:

- 1) пользователю предоставляется полная схема расположения всех зданий студгородка данного учебного заведения.

Недостатки:

- 1) данная схема позволяет только просмотреть как расположены корпуса института без какой-либо возможности проложить маршрут;
- 2) отсутствуют схемы устройства корпусов с расположением аудиторий и других мест внутри зданий;
- 3) для поиска какого-либо места пользователю необходимо самому найти данное место на карте, затем найти его расшифровку в перечне.

#### 1.2.4 Схема расположения зданий Московского государственного университета.

Данная схема представляет собой фрагмент Яндекс-карт, отображающий студгородок университета. Основная страница для ее просмотра представлена на Рисунке 1.9



Рисунок 1.9 – «Основная страница для просмотра схемы студгородка МГУ»



На данной схеме пользователь может выделить интересующий его корпус и получить небольшую информацию о выбранном объекте, состоящую из фотографии места, описания и рекомендаций по поводу того, как добраться. Пример изображен на рисунке 1.10

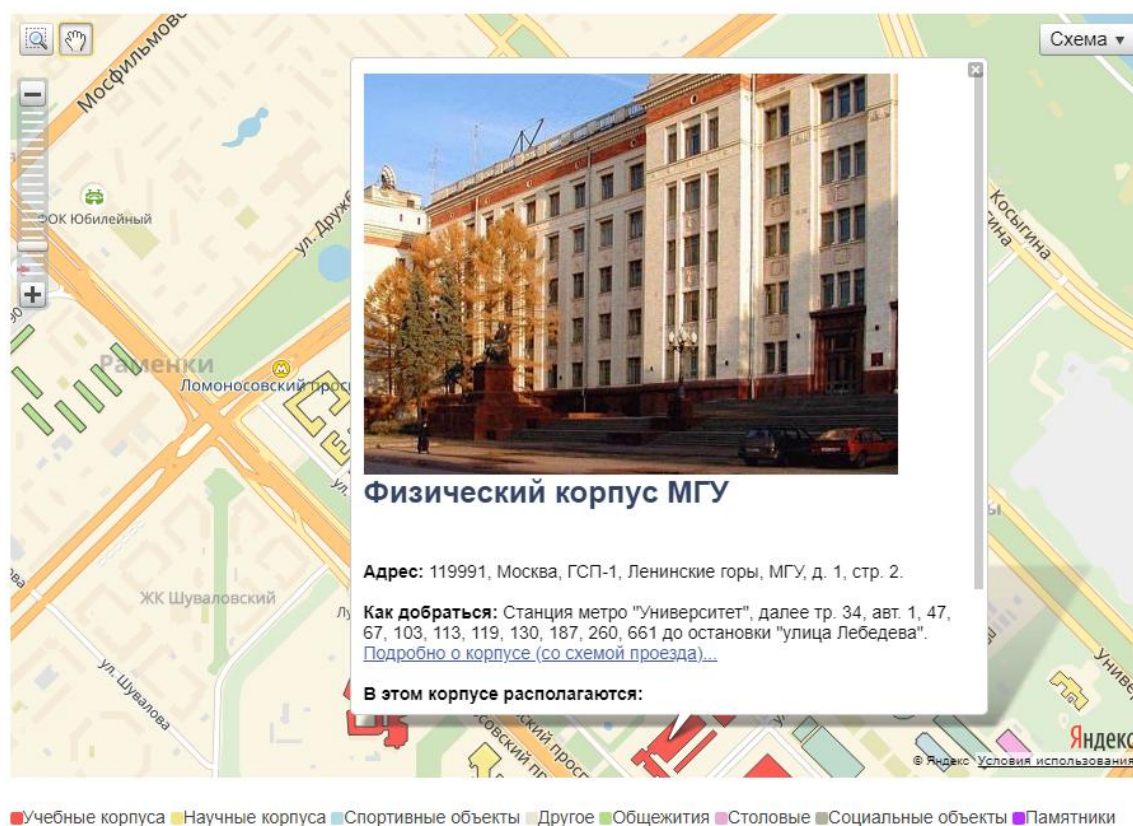


Рисунок 1.10 – «Информация о выбранном объекте»

Также присутствует отдельная схема для внутреннего устройства корпуса исторического факультета. На ней пользователь может более подробно ознакомиться с внутренним устройством корпуса, т.е. просмотреть расположения различных мест на каждом из этажей. Пример на Рисунке 1.11

## Шуваловский корпус МГУ

Аудиторный фонд и служебные помещения, используемые историческим факультетом, расположены на 1-2 и 4-6 этажах Шуваловского корпуса МГУ



### ▼ 1 ЭТАЖ

почтовые и учебные аудитории, приемная комиссия, диссертационные советы, редакция "Вестника МГУ" (серия "История"), Совет ветеранов, Комната боевой славы, лаборатория телекоммуникаций (серверная), типография, столовая



Е101 – учебная аудитория  
Е102 – учебная аудитория  
Е103 – учебная аудитория  
Е106 – учебная аудитория  
Е107 – учебная аудитория

Е124 – диссертационные советы. Редакция "Вестника МГУ" (серия 8, история)  
Е126 – типография  
Е128 – приемная комиссия  
Е129 – техническое помещение

Рисунок 1.11 – «Схема внутреннего устройства корпуса исторического факультета»

После каждой схемы, представленной на данном портале, идет расшифровка всех используемых обозначений.

Достоинства:

- 1) предоставляется полная схема расположения всех корпусов университета, в соответствии с их реальным местонахождением;
- 2) пользователь может получить информацию о том, как добраться до какого-либо корпуса;
- 3) в случае с корпусом исторического факультета, можно увидеть внутреннее устройство здания и найти интересующее место внутри него.

Недостатки:

- 1) пользователь должен вручную искать интересующее его место;
- 2) отсутствует возможность прокладки маршрута;

- 3) во всех корпусах, кроме исторического, отсутствует схема внутреннего устройства;
- 4) в схеме корпуса исторического факультета необходимо разбираться с расшифровкой обозначений, для того, чтобы найти необходимое место.

### 1.3 Постановка задачи

Целью данного проекта является исследование организации внутреннего устройства студенческого городка Южно-Уральского государственного университета и написание мобильного приложения для навигации по его территории. Данное приложение должно позволять ориентироваться пользователю на всей области, где расположено данное высшее учебное заведение и сообщать ему всю необходимую информацию о местах на карте.

### 1.4 Требования к программе

Требования к функциональным характеристикам:

- возможность построения маршрута из точки А в точку В;
- поиск необходимого места через поисковую строку;
- описание аудитории, здания на карте, позволяющее пользователю понять, что конкретно находится в указанном месте.

Требования к составу и параметрам технических средств:

Система должна работать на устройствах, совместимых с операционной системой Android 4.1 и выше, с установленным приложением Google Services. Для успешного функционирования системы необходимы:

- оперативная память не менее 20 Мб;
- свободная память не менее 50 Мб.

### 1.5 Выводы по разделу

В данном разделе были рассмотрены существующие аналогии подобного мобильного приложения. На основании этого можно сказать, что они не позволяют пользователю ориентироваться в полной мере на территории Южно-Уральского государственного университета, так как в них отсутствует информация о внутреннем устройстве ЮУрГУ, либо имеется лишь частично.

Также была сформулирована и поставлена задача проекта.

## 2 РАЗРАБОТКА ПРИЛОЖЕНИЯ

### 2.1 Диаграмма прецедентов

Данная диаграмма описывает варианты использования данного приложения и поведение системы в каждом случае. При ее создании использовался специальный универсальный язык для моделирования – UML.

На рисунке 2.1 приведена диаграмма, описывающая то, как пользователь взаимодействует с системой.

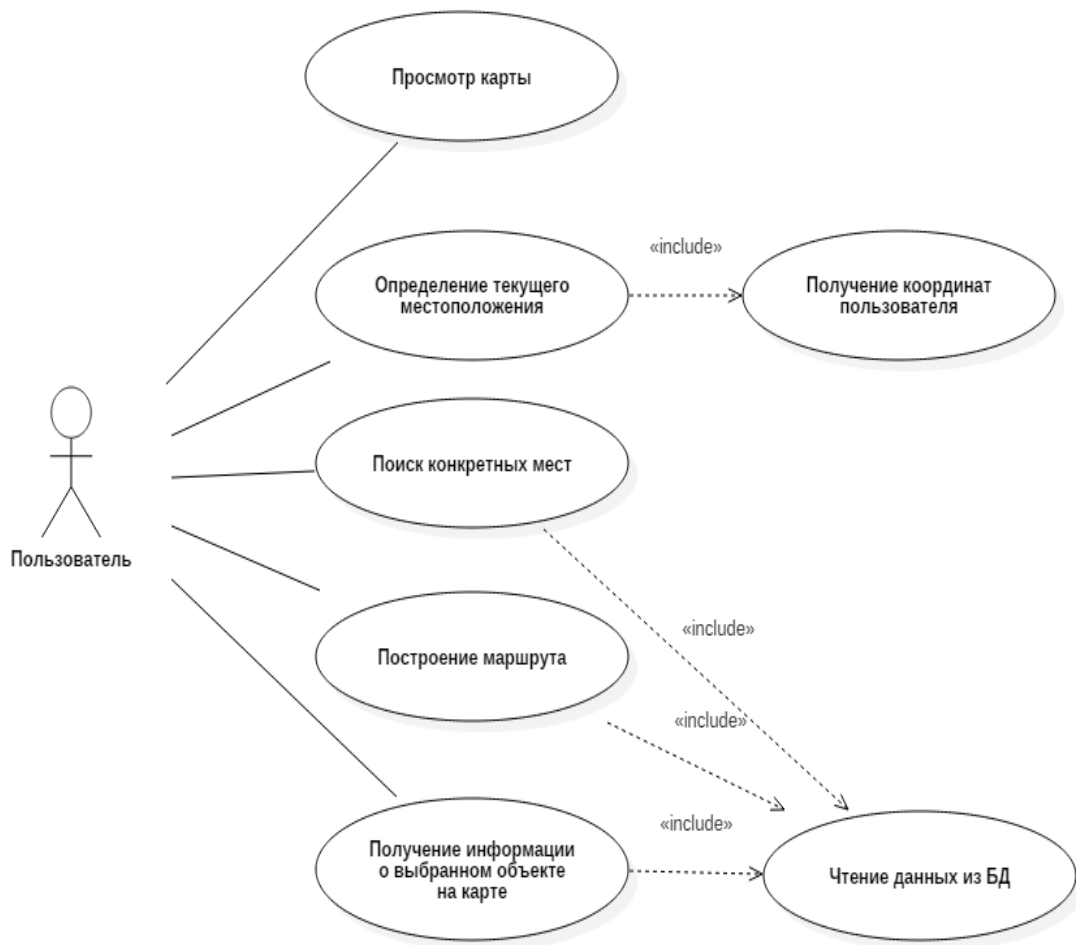


Рисунок 2.1 – Общая диаграмма прецедентов



## 2.2 Диаграмма классов

Подобный вид диаграммы описывает связи, отношения и структуру элементов системы между собой. На Рисунке 2.2 представлены классы данного приложения.

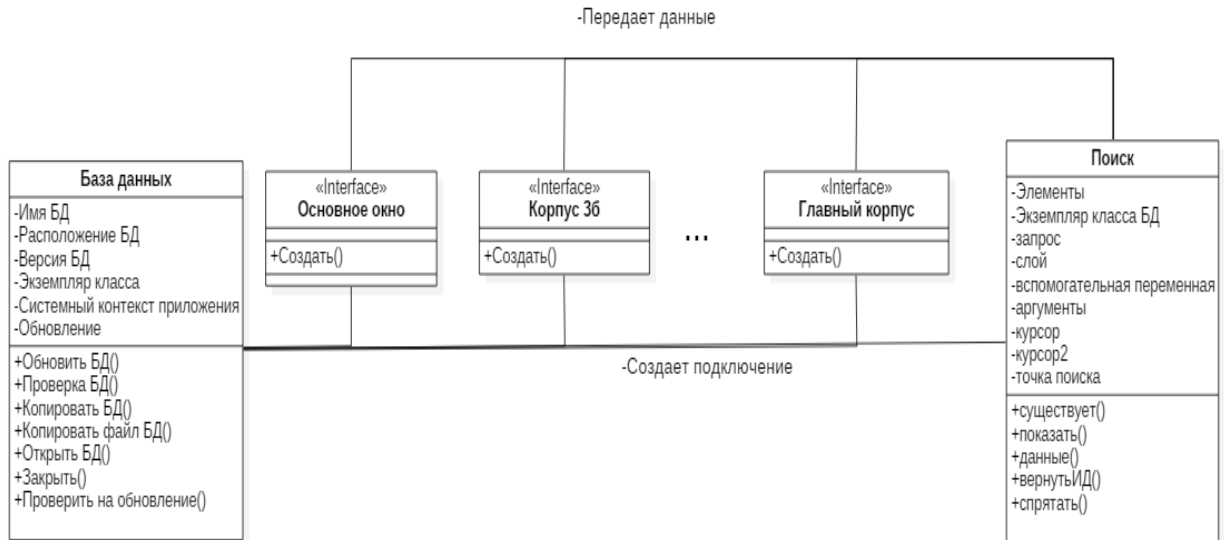


Рисунок 2.2 – Диаграмма классов

На данной диаграмме представлены классы, которые были созданы для работы приложения. Они все являются наследниками стандартного класса `Activity` из `Android API`. Класс `DataBaseHelper` создает подключение к базе данных и необходим для работы с ее функционалом. Класс `Activity_search` используется для реализации поисковых возможностей программы. Классы `MAIN`, `Activity_3b` и т.д. являются классами, реализующие работу с интерфейсом программы.

### 2.3 Диаграмма активности

Данный вид диаграммы описывает поведение программы, т.е. отображает последовательность выполнения команд во время работы приложения.

На Рисунке 2.3 представлена общая диаграмма активности работы приложения.

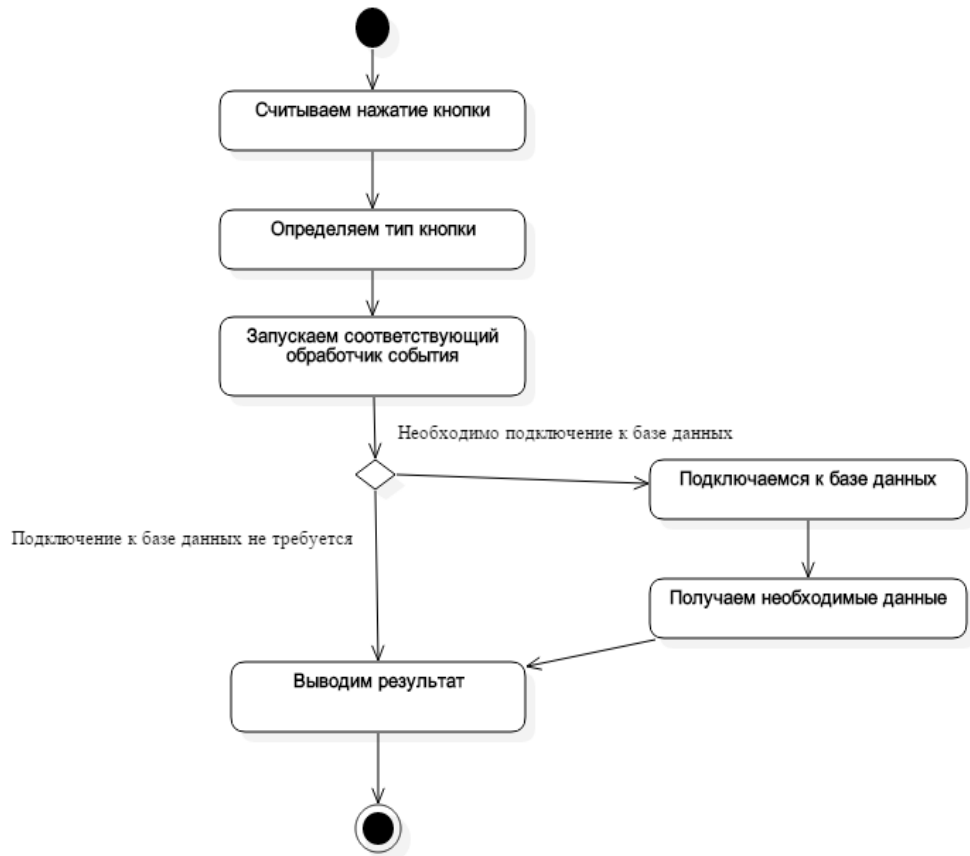


Рисунок 2.3 – «Диаграмма активности работы приложения»

Каждый объект на карте является отдельной кнопкой, нажатие по которой запускает соответствующий обработчик события для отображения информации об объекте, запуске отдельных функций или открытия дополнительных окон и полей.

Рисунок 2.4 представляет собой диаграмму активности для поиска пользователем объекта.

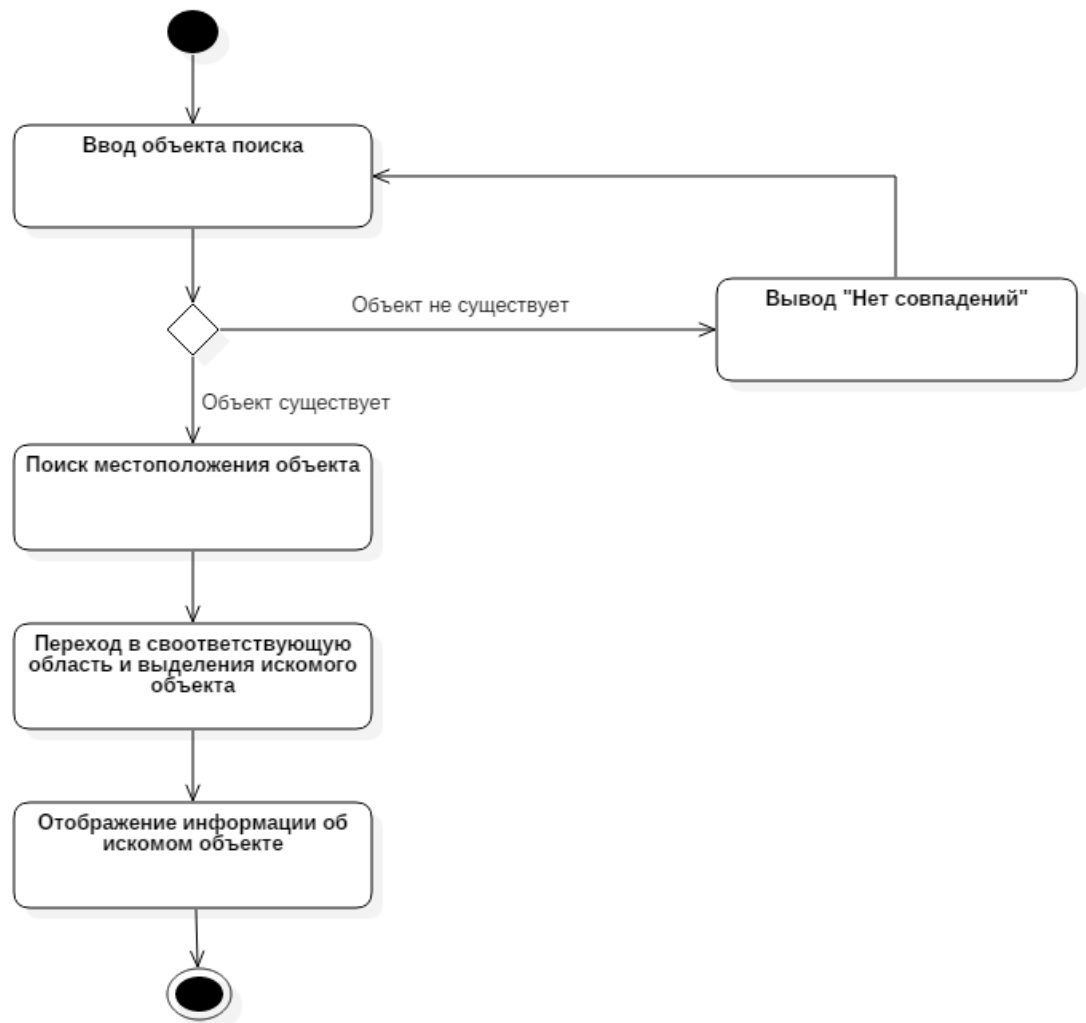


Рисунок 2.4 – «Диаграмма активности для поиска объекта»

Система проверяет введенные данные на наличие, сверяясь с базой данных. Если такой объект был найден, то функция возвращает значение «истина», иначе «ложь». После чего запускается другая функция, отображающая искомый элемент на карте.

На рисунке 2.5 изображена диаграмма активности для отображения маршрута между двумя указанными точками.

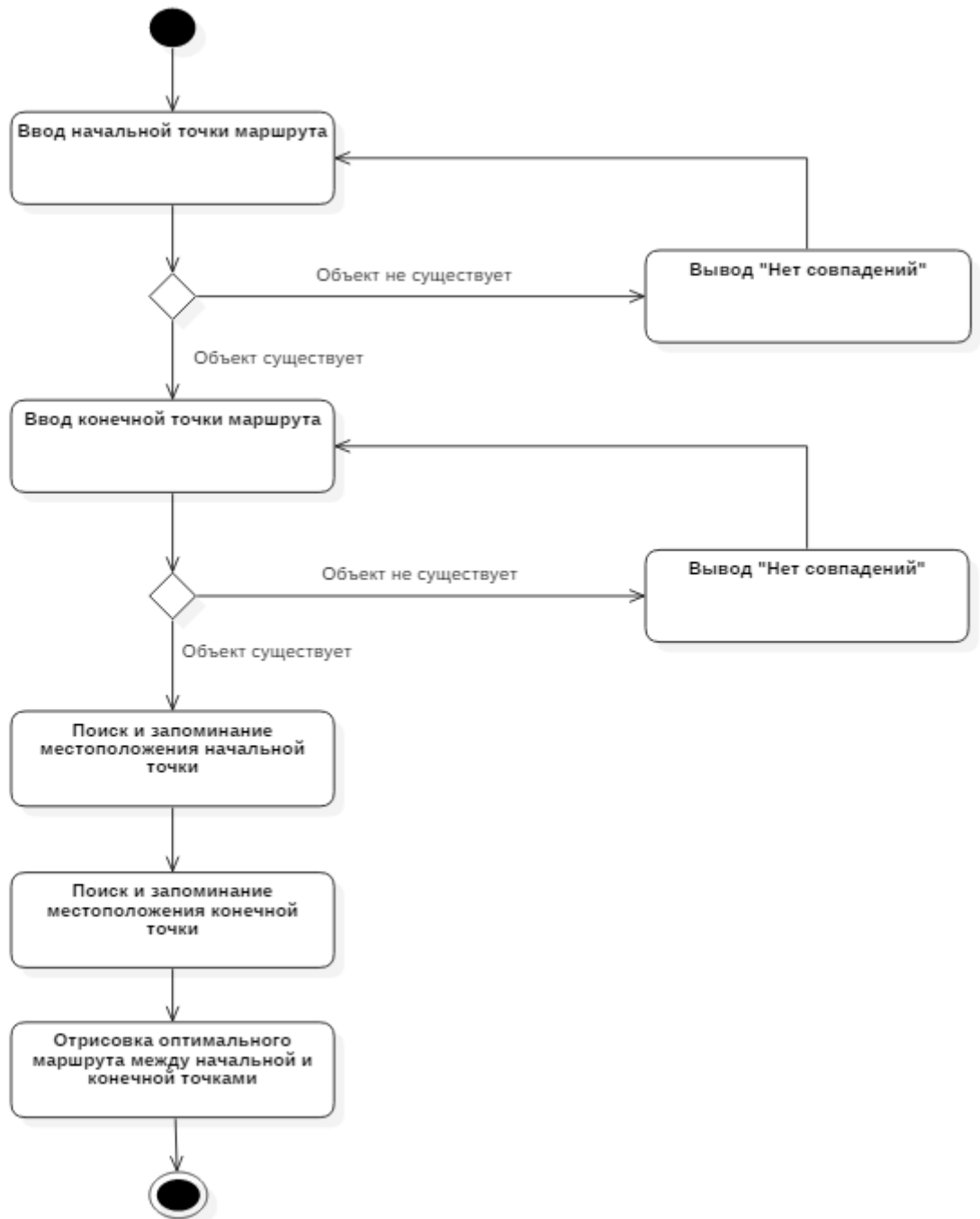


Рисунок 2.5 – «Диаграмма активности для отображения маршрута между двумя указанными точками»

В этом случае система также сверяется с базой данных для проверки на наличие указанных объектов. Но отображение изначально строится в том же самом окне, где находится начальный объект, а дальнейший путь передается в другие слои и отображается по мере того, как пользователь запускает необходимое окно.

## 2.4 Выводы по разделу

В данном разделе была разработана архитектура системы, включающая в себя диаграмму классов. Также были созданы диаграммы прецедентов и активности. Диаграммы были описаны с помощью языка UML.

### 3 РАЗРАБОТКА БАЗЫ ДАННЫХ

Для разработки данного мобильного приложения используется база данных SQLite, которая является встраиваемой СУБД. Встраиваемость подразумевает под собой, что она не использует общераспространённую парадигму клиент-сервер.

Данный тип СУБД является оптимально подходящим для разработки подобного мобильного приложения, так как в ходе работы пользователь взаимодействует с базой данных только для получения из нее информации.

#### 3.1 Построение ER-диаграммы

Данная диаграмма описывает ключевые сущности и связи, которые установлены между ними.

Выделено четыре сильные сущности: Layout (Макет), Buildings (Здания), Classes (Аудитории), SafePoints (Безопасные точки) и четыре связи между ними типа «один ко многим».

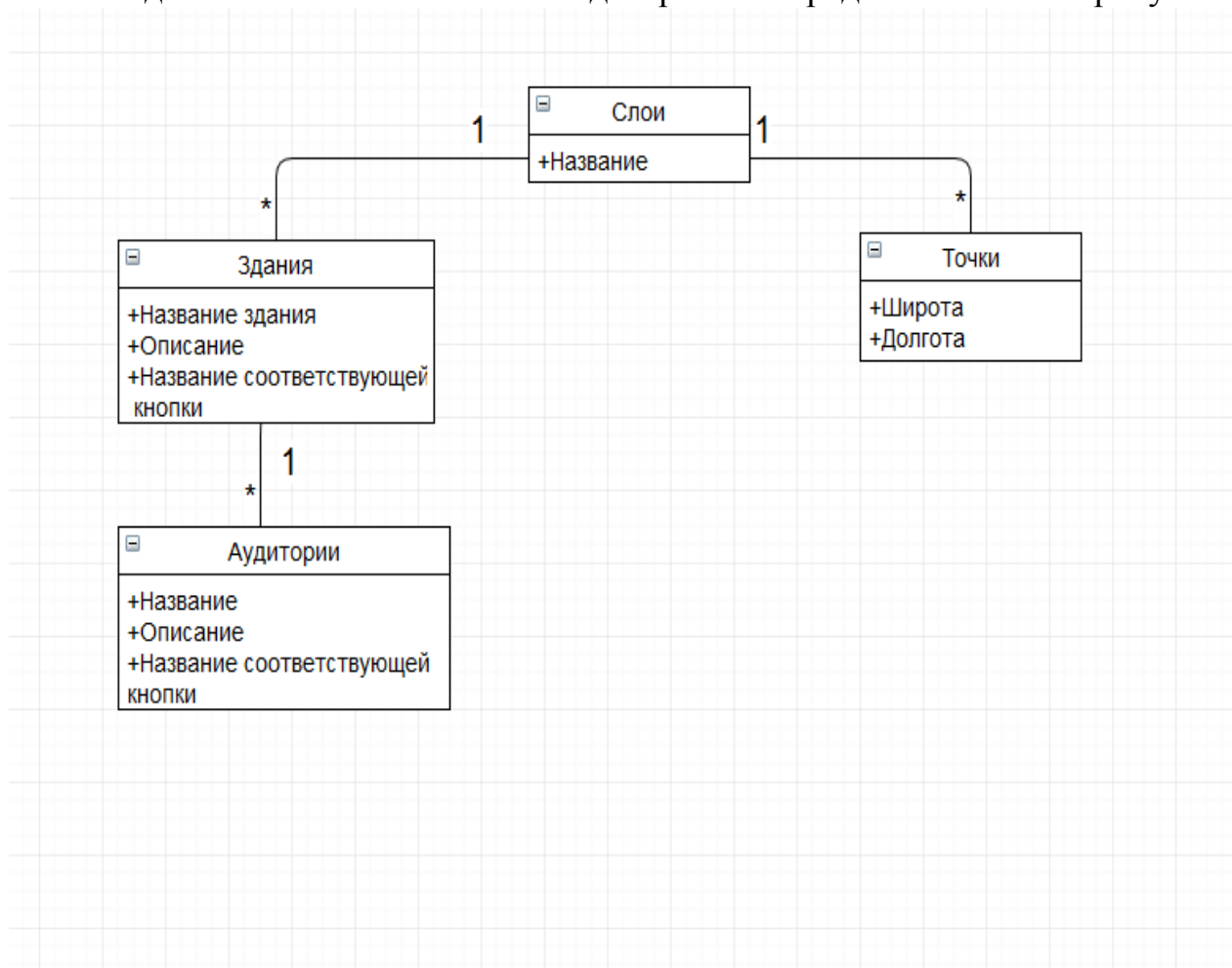


Рисунок 3.1 – «ER-диаграмма базы данных»

### 3.2 Структура связей реляционной базы данных

В этом подразделе описывается структура связей между таблицами построенной базы данных. Для большей наглядности эти связи представлены с помощью диаграммы, изображенной на рисунке 3.2

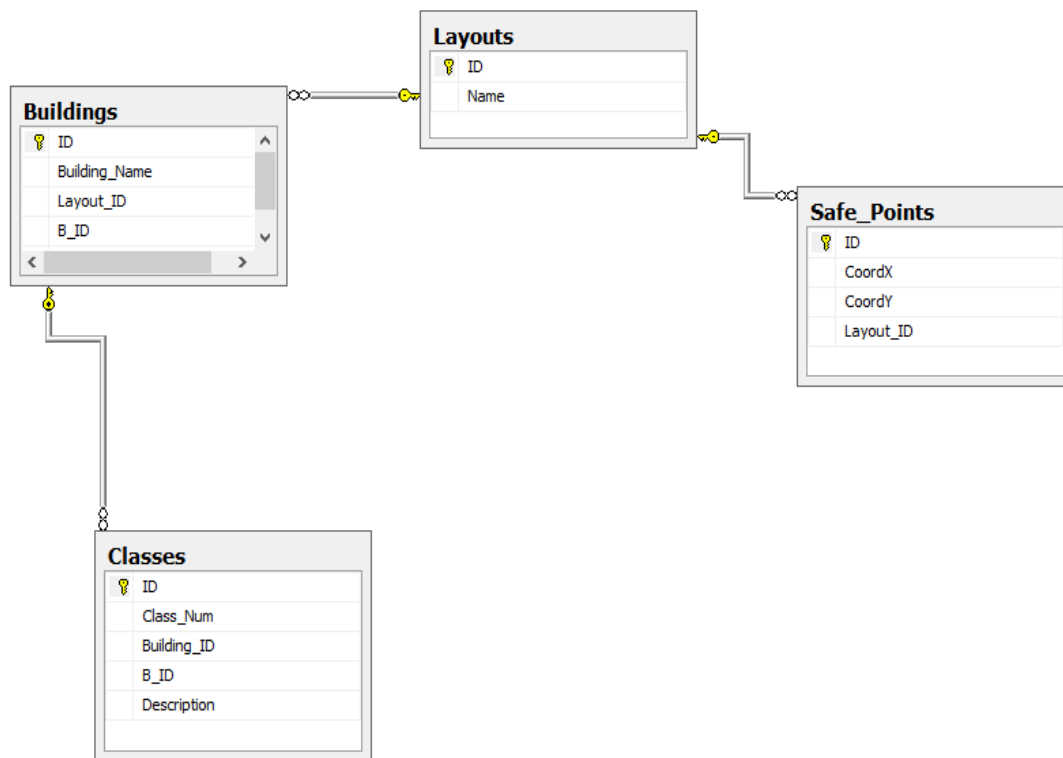


Рисунок 3.2 – «Диаграмма связей реляционной базы данных»

### 3.3 Выявление ограничений и правил поддержания целостности данных

В таблицах 3.1-3.4 представлены назначения и свойства полей базы данных приложения.

Таблица “Layouts”

	Тип данных	Значение по умолчанию	Обязательность	Первичный ключ	Внешний ключ	Ограничения
ID	INTEGER	-	+	+	-	-
Name	TEXT	-	+	-	-	-

В данной таблице хранится информация о существующих макетах в приложении. Поле Name – название макета, существующего в проекте приложения.

Таблица “Buildings”

	Тип данных	Значение по умолчанию	Обязательность	Первичный ключ	Внешний ключ	Ограничения
ID	INTEGER	-	+	+	-	-
Building_Name	TEXT	-	+	-	-	-
Layout_ID	INTEGER	-	+	-	+	-
Description	TEXT	-	+	-	-	-
B_ID	TEXT	-	+	-	-	-

Эта таблица хранит данные о всех зданиях (Building\_Name), их принадлежность к определенному макету (Layout\_ID), описание (Description), а также название элемента, к которому оно относится (B\_ID).



Таблица “Classes”

	Тип данных	Значение по умолчанию	Обязательность	Первичный ключ	Внешний ключ	Ограничения
ID	INTEGER	-	+	+	-	-
Class_Num	TEXT	-	+	-	-	-
Building_ID	INTEGER	-	+	-	+	-
Layout_ID	INTEGER	-	+	-	+	-
Description	TEXT	-	+	-	+	-
B_ID	TEXT	-	+	-	-	-

Эта таблица содержит аудитории (Class\_Num), их принадлежность к определенному макету (Layout\_ID) и зданию (Building\_ID), описание (Description), а также название элемента, к которому оно относится (B\_ID).

Таблица 3.4

Таблица “SafePoints”

	Тип данных	Значение по умолчанию	Обязательность	Первичный ключ	Внешний ключ	Ограничения
ID	INTEGER	-	+	+	-	-
CoordX	NUMERIC	-	+	-	-	-
CoordY	NUMERIC	-	+	-	-	-
Layout_ID	INTEGER	-	+	-	+	-

Данная таблица содержит координаты точек (CoordX, CoordY), их принадлежность к определенному макету (Layout\_ID).

### 3.4 Выводы по разделу

В текущем разделе была разработана ER-диаграмма на основании сущностей и связей базы данных. Затем был осуществлен переход к даталогическому проектированию – было приведено описание и свойства всех таблиц и атрибутов, а также схема связей реляционной базы данных.

## 4 РЕАЛИЗАЦИЯ СИСТЕМЫ

### 4.1 Реализация алгоритмов

#### 4.1.1 Общий алгоритм системы

Общий алгоритм системы начинается с момента инициализации приложения, т.е. с момента его запуска. После чего система ожидает события от пользователя. На рисунке 4.1 приведена схема общего алгоритма системы.

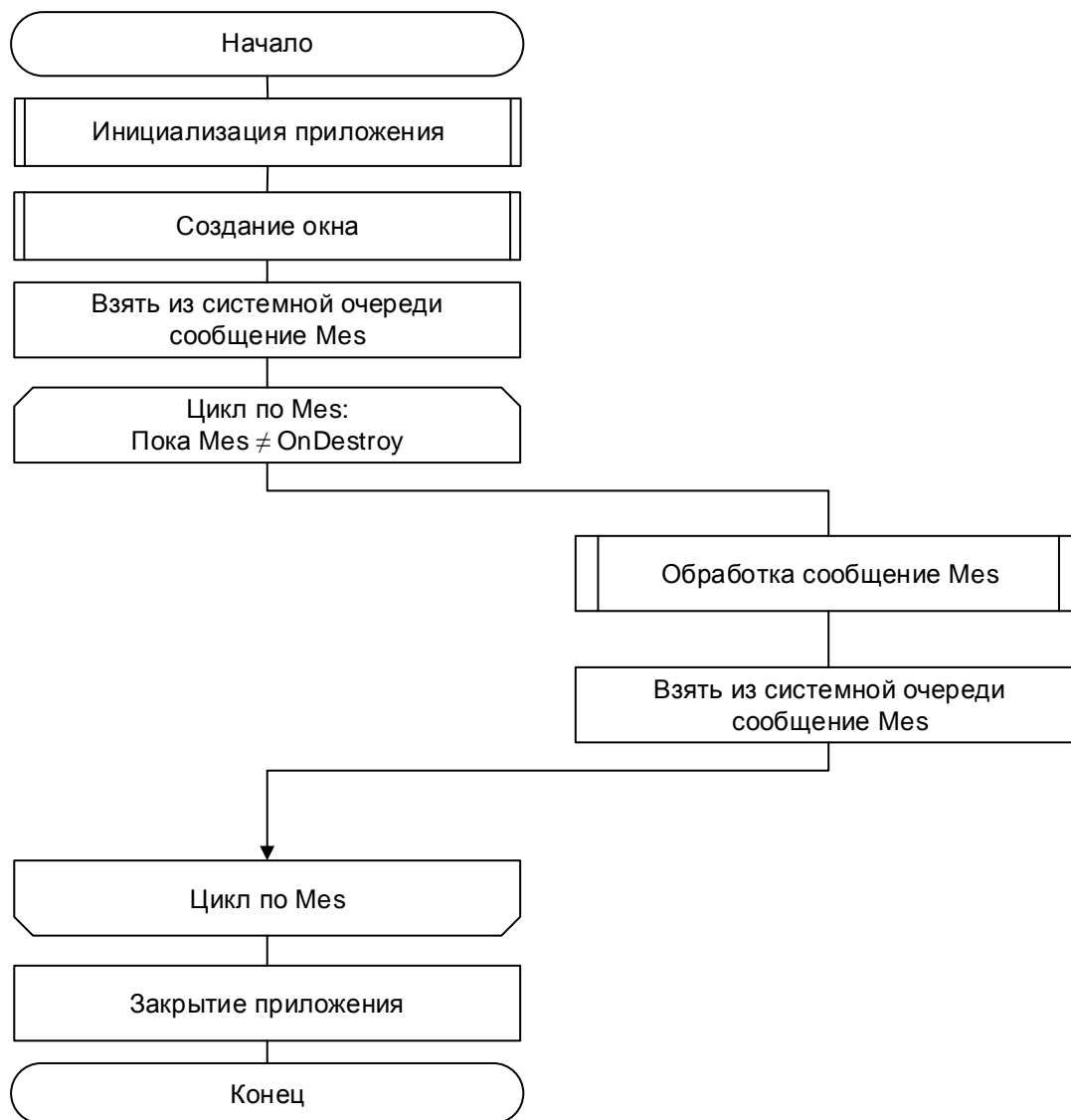


Рисунок 4.1 – «Основной алгоритм системы»

#### 4.1.2 Алгоритм обработки событий окна

На рисунке 4.2 представлен вспомогательный алгоритм обработки событий окна приложения.

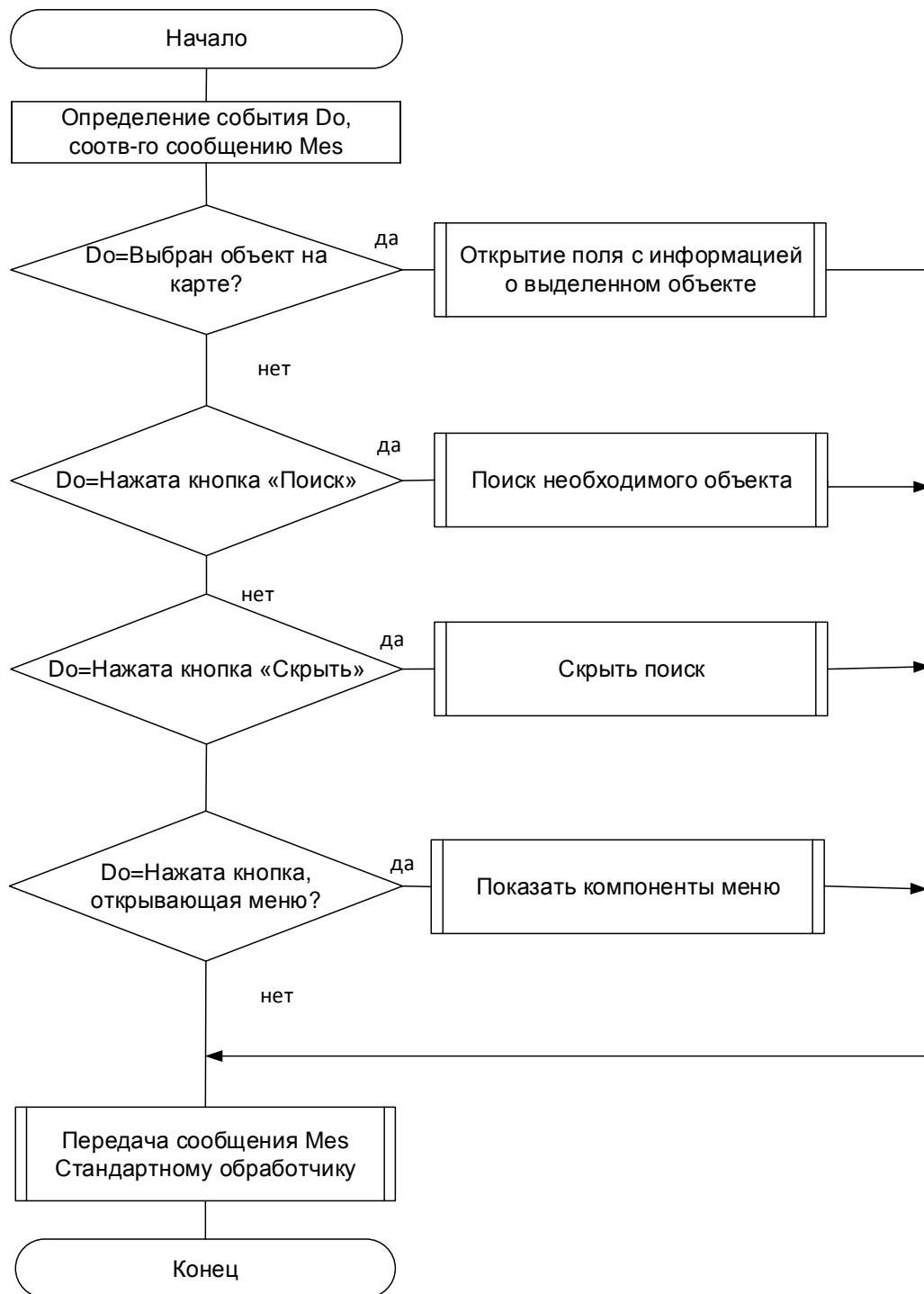


Рисунок 4.1 – «Вспомогательный алгоритм обработки событий окна»

Во время работы программа получает события выполненные пользователем и обрабатывает их в соответствии с алгоритмом.

Если пользователь осуществил нажатие по какому-либо объекту на карте, то происходит вызов соответствующего обработчика и открывается соответствующее поле.

В случае, когда нажата кнопка «Поиск», запускается обработчик данного события и передаются необходимые для дальнейшей работы данные.

В системе также реализована работа с геолокацией. Для этого программа получает информацию о текущем местонахождении пользователя от систем GPS или сетей WI-FI через встроенные функции Android, затем преобразует эти координаты в необходимый приложению вид и на их основании отображает текущее местоположение. Данные собираются с интервалом в 5 секунд.

#### 4.1.3 Алгоритм обработки событий поля с информацией об объекте

Алгоритм на Рисунке 4.3 описывает обработку приложением действий пользователя в окне информации о выделенном объекте.

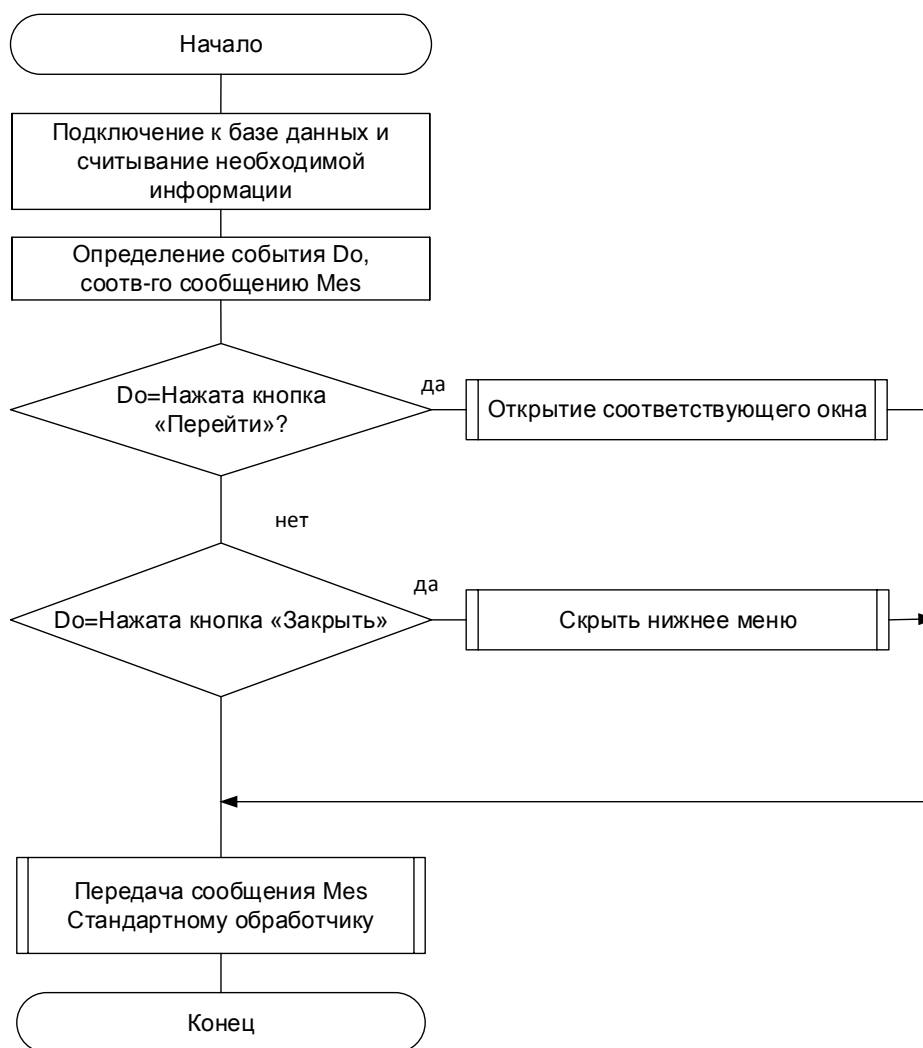


Рисунок 4.3 – «Вспомогательный алгоритм обработки событий поля информации об объекте»

После нажатия пользователем по какому-либо объекту на карте, открывается соответствующее поле, с которым пользователь также может взаимодействовать. В зависимости от типа элемента в данном поле может иметься кнопка, позволяющая перейти внутрь здания, после нажатия которой открывается соответствующее окно.

#### 4.1.4 Алгоритм обработки событий меню приложения

На рисунке 4.4 представлен вспомогательный алгоритм обработки событий меню программы.

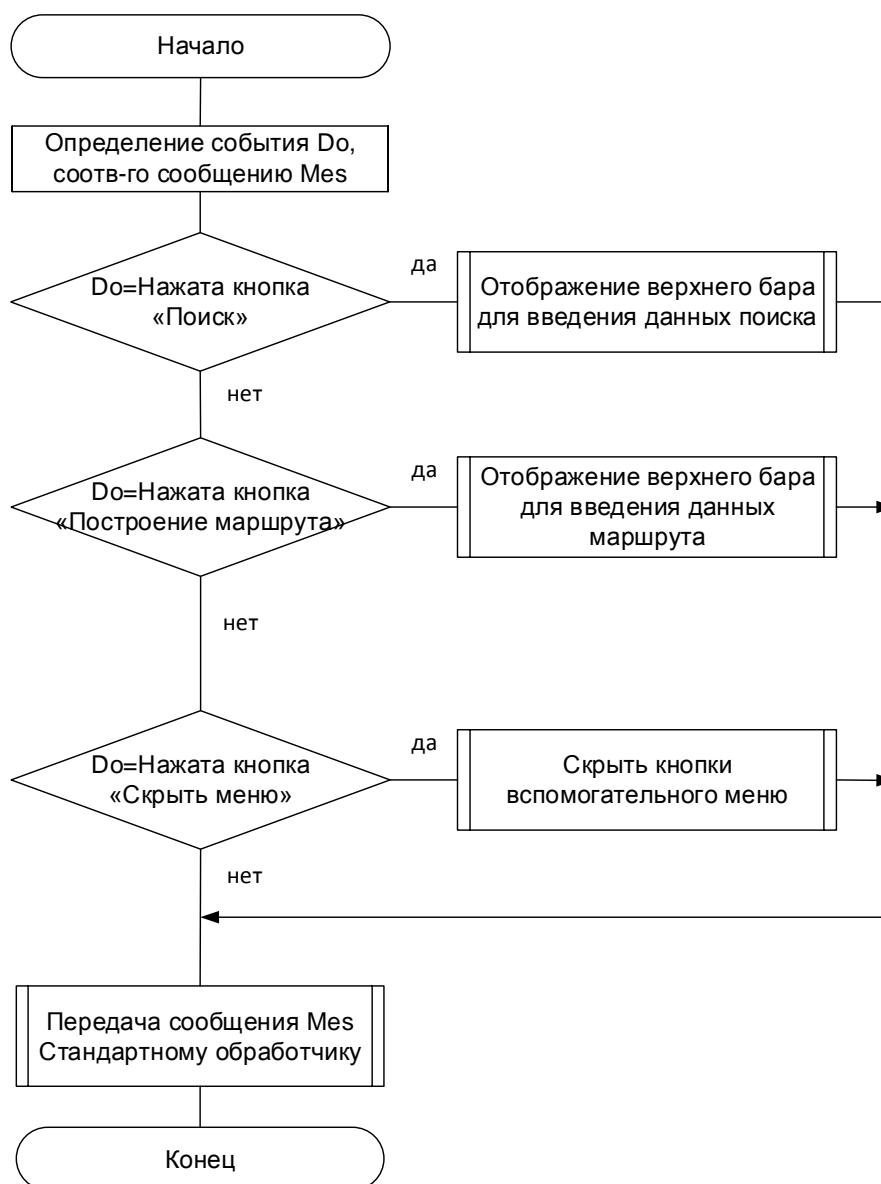


Рисунок 4.4 – «Вспомогательный алгоритм обработки событий меню»

Меню программы содержит кнопки для отображения полей ввода данных для поиска и проложения маршрутов.

#### 4.1.5 Алгоритм осуществления поиска

Рисунок 4.5 содержит вспомогательный алгоритм осуществления поиска введенного объекта.

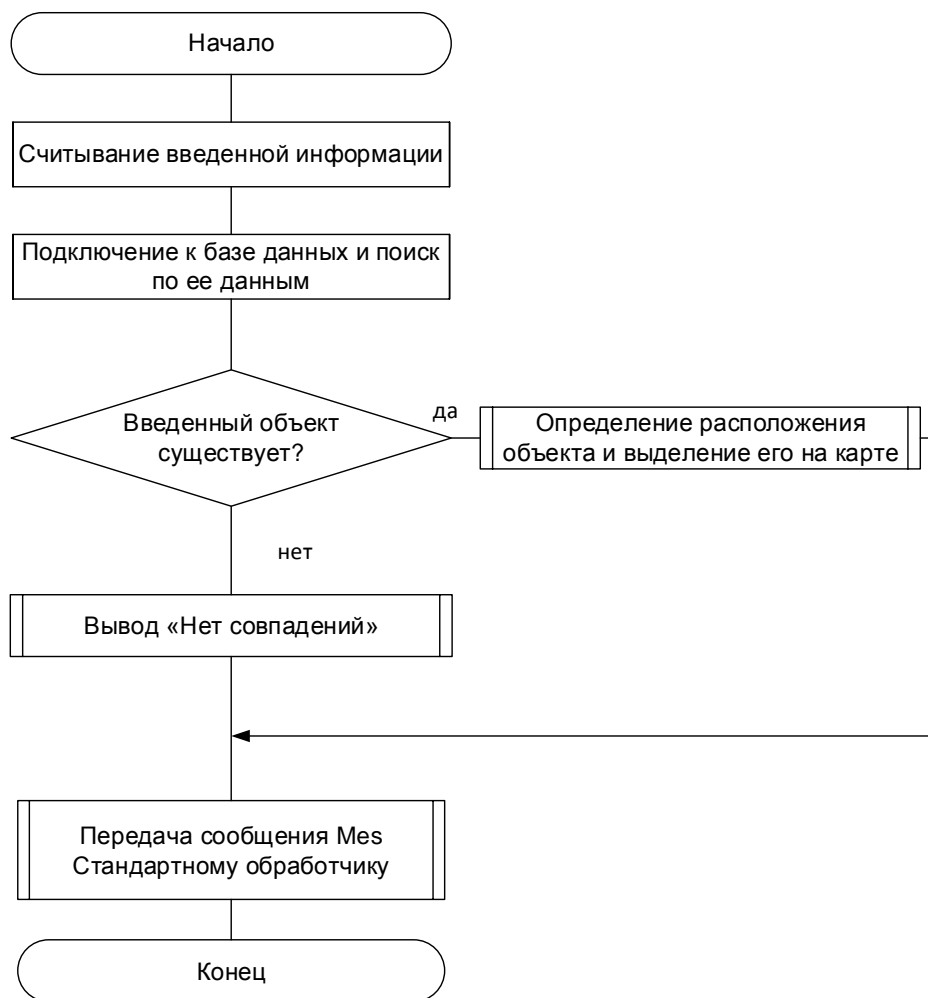


Рисунок 4.5 – «Вспомогательный алгоритм поиска»

Поиск осуществляется по данным, хранящимся в базе данных. Если введенный объект существует, то система получает информацию о его местоположении и отображает пользователю путем выделения данного объекта на карте и открытия поля с соответствующей информацией.

#### 4.1.6 Алгоритм построения маршрута

На рисунке 4.6 изображен вспомогательный алгоритм построения маршрута между двумя введенными точками.

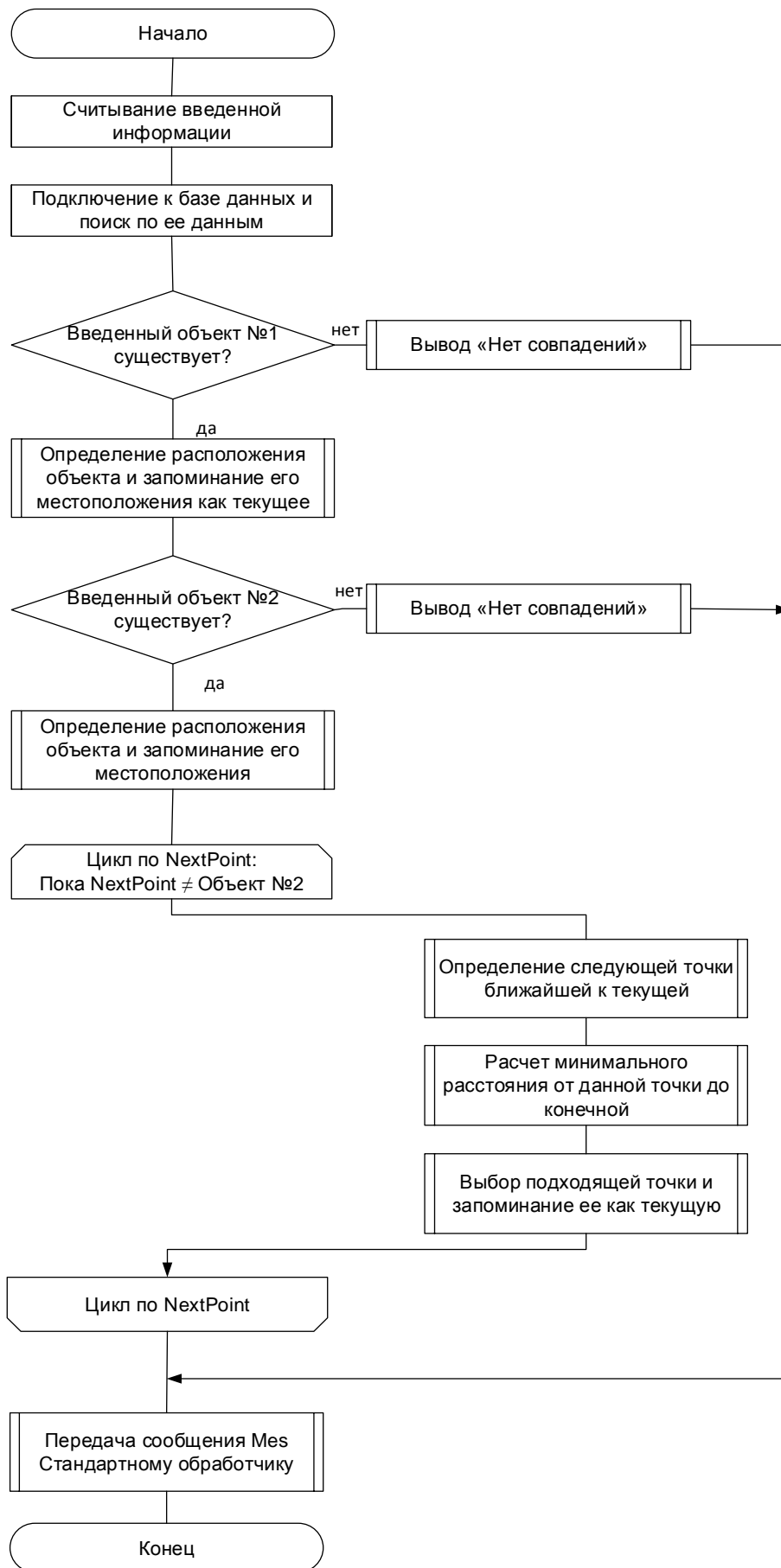


Рисунок 4.6 – «Вспомогательный алгоритм построения маршрута»



Данный алгоритм является реализацией алгоритма Дейкстры, суть которого является в нахождении кратчайшего пути от одной из вершин графа к другой, путем перебора всех возможных направлений движения из одной точки в другую, и построения необходимого маршрута на основании минимального полученного расстояния.

Его использование обусловлено тем, что нам необходимо построить самый оптимальный путь между двумя точками (начальной и конечной), имея в своем распоряжении промежуточные точки, через которые можно «дойти» до нужного объекта.

После запуска данной функции система проверяет наличие введенных объектов в базе данных. Если объекты были найдены, то система передает данные в процедуру, отвечающую за построение маршрута, и осуществляет отображение маршрута.

## 4.2 Проверка работоспособности

После запуска приложения пользователю открывается окно, на котором отображена карта студенческого городка ЮУрГУ. Пример на Рисунке 4.7



Рисунок 4.6 – «Начальное окно после запуска приложения»

В данном окне пользователь может нажать на любой из представленных корпусов. После чего открывается поле с информацией о выделенном объекте и кнопкой для перехода внутрь корпуса.

После нажатия кнопки в появившемся поле открывается другое окно, отображающее внутреннее устройство, где пользователь так же может выбирать объект для получения подробной информации. Пример на Рисунке 4.7

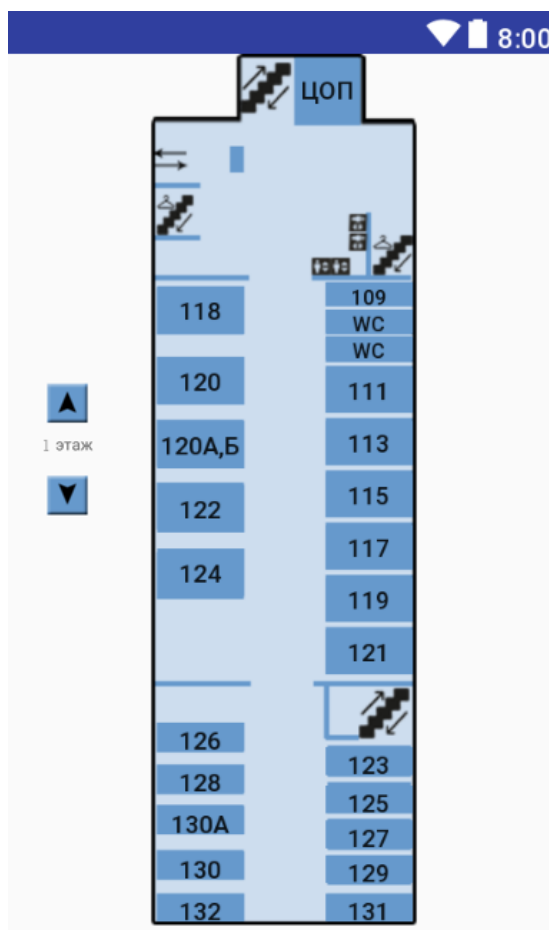


Рисунок 4.7 – «Окно отображения внутреннего устройства корпуса»

Также пользователь может в специальном поле для поиска ввести интересующее его место на карте, и программа автоматически отобразит это место. Пример на Рисунке 4.8

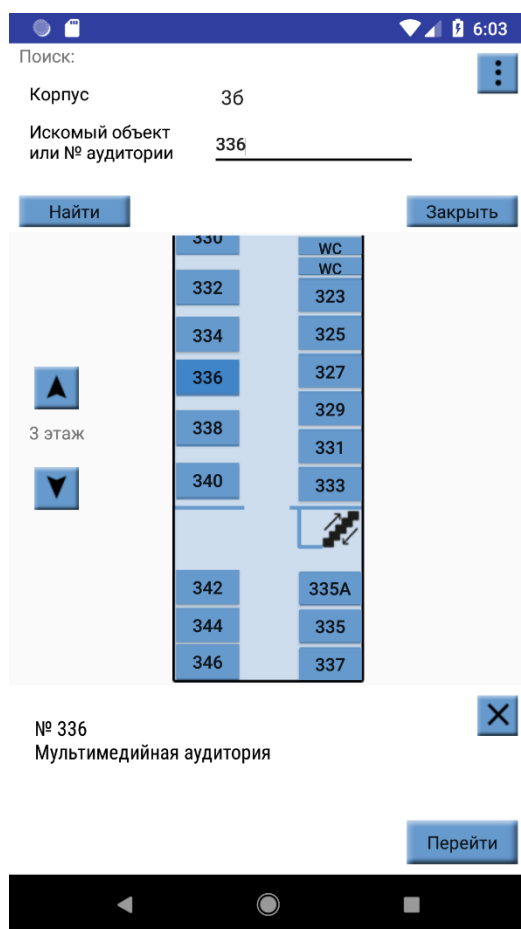


Рисунок 4.8 – «Отображение объекта поиска»

Пользователь может указать места в специальном поле для построения маршрута. Так, например, если пользователю необходимо проложить маршрут от 3б корпуса до 111 аудитории того же корпуса - система отобразит путь, по которому пользователь может до нее добраться. Пример на Рисунке 4.9

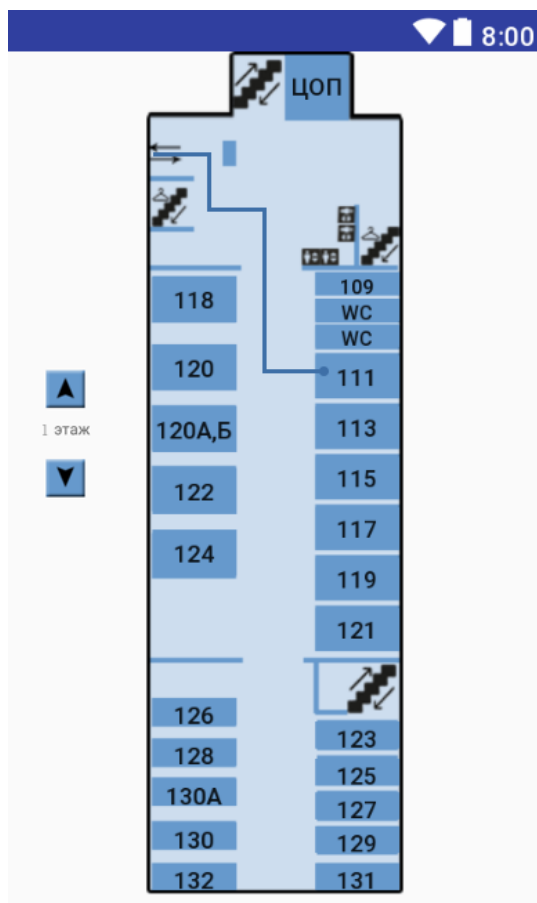


Рисунок 4.9 – «Отображение маршрута между введенными точками»

В данной системе имеется возможность работы с геолокацией. Так, если пользователь находится на территории корпусов ЮУрГУ, его местоположение отображается точкой на карте. Пример на Рисунке 4.10

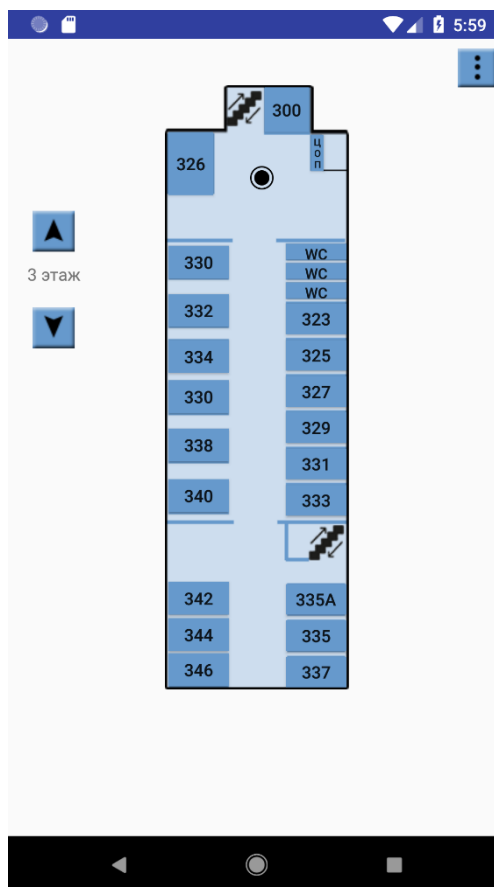


Рисунок 4.10 – «Отображение местоположения пользователя»

#### 4.3 Выводы по разделу

В данном разделе были созданы различные алгоритмы работы системы и его функций. На их основании было разработано само мобильное приложение и приведены примеры его работы. Исходя из этого видно, что приложение работает корректно и все необходимые требования были соблюдены.

## ЗАКЛЮЧЕНИЕ

Данная работа посвящена разработке мобильного приложения для навигации по учебным корпусам ЮУрГУ, которое, в отличие от уже существующих аналогов, позволяет пользователям просматривать и ориентироваться как внутри корпусов, так и снаружи.

Система позволяет осуществлять поиск необходимого места, строить маршрут между двумя указанными точками и определять свое местоположение на территории ВУЗа.

В ходе работы над проектом решены следующие задачи:

- 1) выполнен обзор существующих аналогичных приложений;
- 2) сформулированы функциональные требования к системе;
- 3) разработана архитектура приложения;
- 4) разработаны и реализованы алгоритмы приложения;
- 5) реализовано мобильное приложение для навигации по учебным корпусам ЮУрГУ.

Все поставленные задачи успешно выполнены.

В дальнейшем планируется реализовать большее количество корпусов для навигации по ним, а также функцию для добавления мест в закладки.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Блох, Д. Java: Эффективное программирование. / Д. Блох. – 2-е изд. – М.: «Лори», 2014. – 461 с.
- 2 Буч, Г. Язык UML. Руководство пользователя. / Г. Буч, Д. Рамбо, И. Якобсон. — М.: ДМК Пресс, 2008. — 496 с.
- 3 ГОСТ 19.504-79. Руководство программиста. Требование к содержанию и оформлению. М.: Стандартинформ, 2010. – 2с.
- 4 Дейтел, П. Андроид для разработчиков. / П. Дейтел, Х. Дейтел, А. Уолд. – 3-е изд. – СПб.: Питер, 2016. – 512 с.
- 5 Дмитриев, М. А. Операционная система Android: учебное пособие для вузов. / М. А. Дмитриев, А. В. Зуйков, А. А. Кузин, П. Е. Минин – М: МИФИ, 2012.- 64 с.
- 6 Лафоре, Р. Структуры данных и алгоритмы Java. / Р. Лафоре. – 2-е изд. – СПб.: Питер, 2013. – 704 с.
- 7 Майерс, Г. Д. Искусство тестирования программ. / Г. Д. Майерс. – М.: Финансы и статистика, 1982. – 176 с.
- 8 Новиков, Ф. А. Учебно методическое пособие по дисциплине «Анализ и проектирование на UML». / Ф. А. Новиков. – СПб.: НИУ ИТМО, 2007. – 286с.
- 9 Одиночкина, С. В. Основы технологий XML: учебное пособие. / С. В. Одиночкина. – СПб.: НИУ ИТМО, 2013. – 56 с.
- 10 Пайлон, Д. UML 2 для программистов. / Д. Пайлон, Н. Питмен. – СПб.: Питер, 2012. – 240 с.
- 11 Ревунков, Г. И. Проектирование баз данных. / Г. И. Ревунков – М.: МГТУ им. Н.Э. Баумана, 2009. – 20с.
- 12 Тарасов, С. В. СУБД для программиста. Базы данных изнутри. / С. В. Тарасов. – М.: СОЛОН-Пресс, 2015. – 320 с.
- 13 Фелкер, Д. Android: Разработка приложений для чайников. / Д. Фелкер. – М.: - ООО"И.Д.Вильямс", 2012. – 336 с.
- 14 Харди, Б. Android программирование для профессионалов. / Б. Харди, Б. Филлипс, К. Стюарт, К. Марсикано. – 2-е изд. – СПб.: Питер, 2016. – 640 с.
- 15 Хантер, Д. XML. Базовый курс. / Д. Хантер, Д. Рафтер, Д. Фосетт, Д. Айерс, Д. Дакетт , Э. Уатт, Э. ван дер Влишт. – 4-е изд. – М.: - ООО"И.Д.Вильямс, 2018. – 1344 с.
- 16 Шилдт, Г. Java 8. Полное руководство. / Г. Шилдт. – 9-е изд. – М.: - ООО"И.Д.Вильямс", 2015. – 1376 с.
- 17 Documentation for app developers [Электронный ресурс]. – Режим доступа: <https://developer.android.com/docs>. – Заглавие с экрана. – (Дата обращения 22.04.2018).

- 18 Sibsankar, H. SQLite Database System Design and Implementation. / H. Sibsankar. – 2-е изд. – O'Reilly, 2015. – 256 с.
- 19 Start Android - учебник по Android для начинающих и продвинутых [Электронный ресурс]. – Режим доступа: <http://startandroid.ru/html>. – Заглавие с экрана. - (Дата обращения: 17.04.2018).
- 20 Sunny, K. A. Android SQLite Essentials. / K. A. Sunny, K. K. Vikash. – Packt Publishing, 2014. – 110 с.



## ПРИЛОЖЕНИЕ 1 ТЕКСТ ПРОГРАММЫ

### AndroidManifest.xml - конфиг-файл приложения

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example">
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".Main">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".activity_a3" >
            <intent-filter>
                <action android:name="android.intent.action.activity_a3" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity android:name=".activity_a2" >
            <intent-filter>
                <action android:name="android.intent.action.activity_a2" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity android:name=".activity_3b">
            <intent-filter>
                <action android:name="android.intent.action.activity_3b" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity android:name=".activity_3b_2">
            <intent-filter>
                <action android:name="android.intent.action.activity_3b_2" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity android:name=".activity_3b_3">
            <intent-filter>
                <action android:name="android.intent.action.activity_3b_3" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity android:name=".activity_3b_4">
            <intent-filter>
                <action android:name="android.intent.action.activity_3b_4" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity android:name=".activity_3b_5">
            <intent-filter>
                <action android:name="android.intent.action.activity_3b_5" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

```

        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
<activity android:name=".activity_3b_6">
    <intent-filter>
        <action android:name="android.intent.action.activity_3b_6" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
<activity android:name=".activity_3b_7">
    <intent-filter>
        <action android:name="android.intent.action.activity_3b_7" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
<activity android:name=".activity_3b_8">
    <intent-filter>
        <action android:name="android.intent.action.activity_3b_8" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
<activity android:name=".activity_3b_9">
    <intent-filter>
        <action android:name="android.intent.action.activity_3b_9" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
<activity android:name=".activity_3b_10">
    <intent-filter>
        <action android:name="android.intent.action.activity_3b_10" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
<activity android:name=".activity_g3" >
    <intent-filter>
        <action android:name="android.intent.action.activity_g3" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
<activity android:name=".activity_a6" >
    <intent-filter>
        <action android:name="android.intent.action.activity_a6" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
<activity android:name=".activity_guk" >
    <intent-filter>
        <action android:name="android.intent.action.activity_guk" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
<activity android:name=".activity_usk" >
    <intent-filter>
        <action android:name="android.intent.action.activity_usk" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
<activity android:name=".activity_a4">
    <intent-filter>
        <action android:name="android.intent.action.activity_a4" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>

```

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

## Activity\_main.xml – содержит все элементы View основного окна

```
<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/conLayout"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.Main">

    <android.support.constraint.ConstraintLayout
        android:id="@+id/layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <android.support.constraint.ConstraintLayout
            android:id="@+id/CL"
            android:layout_width="wrap_content"
            android:layout_height="match_parent">

            <ScrollView
                android:id="@+id/scroll"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                app:layout_constraintEnd_toEndOf="parent"

                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent">

                <HorizontalScrollView
                    android:id="@+id/horizontalScrollView"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    app:layout_constraintLeft_toLeftOf="parent"
                    app:layout_constraintTop_toTopOf="parent">

                    <android.support.constraint.ConstraintLayout
                        android:id="@+id/conLay"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:minHeight="0dp"
                        android:visibility="visible">

                        <ImageView
                            android:id="@+id/map"
                            android:layout_width="540dp"
                            android:layout_height="540dp"
                            android:visibility="visible"
                            app:layout_constraintBottom_toBottomOf="parent"
                            app:layout_constraintEnd_toEndOf="parent"
                            app:layout_constraintHorizontal_bias="1.0"
```

```

        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0"
        app:srcCompat="@drawable/susu2" />

<android.support.constraint.ConstraintLayout
    android:id="@+id/pos_lay"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:visibility="visible"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <ImageView
        android:id="@+id/pos"
        android:layout_width="15dp"
        android:layout_height="15dp"
        android:visibility="invisible"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0"
        app:srcCompat="@drawable/pos" />
</android.support.constraint.ConstraintLayout>

<Button
    android:id="@+id/b3"
    android:layout_width="19dp"
    android:layout_height="86dp"
    android:layout_marginLeft="66dp"
    android:layout_marginStart="66dp"
    android:layout_marginTop="321dp"
    android:background="@color/colorBuilding"
    android:text="36"
    android:visibility="visible"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/g3"
    android:layout_width="32dp"
    android:layout_height="64dp"
    android:layout_marginLeft="144dp"
    android:layout_marginStart="144dp"
    android:layout_marginTop="344dp"
    android:background="@drawable/border"
    android:text="3r"
    android:visibility="visible"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/map" />

<Button
    android:id="@+id/aaa2"
    android:layout_width="55dp"
    android:layout_height="12dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="292dp"
    android:layout_marginRight="292dp"

```

```

        android:layout_marginTop="321dp"
        android:background="@color/colorBuilding"
        android:text="2"

    android:textAppearance="@style/TextAppearance.AppCompat.Button"
    android:textColor="@color/colorBuilding"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="@+id/map"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="@+id/map"
    app:layout_constraintVertical_bias="0.0" />

```

```

<Button
    android:id="@+id/aa2"
    android:layout_width="13dp"
    android:layout_height="36dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="125dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="125dp"
    android:layout_marginTop="338dp"
    android:background="@color/colorBuilding"
    android:text="2"
    android:textColor="@color/colorBuilding"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="@+id/map"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toEndOf="@+id/g3"
    app:layout_constraintTop_toTopOf="@+id/map"
    app:layout_constraintVertical_bias="0.0" />

```

```

<Button
    android:id="@+id/a2"
    android:layout_width="62.5dp"
    android:layout_height="16dp"
    android:layout_marginEnd="232dp"
    android:layout_marginRight="232dp"
    android:layout_marginTop="321dp"
    android:background="@color/colorBuilding"
    android:text="2"
    android:visibility="visible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="@+id/map" />

```

```

<Button
    android:id="@+id/a6"
    android:layout_width="64dp"
    android:layout_height="20dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="166dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="166dp"
    android:layout_marginTop="416dp"
    android:background="@drawable/border"
    android:text="6"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/aa2"
    app:layout_constraintHorizontal_bias="0.0"

```

```
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.0" />
```

```
<Button  
    android:id="@+id/usk2"  
    android:layout_width="21dp"  
    android:layout_height="59dp"  
    android:layout_marginBottom="8dp"  
    android:layout_marginEnd="8dp"  
    android:layout_marginLeft="139dp"  
    android:layout_marginRight="8dp"  
    android:layout_marginStart="139dp"  
    android:layout_marginTop="415dp"  
    android:background="@color/colorBuilding"  
    android:text="YCK"  
    android:textColor="@color/colorBuilding"  
    android:visibility="visible"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="@+id/usk"  
    app:layout_constraintHorizontal_bias="0.0"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.0" />
```

```
<Button  
    android:id="@+id/usk"  
    android:layout_width="72dp"  
    android:layout_height="31dp"  
    android:layout_marginLeft="173dp"  
    android:layout_marginStart="173dp"  
    android:layout_marginTop="440dp"  
    android:background="@color/colorBuilding"  
    android:elevation="0dp"  
    android:text="YCK"  
    android:visibility="visible"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button  
    android:id="@+id/a4"  
    android:layout_width="35dp"  
    android:layout_height="22dp"  
    android:layout_marginEnd="8dp"  
    android:layout_marginLeft="144dp"  
    android:layout_marginRight="8dp"  
    android:layout_marginStart="144dp"  
    android:layout_marginTop="64dp"  
    android:background="@drawable/border"  
    android:text="4"  
    android:visibility="visible"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.0"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button  
    android:id="@+id/guk"  
    android:layout_width="32dp"  
    android:layout_height="33dp"  
    android:layout_marginBottom="8dp"
```

```
android:layout_marginEnd="8dp"
android:layout_marginLeft="348dp"
android:layout_marginRight="8dp"
android:layout_marginStart="348dp"
android:layout_marginTop="155dp"
android:background="@color/colorBuilding"
android:text="1"
android:visibility="visible"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="@+id/map"
app:layout_constraintTop_toTopOf="@+id/map"
app:layout_constraintVertical_bias="0.0" />
```

<Button

```
android:id="@+id/guk2"
android:layout_width="13dp"
android:layout_height="50dp"
android:layout_marginBottom="8dp"
android:layout_marginEnd="8dp"
android:layout_marginLeft="261dp"
android:layout_marginRight="8dp"
android:layout_marginStart="261dp"
android:layout_marginTop="187dp"
android:background="@color/colorBuilding"
android:text="1"
android:textColor="@color/colorBuilding"
android:visibility="visible"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.0" />
```

<Button

```
android:id="@+id/guk3"
android:layout_width="13dp"
android:layout_height="50dp"
android:layout_marginBottom="8dp"
android:layout_marginEnd="8dp"
android:layout_marginLeft="458dp"
android:layout_marginRight="8dp"
android:layout_marginStart="458dp"
android:layout_marginTop="187dp"
android:background="@color/colorBuilding"
android:text="1"
android:textColor="@color/colorBuilding"
android:visibility="visible"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.0" />
```

<Button

```
android:id="@+id/guk4"
android:layout_width="13dp"
```

```

android:layout_height="50dp"
android:layout_marginBottom="8dp"
android:layout_marginEnd="8dp"
android:layout_marginLeft="311dp"
android:layout_marginRight="8dp"
android:layout_marginStart="311dp"
android:layout_marginTop="187dp"
android:background="@color/colorBuilding"
android:text="1"
android:textColor="@color/colorBuilding"
android:visibility="visible"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.0" />

```

<Button

```

android:id="@+id/guk5"
android:layout_width="13dp"
android:layout_height="50dp"
android:layout_marginBottom="8dp"
android:layout_marginEnd="8dp"
android:layout_marginLeft="408dp"
android:layout_marginRight="8dp"
android:layout_marginStart="408dp"
android:layout_marginTop="187dp"
android:background="@color/colorBuilding"
android:text="1"
android:textColor="@color/colorBuilding"
android:visibility="visible"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.0" />

```

<Button

```

android:id="@+id/guk6"
android:layout_width="36dp"
android:layout_height="12dp"
android:layout_marginBottom="8dp"
android:layout_marginEnd="8dp"
android:layout_marginLeft="276dp"
android:layout_marginRight="8dp"
android:layout_marginStart="276dp"
android:layout_marginTop="224dp"
android:background="@color/colorBuilding"
android:text="1"
android:textColor="@color/colorBuilding"
android:visibility="visible"
app:layout_constraintBottom_toBottomOf="@+id/map"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.0" />

```



```

<Button
    android:id="@+id/guk7"
    android:layout_width="36dp"
    android:layout_height="12dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="420dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="420dp"
    android:layout_marginTop="224dp"
    android:background="@color/colorBuilding"
    android:text="1"
    android:textColor="@color/colorBuilding"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />

```

```

<Button
    android:id="@+id/guk8"
    android:layout_width="82dp"
    android:layout_height="16dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="325dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="325dp"
    android:layout_marginTop="187dp"
    android:background="@color/colorBuilding"
    android:text="1"
    android:textColor="@color/colorBuilding"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />

```

```

<Button
    android:id="@+id/a3"
    android:layout_width="105dp"
    android:layout_height="17dp"
    android:layout_marginLeft="86dp"
    android:layout_marginStart="86dp"
    android:layout_marginTop="325dp"
    android:background="@color/colorBuilding"
    android:text="3a"
    android:visibility="visible"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

```

</android.support.constraint.ConstraintLayout>

```

```

</HorizontalScrollView>
</ScrollView>

```

```

</android.support.constraint.ConstraintLayout>

<android.support.constraint.ConstraintLayout
    android:id="@+id/btm"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    </android.support.constraint.ConstraintLayout>
</android.support.constraint.ConstraintLayout>

</android.support.constraint.ConstraintLayout>

```

## Activity\_3b.xml – содержит все элементы View окна одного из корпусов

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/b3_layout_1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.activity_3b">

    <android.support.constraint.ConstraintLayout
        android:id="@+id/mapCL_1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_constraintTop_toTopOf="parent">

        <android.support.constraint.ConstraintLayout
            android:id="@+id/lay_1"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            app:layout_constraintTop_toTopOf="parent">

            <TextView
                android:id="@+id/textView9"
                android:layout_width="wrap_content"
                android:layout_height="19dp"
                android:layout_marginLeft="16dp"
                android:layout_marginStart="16dp"
                android:layout_marginTop="8dp"
                android:text="1 этаж"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/go_up" />

            <Button
                android:id="@+id/go_up"
                android:layout_width="35dp"
                android:layout_height="34dp"
                android:layout_marginLeft="20dp"
                android:layout_marginStart="20dp"
                android:layout_marginTop="104dp"
                android:background="@drawable/up"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent" />

            <ScrollView
                android:layout_width="match_parent"

```

```

android:layout_height="wrap_content"
app:layout_constraintTop_toTopOf="parent">

<FrameLayout
    android:id="@+id/flay_1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal">

    <android.support.constraint.ConstraintLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            app:srcCompat="@drawable/b3_1" />

        <android.support.constraint.ConstraintLayout
            android:id="@+id/pos_lay_b3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:visibility="visible"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent">

            <ImageView
                android:id="@+id/pos_b3"
                android:layout_width="15dp"
                android:layout_height="15dp"
                android:visibility="invisible"
                app:layout_constraintBottom_toBottomOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintHorizontal_bias="0.0"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintVertical_bias="0.0"
                app:srcCompat="@drawable/pos" />
        </android.support.constraint.ConstraintLayout>

        <Button
            android:id="@+id/cop1"
            android:layout_width="36dp"
            android:layout_height="37dp"
            android:layout_marginEnd="33dp"
            android:layout_marginRight="33dp"
            android:layout_marginTop="2dp"
            android:background="@color/colorBuilding"
            android:text="ЦОП"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <Button
            android:id="@+id/b118"
            android:layout_width="50dp"
            android:layout_height="27dp"
            android:layout_marginLeft="3dp"
            android:layout_marginStart="3dp"

```

```

        android:layout_marginTop="133dp"
        android:background="@color/colorBuilding"
        android:text="118"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/b120"
    android:layout_width="50dp"
    android:layout_height="27dp"
    android:layout_marginLeft="3dp"
    android:layout_marginStart="3dp"
    android:layout_marginTop="13dp"
    android:background="@color/colorBuilding"
    android:text="120"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/b118" />

<Button
    android:id="@+id/b120ab"
    android:layout_width="50dp"
    android:layout_height="27dp"
    android:layout_marginLeft="3dp"
    android:layout_marginStart="3dp"
    android:layout_marginTop="9dp"
    android:background="@color/colorBuilding"
    android:text="120a,6"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/b120" />

<Button
    android:id="@+id/b122"
    android:layout_width="50dp"
    android:layout_height="27dp"
    android:layout_marginLeft="3dp"
    android:layout_marginStart="3dp"
    android:layout_marginTop="9dp"
    android:background="@color/colorBuilding"
    android:text="122"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/b120ab" />

<Button
    android:id="@+id/b124"
    android:layout_width="50dp"
    android:layout_height="27dp"
    android:layout_marginLeft="3dp"
    android:layout_marginStart="3dp"
    android:layout_marginTop="9dp"
    android:background="@color/colorBuilding"
    android:text="124"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/b122" />

<Button
    android:id="@+id/b126"
    android:layout_width="50dp"
    android:layout_height="16dp"
    android:layout_marginBottom="8dp"
    android:layout_marginLeft="3dp"
    android:layout_marginStart="3dp"
    android:background="@color/colorBuilding"

```

```

        android:text="126"
        app:layout_constraintBottom_toTopOf="@+id/b128"
        app:layout_constraintStart_toStartOf="parent" />

<Button
    android:id="@+id/b130a"
    android:layout_width="50dp"
    android:layout_height="16dp"
    android:layout_marginBottom="10dp"
    android:layout_marginLeft="3dp"
    android:layout_marginStart="3dp"
    android:background="@color/colorBuilding"
    android:text="130a"
    app:layout_constraintBottom_toTopOf="@+id/b130"
    app:layout_constraintStart_toStartOf="parent" />

<Button
    android:id="@+id/b128"
    android:layout_width="50dp"
    android:layout_height="16dp"
    android:layout_marginBottom="8dp"
    android:layout_marginLeft="3dp"
    android:layout_marginStart="3dp"
    android:background="@color/colorBuilding"
    android:text="128"
    app:layout_constraintBottom_toTopOf="@+id/b130a"
    app:layout_constraintStart_toStartOf="parent" />

<Button
    android:id="@+id/b130"
    android:layout_width="50dp"
    android:layout_height="16dp"
    android:layout_marginBottom="26dp"
    android:layout_marginLeft="3dp"
    android:layout_marginStart="3dp"
    android:background="@color/colorBuilding"
    android:text="130"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

<Button
    android:id="@+id/b132"
    android:layout_width="50dp"
    android:layout_height="16dp"
    android:layout_marginBottom="2dp"
    android:layout_marginLeft="3dp"
    android:layout_marginStart="3dp"
    android:background="@color/colorBuilding"
    android:text="132"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

<Button
    android:id="@+id/b131"
    android:layout_width="50dp"
    android:layout_height="16dp"
    android:layout_marginBottom="2dp"
    android:layout_marginEnd="2dp"
    android:layout_marginRight="2dp"
    android:background="@color/colorBuilding"
    android:text="131"
    android:visibility="visible"

```

```

        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

<Button
    android:id="@+id/b129"
    android:layout_width="50dp"
    android:layout_height="16dp"
    android:layout_marginBottom="24dp"
    android:layout_marginEnd="2dp"
    android:layout_marginRight="2dp"
    android:background="@color/colorBuilding"
    android:text="129"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

<Button
    android:id="@+id/b127"
    android:layout_width="50dp"
    android:layout_height="16dp"
    android:layout_marginBottom="44dp"
    android:layout_marginEnd="2dp"
    android:layout_marginRight="2dp"
    android:background="@color/colorBuilding"
    android:text="127"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

<Button
    android:id="@+id/b125"
    android:layout_width="50dp"
    android:layout_height="16dp"
    android:layout_marginBottom="64dp"
    android:layout_marginEnd="2dp"
    android:layout_marginRight="2dp"
    android:background="@color/colorBuilding"
    android:text="125"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

<Button
    android:id="@+id/b123"
    android:layout_width="50dp"
    android:layout_height="16dp"
    android:layout_marginBottom="86dp"
    android:layout_marginEnd="2dp"
    android:layout_marginRight="2dp"
    android:background="@color/colorBuilding"
    android:text="123"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

<Button
    android:id="@+id/b121"
    android:layout_width="50dp"
    android:layout_height="25dp"
    android:layout_marginBottom="144dp"
    android:layout_marginEnd="2dp"
    android:layout_marginRight="2dp"

```

```
android:background="@color/colorBuilding"  
android:text="121"  
android:visibility="visible"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent" />
```

```
<Button  
    android:id="@+id/b119"  
    android:layout_width="50dp"  
    android:layout_height="25dp"  
    android:layout_marginBottom="174dp"  
    android:layout_marginEnd="2dp"  
    android:layout_marginRight="2dp"  
    android:background="@color/colorBuilding"  
    android:text="119"  
    android:visibility="visible"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent" />
```

```
<Button  
    android:id="@+id/b117"  
    android:layout_width="50dp"  
    android:layout_height="25dp"  
    android:layout_marginBottom="204dp"  
    android:layout_marginEnd="2dp"  
    android:layout_marginRight="2dp"  
    android:background="@color/colorBuilding"  
    android:text="117"  
    android:visibility="visible"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent" />
```

```
<Button  
    android:id="@+id/b115"  
    android:layout_width="50dp"  
    android:layout_height="25dp"  
    android:layout_marginBottom="234dp"  
    android:layout_marginEnd="2dp"  
    android:layout_marginRight="2dp"  
    android:background="@color/colorBuilding"  
    android:text="115"  
    android:visibility="visible"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent" />
```

```
<Button  
    android:id="@+id/b113"  
    android:layout_width="50dp"  
    android:layout_height="25dp"  
    android:layout_marginBottom="264dp"  
    android:layout_marginEnd="2dp"  
    android:layout_marginRight="2dp"  
    android:background="@color/colorBuilding"  
    android:text="113"  
    android:visibility="visible"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent" />
```

```
<Button  
    android:id="@+id/b111"  
    android:layout_width="50dp"  
    android:layout_height="25dp"
```

```

        android:layout_marginBottom="294dp"
        android:layout_marginEnd="2dp"
        android:layout_marginRight="2dp"
        android:background="@color/colorBuilding"
        android:text="111"
        android:visibility="visible"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

<Button
    android:id="@+id/b109"
    android:layout_width="50dp"
    android:layout_height="13dp"
    android:layout_marginEnd="2dp"
    android:layout_marginRight="2dp"
    android:layout_marginTop="131dp"
    android:background="@color/colorBuilding"
    android:text="109"
    android:textSize="12sp"
    android:visibility="visible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/tb12"
    android:layout_width="50dp"
    android:layout_height="13dp"
    android:layout_marginEnd="2dp"
    android:layout_marginRight="2dp"
    android:layout_marginTop="2dp"
    android:background="@color/colorBuilding"
    android:text="WC"
    android:textSize="12sp"
    android:visibility="visible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/b109" />

<Button
    android:id="@+id/tb13"
    android:layout_width="50dp"
    android:layout_height="13dp"
    android:layout_marginEnd="2dp"
    android:layout_marginRight="2dp"
    android:layout_marginTop="2dp"
    android:background="@color/colorBuilding"
    android:text="WC"
    android:textSize="12sp"
    android:visibility="visible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tb12" />
</android.support.constraint.ConstraintLayout>

</FrameLayout>
</ScrollView>
</android.support.constraint.ConstraintLayout>

</android.support.constraint.ConstraintLayout>

</android.support.constraint.ConstraintLayout>

```

**bottom\_bar.xml** - содержит view элементы поля с информацией об объекте



```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/bottombar_lay"
android:layout_width="match_parent"
android:layout_height="match_parent">

<TextView
    android:id="@+id/btmBar"
    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:background="@color/colorWhite"
    android:fontFamily="sans-serif-condensed"
    android:gravity="start"
    android:inputType="textPersonName"
    android:paddingLeft="20dp"
    android:paddingTop="15dp"
    android:textAlignment="textStart"
    android:textColor="@color/Black"
    android:textSize="20sp"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent" />

<Button
    android:id="@+id/close"
    android:layout_width="32dp"
    android:layout_height="32dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginTop="8dp"
    android:background="@drawable/cl12"
    android:visibility="visible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintRight_toRightOf="@+id/btmBar"
    app:layout_constraintTop_toTopOf="@+id/btmBar" />

<Button
    android:id="@+id/goIn"
    android:layout_width="wrap_content"
    android:layout_height="35dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:background="@drawable/but"
    android:text="Перейти"
    android:textAppearance="@style/TextAppearance.AppCompat"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintRight_toRightOf="@+id/btmBar" />

<TextView
    android:id="@+id/info"
    android:layout_width="276dp"
    android:layout_height="98dp"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    android:background="@color/colorWhite"
    android:enabled="false"

```

```

    android:fontFamily="sans-serif-condensed"
    android:gravity="start"
    android:inputType="textMultiLine"
    android:lines="4"
    android:maxLines="4"
    android:paddingLeft="20dp"
    android:paddingTop="15dp"
    android:textAlignment="textStart"
    android:textColor="@color/Black"
    android:textSize="16sp"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="@+id/btmBar"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/btmBar"
    app:layout_constraintVertical_bias="0.0" />

```

```
</android.support.constraint.ConstraintLayout>
```

## **menu\_btns.xml - СОДЕРЖИТ VIEW ЭЛЕМЕНТЫ ЭЛЕМЕНТОВ МЕНЮ**

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/btn_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/ClmenuBtn"
        android:layout_width="32dp"
        android:layout_height="32dp"
        android:layout_marginEnd="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:background="@drawable/cl2"
        android:visibility="invisible"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/findmenuBtn"
        android:layout_width="32dp"
        android:layout_height="32dp"
        android:layout_marginEnd="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:background="@drawable/menu1"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/routeBtn"
        android:layout_width="32dp"
        android:layout_height="32dp"
        android:layout_marginTop="44dp"
        android:background="@drawable/route2"
        android:visibility="invisible"
        app:layout_constraintEnd_toEndOf="@+id/findmenuBtn"
        app:layout_constraintTop_toTopOf="@+id/findmenuBtn" />

```

```

<Button
    android:id="@+id/searchBtn"
    android:layout_width="32dp"
    android:layout_height="32dp"
    android:layout_marginEnd="12dp"
    android:layout_marginRight="12dp"
    android:layout_marginTop="8dp"
    android:background="@drawable/search"
    android:visibility="invisible"
    app:layout_constraintEnd_toStartOf="@+id/findmenuBtn"
    app:layout_constraintTop_toTopOf="@+id/findmenuBtn" />
</android.support.constraint.ConstraintLayout>

```

## route.xml - содержит view элементы поля для построения маршрута

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/RConLay"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:focusable="true"
    android:focusableInTouchMode="true"
    tools:context="com.example.activity_guk">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="220dp"
        android:background="@color/colorWhite"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="25dp"
        android:layout_marginLeft="16dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="30dp"
        android:text="Корпус"
        android:textColor="@color/Black"
        android:textSize="14sp"
        app:layout_constraintLeft_toLeftOf="@+id/textView"
        app:layout_constraintStart_toStartOf="@+id/textView"
        app:layout_constraintTop_toTopOf="@+id/textView" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="125dp"
        android:layout_height="39dp"
        android:layout_marginLeft="16dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="60dp"
        android:text="Искомый объект или № аудитории"
        android:textColor="@color/Black"
        android:textSize="14sp"
        app:layout_constraintLeft_toLeftOf="@+id/textView"

```

```

        app:layout_constraintStart_toStartOf="@+id/textView"
        app:layout_constraintTop_toTopOf="@+id/textView" />

<Spinner
    android:id="@+id/spinner1"
    android:layout_width="175dp"
    android:layout_height="25dp"
    android:layout_marginLeft="160dp"
    android:layout_marginStart="168dp"
    android:layout_marginTop="30dp"
    app:layout_constraintLeft_toLeftOf="@+id/textView"
    app:layout_constraintStart_toStartOf="@+id/textView"
    app:layout_constraintTop_toTopOf="@+id/textView" />

<EditText
    android:id="@+id/RouteText1"
    android:layout_width="163dp"
    android:layout_height="35dp"
    android:layout_marginStart="168dp"
    android:layout_marginTop="66dp"
    android:backgroundTint="@color/Black"
    android:ems="10"
    android:hint="111"
    android:inputType="text"
    android:textAppearance="@style/TextAppearance.AppCompat.Button"
    android:textCursorDrawable="@null"
    android:textSize="14sp"
    app:layout_constraintStart_toStartOf="@+id/textView"
    app:layout_constraintTop_toTopOf="@+id/textView" />

<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="25dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="5dp"
    android:text="Начальная точка:"
    android:textSize="14sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/textView" />

<Button
    android:id="@+id/CloseTop2"
    android:layout_width="wrap_content"
    android:layout_height="25dp"
    android:layout_marginBottom="4dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="16dp"
    android:background="@drawable/but"
    android:text="Закреть"
    android:textAppearance="@style/TextAppearance.AppCompat"
    app:layout_constraintBottom_toBottomOf="@+id/textView"
    app:layout_constraintEnd_toEndOf="@+id/textView"
    app:layout_constraintRight_toRightOf="@+id/textView" />

<Button
    android:id="@+id/goRoute"
    android:layout_width="wrap_content"
    android:layout_height="25dp"
    android:layout_marginBottom="4dp"
    android:layout_marginStart="8dp"
    android:background="@drawable/but"

```

```

        android:text="Построить"
        android:textAppearance="@style/TextAppearance.AppCompat"
        app:layout_constraintBottom_toBottomOf="@+id/textView"
        app:layout_constraintStart_toStartOf="parent" />

<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="25dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="100dp"
    android:text="Конечная точка:"
    android:textSize="14sp"
    app:layout_constraintStart_toStartOf="@+id/textView"
    app:layout_constraintTop_toTopOf="@+id/textView" />

<TextView
    android:id="@+id/textView7"
    android:layout_width="wrap_content"
    android:layout_height="25dp"
    android:layout_marginLeft="16dp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="120dp"
    android:text="Корпус"
    android:textColor="@color/Black"
    android:textSize="14sp"
    app:layout_constraintLeft_toLeftOf="@+id/textView"
    app:layout_constraintStart_toStartOf="@+id/textView"
    app:layout_constraintTop_toTopOf="@+id/textView" />

<TextView
    android:id="@+id/textView8"
    android:layout_width="125dp"
    android:layout_height="39dp"
    android:layout_marginLeft="16dp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="145dp"
    android:text="Искомый объект или № аудитории"
    android:textColor="@color/Black"
    android:textSize="14sp"
    app:layout_constraintLeft_toLeftOf="@+id/textView"
    app:layout_constraintStart_toStartOf="@+id/textView"
    app:layout_constraintTop_toTopOf="@+id/textView" />

<Spinner
    android:id="@+id/spinner2"
    android:layout_width="175dp"
    android:layout_height="25dp"
    android:layout_marginLeft="160dp"
    android:layout_marginStart="168dp"
    android:layout_marginTop="120dp"
    app:layout_constraintLeft_toLeftOf="@+id/textView"
    app:layout_constraintStart_toStartOf="@+id/textView"
    app:layout_constraintTop_toTopOf="@+id/textView" />

<EditText
    android:id="@+id/RouteText2"
    android:layout_width="163dp"
    android:layout_height="35dp"
    android:layout_marginStart="168dp"
    android:layout_marginTop="150dp"
    android:backgroundTint="@color/Black"

```

```

    android:ems="10"
    android:hint="111"
    android:inputType="text"
    android:textAppearance="@style/TextAppearance.AppCompat.Button"
    android:textCursorDrawable="@null"
    android:textSize="14sp"
    app:layout_constraintStart_toStartOf="@+id/textView"
    app:layout_constraintTop_toTopOf="@+id/textView" />

```

```
</android.support.constraint.ConstraintLayout>
```

## **search.xml - содержит view элементы поля для поиска**

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/SearchLay"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:focusable="true"
    android:focusableInTouchMode="true"
    tools:context="com.example.activity_guk">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:background="@color/colorWhite"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="25dp"
        android:layout_marginLeft="16dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="30dp"
        android:text="Корпус"
        android:textColor="@color/Black"
        android:textSize="14sp"
        app:layout_constraintLeft_toLeftOf="@+id/textView"
        app:layout_constraintStart_toStartOf="@+id/textView"
        app:layout_constraintTop_toTopOf="@+id/textView" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="125dp"
        android:layout_height="39dp"
        android:layout_marginLeft="16dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="60dp"
        android:text="Искомый объект или № аудитории"
        android:textColor="@color/Black"
        android:textSize="14sp"
        app:layout_constraintLeft_toLeftOf="@+id/textView"
        app:layout_constraintStart_toStartOf="@+id/textView"
        app:layout_constraintTop_toTopOf="@+id/textView" />

```

```

<Spinner
    android:id="@+id/spinner"
    android:layout_width="175dp"
    android:layout_height="25dp"
    android:layout_marginLeft="160dp"
    android:layout_marginStart="160dp"
    android:layout_marginTop="30dp"
    app:layout_constraintLeft_toLeftOf="@+id/textView"
    app:layout_constraintStart_toStartOf="@+id/textView"
    app:layout_constraintTop_toTopOf="@+id/textView" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="25dp"
    android:layout_marginStart="8dp"
    android:text="Поиск:"
    android:textSize="14sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/textView" />

<EditText
    android:id="@+id/searchText"
    android:layout_width="163dp"
    android:layout_height="35dp"
    android:layout_marginLeft="160dp"
    android:layout_marginStart="160dp"
    android:layout_marginTop="65dp"
    android:backgroundTint="@color/Black"
    android:ems="10"
    android:hint="111"
    android:inputType="text"
    android:textAppearance="@style/TextAppearance.AppCompat.Button"
    android:textCursorDrawable="@null"
    android:textSize="14sp"
    app:layout_constraintStart_toStartOf="@+id/textView"
    app:layout_constraintTop_toTopOf="@+id/textView" />

<Button
    android:id="@+id/go"
    android:layout_width="wrap_content"
    android:layout_height="25dp"
    android:layout_marginBottom="4dp"
    android:layout_marginStart="8dp"
    android:background="@drawable/but"
    android:text="Найти"
    android:textAppearance="@style/TextAppearance.AppCompat"
    app:layout_constraintBottom_toBottomOf="@+id/textView"
    app:layout_constraintStart_toStartOf="parent" />

<Button
    android:id="@+id/closeTop"
    android:layout_width="88dp"
    android:layout_height="25dp"
    android:layout_marginBottom="4dp"
    android:layout_marginEnd="8dp"
    android:background="@drawable/but"
    android:text="Закрывать"
    android:textAppearance="@style/TextAppearance.AppCompat"
    app:layout_constraintBottom_toBottomOf="@+id/textView"
    app:layout_constraintEnd_toEndOf="parent" />

```

```
</android.support.constraint.ConstraintLayout>
```

## **colors.xml** - содержит цвета используемые в приложении

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="Black">#000000</color>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
    <color name="colorWhite">#FFFFFF</color>
    <color name="colorBuilding">#6699cc</color>
    <color name="colorClickBuilding">#4185c9</color>
</resources>
```

## **strings.xml** - содержит строки используемые в приложении

```
<resources>
    <string name="app_name">БЮУрГУ</string>
    <string name="provider_gps">GPS</string>
    <string name="provider_network">Network</string>
    <string name="location_settings">Location settings</string>
</resources>
```

## **styles.xml** - содержит стили приложения

```
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

</resources>
```

## **Main.java** - содержит обработчики событий окна, функции для работы программы

```
package com.example;

import android.Manifest;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import java.util.Date;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.support.constraint.ConstraintLayout;
```



```

import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import java.io.IOException;

public class Main extends AppCompatActivity implements OnClickListener {
    //компоненты программы
    int btn_id=0; //id нажатой кнопки
    Intent intent;
    String btn=""; //для запоминания нажатой кнопки
    TextView info; //описание
    Button close; //кнопка, скрывающая описание
    Button a3; //3а корпус
    Button b3; //build_b3 корпус
    Button g3; //3g корпус
    Button aaa2; //2 орпус
    Button aa2; //2 корпус
    Button a2; //2 корпус
    Button a6; //6 корпус
    Button usk; //usk
    Button usk2; //usk
    Button a4; //4корпус
    Button guk; //guk
    Button guk2; //guk
    Button guk3; //guk
    Button guk4; //guk
    Button guk5; //guk
    Button guk6; //guk
    Button guk7; //guk
    Button guk8; //guk
    Button goIn; //перейти внутрь корпуса
    Button findBtn; //кнопки поиска
    Button ClfindBtn; //скрыть кнопки поиска
    Button searchBtn; //кнопка поиска точки
    Button routeBtn; //кнопка построения маршрута
    Button clSearch; //закрыть поиск
    Button clRoute; //закрыть построение маршрута
    Button GoFind; //начать поиск
    Button goRoute; //построить маршрут
    View Sview; //view search
    View Rview; //view route
    View Bview; //view bottom_bar
    View Btnview; //view btn_layout
    ConstraintLayout ScL; //ConstraintLayout для другого view
    ConstraintLayout RcL; //ConstraintLayout для другого view
    ConstraintLayout BcL; //ConstraintLayout для другого view
    ConstraintLayout BtncL; //ConstraintLayout для другого view
    ConstraintLayout.LayoutParams lParams; //параметры layout
    ConstraintLayout.LayoutParams posParams; //параметры layout
    ConstraintLayout pos_lay; //параметры layout

```

```

        ConstraintLayout innerCL; //constraint для макета нижнего бара и
открытия поиска
        ConstraintLayout layout; //общий constraint для макетов
        ConstraintLayout ConLayout; //общий constraint для макетов
        Spinner spin_search; //выпадающий список корпусов для поиска
        Spinner spin_route1; //выпадающий список корпусов для поиска
        Spinner spin_route2; //выпадающий список корпусов для поиска
        activity_search search_obj; //экземпляр класса для поиска
        EditText search_text; //строка поиска
        EditText route_text1; //строка поиска
        EditText route_text2; //строка поиска
        DatabaseHelper mDBHelper;
        SQLiteDatabase mDb;
        String query;
        Cursor cursor;
        LocationManager locationManager;
        StringBuilder sbGPS = new StringBuilder();
        StringBuilder sbNet = new StringBuilder();
        String X, Y;
        Double Start_X=55.157248;
        Double Start_Y=61.364745;
        Double End_X=55.162048;
        Double End_Y=61.364161;
        ImageView pos;
        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
            locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);
            intent = getIntent();
            pos=(ImageView) findViewById(R.id.pos);
            pos_lay=(ConstraintLayout) findViewById(R.id.pos_lay) ;
            posParams=(ConstraintLayout.LayoutParams) pos_lay.getLayoutParams();
            //корпуса
            a3=(Button) findViewById(R.id.a3);
            a3.setOnClickListener(BuildBut);
            b3=(Button) findViewById(R.id.b3);
            b3.setOnClickListener(BuildBut);
            g3=(Button) findViewById(R.id.g3);
            g3.setOnClickListener(BuildBut);
            aaa2=(Button) findViewById(R.id.aaa2);
            aaa2.setOnClickListener(BuildBut);
            aa2=(Button) findViewById(R.id.aa2);
            aa2.setOnClickListener(BuildBut);
            a2=(Button) findViewById(R.id.a2);
            a2.setOnClickListener(BuildBut);
            a6=(Button) findViewById(R.id.a6);
            a6.setOnClickListener(BuildBut);
            usk=(Button) findViewById(R.id.usk);
            usk.setOnClickListener(BuildBut);
            usk2 =(Button) findViewById(R.id.usk2);
            usk2.setOnClickListener(BuildBut);
            guk=(Button) findViewById(R.id.guk);
            guk.setOnClickListener(BuildBut);
            guk2=(Button) findViewById(R.id.guk2);
            guk2.setOnClickListener(BuildBut);
            guk3=(Button) findViewById(R.id.guk3);
            guk3.setOnClickListener(BuildBut);
            guk4=(Button) findViewById(R.id.guk4);
            guk4.setOnClickListener(BuildBut);
            guk5=(Button) findViewById(R.id.guk5);
            guk5.setOnClickListener(BuildBut);
        }
    }

```

```

guk6=(Button) findViewById(R.id.guk6);
guk6.setOnClickListener(BuildBut);
guk7=(Button) findViewById(R.id.guk7);
guk7.setOnClickListener(BuildBut);
guk8=(Button) findViewById(R.id.guk8);
guk8.setOnClickListener(BuildBut);
a4=(Button) findViewById(R.id.a4);
a4.setOnClickListener(BuildBut);

ConLayout = (ConstraintLayout) findViewById(R.id.conLayout);
layout = (ConstraintLayout) findViewById(R.id.layout);
innerCL = (ConstraintLayout) findViewById(R.id.CL);

//шаблон нижнего бара
LayoutInflater ltInflater = getLayoutInflater();
Bview = ltInflater.inflate(R.layout.bottom_bar, layout, true);
BcL = (ConstraintLayout) Bview.findViewById(R.id.bottombar_lay);
BcL.setVisibility(View.INVISIBLE);
info=(TextView) Bview.findViewById(R.id.info);
close=(Button) Bview.findViewById(R.id.close);
close.setOnClickListener(clBut);
goIn=(Button) Bview.findViewById(R.id.goIn);
goIn.setOnClickListener(this);

//шаблон верхнего бара для поиска
Sview = ltInflater.inflate(R.layout.search, layout, true);
ScL = (ConstraintLayout) Sview.findViewById(R.id.SearchLay);
clSearch = (Button) Sview.findViewById(R.id.closeTop);
clSearch.setOnClickListener(searchClouse);
lParams = (ConstraintLayout.LayoutParams) innerCL.getLayoutParams();
lParams.topMargin=300;
GoFind = (Button) Sview.findViewById(R.id.go);
GoFind.setOnClickListener(startSearch);
search_text = (EditText) Sview.findViewById(R.id.searchText);

//адаптер для выпадающего списка и объект класса для поиска
search_obj = new activity_search(this);
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, search_obj.data());

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

//выпадающий список корпусов поиска
spin_search = (Spinner) Sview.findViewById(R.id.spinner);
spin_search.setAdapter(adapter);
spin_search.setPrompt("Корпуса");

//шаблон верхнего бара для построения маршрута
Rview = ltInflater.inflate(R.layout.route, layout, true);
RcL = (ConstraintLayout) Rview.findViewById(R.id.RConLay);
clRoute = (Button) Rview.findViewById(R.id.CloseTop2);
clRoute.setOnClickListener(RouteClouse);
goRoute = (Button) Rview.findViewById(R.id.goRoute);
goRoute.setOnClickListener(StartRoute);
route_text1 = (EditText) Rview.findViewById(R.id.RouteText1);
route_text2 = (EditText) Rview.findViewById(R.id.RouteText2);
RcL.setVisibility(View.INVISIBLE);

//выпадающие списки корпусов маршрута
spin_routel = (Spinner) Rview.findViewById(R.id.spinner1);

```

```

spin_route1.setAdapter(adapter);
spin_route1.setPrompt("Копнуть");
spin_route2 = (Spinner) Rview.findViewById(R.id.spinner2);
spin_route2.setAdapter(adapter);
spin_route2.setPrompt("Копнуть");

//шаблон поискового меню
Btnview = ltInflater.inflate(R.layout.menu_btns, ConLayout, true);
Btncl = (ConstraintLayout) Btnview.findViewById(R.id.btn_layout);
findBtn=(Button) Btnview.findViewById(R.id.findmenuBtn);
findBtn.setOnClickListener(find);
searchBtn=(Button) Btnview.findViewById(R.id.searchBtn);
searchBtn.setOnClickListener(search);
routeBtn=(Button) Btnview.findViewById(R.id.routeBtn);
routeBtn.setOnClickListener(route);
ClfindBtn=(Button) Btnview.findViewById(R.id.ClmenuBtn);
ClfindBtn.setOnClickListener(clfind);

//подключаемся к базе
mDBHelper = new DatabaseHelper(this);
try {
    mDBHelper.updateDataBase();
} catch (IOException mIOException) {
    throw new Error("UnableToUpdateDatabase");
}
try {
    mDb = mDBHelper.getWritableDatabase();
} catch (SQLException mSQLException) {
    throw mSQLException;
}

Boolean goS = intent.getBooleanExtra("goS", false);
if (goS) {
    int build = intent.getIntExtra("sel", 1);
    spin_search.setSelection(build);
    GoFind.callOnClick();
}

@Override
protected void onResume() {
    super.onResume();
    if (Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
        &&
        Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
    {
        return;
    }
    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1000
* 10, 10, locationListener);
    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,1000 *
10, 10, locationListener);
    locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER,
1000 * 10, 10, locationListener);
    checkEnabled();
}
private LocationListener locationListener = new LocationListener() {

@Override
public void onLocationChanged(Location location) {
    formatLocation(location);
}
}

```

```

    }

    @Override
    public void onProviderDisabled(String provider) {
        checkEnabled();
    }

    @Override
    public void onProviderEnabled(String provider) {
        checkEnabled();
        if (ActivityCompat.checkSelfPermission(getBaseContext(),
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
        && ActivityCompat.checkSelfPermission(getBaseContext(),
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
        {
            return;
        }
        formatLocation(locationManager.getLastKnownLocation(provider));
    }

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {
    }
};

private void formatLocation(Location location) {
    if (location == null)
        X=Y="";
    X=String.format("%1$.8f", location.getLatitude());
    Y=String.format("%2$.8f", location.getLongitude());
    showPos();
}

private void checkEnabled() {
    if (locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER) ||
locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER))
        pos.setVisibility(View.VISIBLE);
    else
        pos.setVisibility(View.INVISIBLE);
}

private void showPos() {
    Double temp_x=0.0, temp_y=0.0;
    temp_x=Math.abs(Double.parseDouble(X)-Start_X);
    temp_y=Math.abs(Double.parseDouble(Y)-End_Y);
    temp_x/=0.00000889;
    temp_y/=0.00000108;
    posParams.leftMargin=temp_x.intValue();
    posParams.topMargin=temp_y.intValue();
    pos_lay.setLayoutParams(posParams);
}

//перейти внутрь корпуса
@Override
public void onClick(View v) {
    switch(btn) {
        case "3a": intent = new Intent(this, activity_a3.class);
        break;
        case "3б": intent = new Intent(this, activity_3b.class);
        break;
        case "2": intent = new Intent(this, activity_a2.class);
        break;
        case "3г": intent = new Intent(this, activity_g3.class);
        break;
    }
}

```

```

        case "6":    intent = new Intent(this, activity_a6.class);
                    break;
        case "УСК": intent = new Intent(this, activity_usk.class);
                    break;
        case "1":    intent = new Intent(this, activity_guk.class);
                    break;
        case "4":    intent = new Intent(this, activity_a4.class);
                    break;
    }
    intent.putExtra("goS", false);
    startActivity(intent);
}

//нажатие по корпусу
OnClickListener BuildBut = new OnClickListener() {
    @Override
    public void onClick(View v) {
        if(btn_id!=0)
            findViewById(btn_id).setBackgroundResource(R.color.colorBuilding);
        btn_id=v.getId();
        lParams.bottomMargin=300;
        innerCL.setLayoutParams(lParams);
        BcL.setVisibility(View.VISIBLE);
        btn=((TextView)v).getText().toString();
        String [] args = {btn};
        query="SELECT Description FROM buildings WHERE ?=Building";
        cursor = mDb.rawQuery(query, args);
        if (cursor.moveToFirst()) {
            info.setText(cursor.getString(0));
        }
        cursor.close();
        v.setBackgroundResource(R.color.colorClickBuilding);
    }
};

//скрыть нижнее меню
OnClickListener clBut = new OnClickListener() {
    @Override
    public void onClick(View v) {
        Button b1=(Button) findViewById(btn_id);
        BcL.setVisibility(View.INVISIBLE);
        b1.setBackgroundResource(R.color.colorBuilding);
        lParams.bottomMargin=0;
        innerCL.setLayoutParams(lParams);
    }
};

//показать кнопки поиска
OnClickListener find = new OnClickListener() {
    @Override
    public void onClick(View v) {
        searchBtn.setVisibility(View.VISIBLE);
        routeBtn.setVisibility(View.VISIBLE);
        findBtn.setVisibility(View.INVISIBLE);
        ClfindBtn.setVisibility(View.VISIBLE);
    }
};

//кнопка поиска
OnClickListener search = new OnClickListener() {
    @Override
    public void onClick(View v) {
        if(RcL.getVisibility()==View.VISIBLE){
            RcL.setVisibility(View.INVISIBLE);
        }
    }
};

```

```

        ScL.setVisibility(View.VISIBLE);
        lParams.topMargin=300;
        innerCL.setLayoutParams(lParams);
    }
};
//скрыть поиск
OnClickListener searchClouse = new OnClickListener() {
    @Override
    public void onClick(View v) {
        ScL.setVisibility(View.INVISIBLE);
        lParams.topMargin=0;
        innerCL.setLayoutParams(lParams);
        search_obj.hide();
    }
};
//кнопка построения маршрута
OnClickListener route = new OnClickListener() {
    @Override
    public void onClick(View v) {
        if(ScL.getVisibility()==View.VISIBLE){
            ScL.setVisibility(View.INVISIBLE);
        }
        RcL.setVisibility(View.VISIBLE);
        lParams.topMargin=500;
        innerCL.setLayoutParams(lParams);
    }
};
//скрыть построение маршрута
OnClickListener RouteClouse = new OnClickListener() {
    @Override
    public void onClick(View v) {
        RcL.setVisibility(View.INVISIBLE);
        lParams.topMargin=0;
        innerCL.setLayoutParams(lParams);
    }
};
//скрыть кнопки поиска
OnClickListener clfind = new OnClickListener() {
    @Override
    public void onClick(View v) {
        searchBtn.setVisibility(View.INVISIBLE);
        routeBtn.setVisibility(View.INVISIBLE);
        findBtn.setVisibility(View.VISIBLE);
        ClfindBtn.setVisibility(View.INVISIBLE);
    }
};
//осуществление поиска
OnClickListener startSearch = new OnClickListener() {
    @Override
    public void onClick(View v) {
        if (search_obj.isexist(spin_search.getSelectedItem().toString(),
search_text.getText().toString())) {
            if(search_obj.show().equals("android.intent.action.MAIN")) {
                if(btn_id!=0)
                    findViewById(btn_id).setBackgroundResource(R.color.colorBuilding);
                btn_id=getResources().getIdentifier(search_obj.retID(), "id",
"com.example");
                findViewById(btn_id).callOnClick();
            }
            else {
                intent = new Intent(search_obj.show());
            }
        }
    }
};

```

```

intent.putExtra("clasForSearch", search_text.getText().toString());
                intent.putExtra("goS", true);
                startActivity(intent);
            }
        } else {
            Toast.makeText(getBaseContext(), "Нет совпадений",
Toast.LENGTH_LONG).show();
        }
    }
};
//построение маршрута
OnClickListener StartRoute = new OnClickListener() {
    @Override
    public void onClick(View v) {
        if (search_obj.isexist(spin_route1.getSelectedItem().toString(),
route_text1.getText().toString())) {
            if (search_obj.isexist(spin_route2.getSelectedItem().toString(),
route_text2.getText().toString())) {

                }
            else
                Toast.makeText(getBaseContext(), "Нет совпадений с конечной
точкой", Toast.LENGTH_LONG).show();
        }
        else
            Toast.makeText(getBaseContext(), "Нет совпадений с начальной
точкой", Toast.LENGTH_LONG).show();
    }
};
}
}

```

### **Activity\_3b.java** - содержит обработчики событий окна, функции для работы программы

```

package com.example;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.support.constraint.ConstraintLayout;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import java.io.IOException;

```



```

public class activity_3b extends AppCompatActivity{
    int btn_id=0; //id нажатой кнопки
    String btn=""; //для запоминания нажатой кнопки
    Intent intent;
    Button findBtn; //кнопки поиска
    Button ClfindBtn; //скрыть кнопки поиска
    Button searchBtn; //кнопка поиска точки
    Button routeBtn; //кнопка построения маршрута
    Button clSearch; //заккрыть поиск
    Button clRoute; //заккрыть построение маршрута
    Button GoFind; //начать поиск
    Button Up; //вверх
    View Sview; //view search
    View Rview; //view route
    View Bview; //view bottom_bar
    View Btnview; //view btn_layout
    ConstraintLayout ScL; //ConstraintLayout для другого view
    ConstraintLayout RcL; //ConstraintLayout для другого view
    ConstraintLayout BcL; //ConstraintLayout для другого view
    ConstraintLayout BtncL; //ConstraintLayout для другого view
    ConstraintLayout.LayoutParams lParams; //параметры layout
    ConstraintLayout innerCL; //constraint для макета нижнего бара и
открытия поиска
    ConstraintLayout cLayout; //общий constraint для макетов
    Spinner spin_search; //выпадающий список корпусов для поиска
    Spinner spin_routel; //выпадающий список корпусов для поиска
    Spinner spin_route2; //выпадающий список корпусов для поиска
    activity_search search_obj; //экземпляр класса для поиска
    EditText search_text; //строка поиска
    TextView info; //описание
    Button close; //кнопка, скрывающая описание
    Button cop1; Button b118;
    Button b109; Button tb12;
    Button b111; Button tb13;
    Button b113; Button b120;
    Button b115; Button b122;
    Button b117; Button b124;
    Button b119; Button b126;
    Button b121; Button b128;
    Button b123; Button b130;
    Button b125; Button b130a;
    Button b127; Button b132;
    Button b129; Button b120ab;
    Button b131;
    DatabaseHelper mDBHelper;
    SQLiteDatabase mDb;
    String query;
    Cursor cursor;
    ConstraintLayout.LayoutParams posParams; //параметры layout
    ConstraintLayout pos_lay; //параметры layout
    LocationManager locationManager;
    StringBuilder sbGPS = new StringBuilder();
    StringBuilder sbNet = new StringBuilder();
    String X, Y;
    Double Start_X=55.158079;
    Double Start_Y=61.365092;
    Double End_X=55.158086;
    Double End_Y=61.364893;
    ImageView pos;
    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_3b);
intent = getIntent();
pos= (ImageView) findViewById(R.id.pos_b3);
pos_lay=(ConstraintLayout) findViewById(R.id.pos_lay_b3) ;
posParams= (ConstraintLayout.LayoutParams) pos_lay.getLayoutParams();

Up = (Button) findViewById(R.id.go_up);
Up.setOnClickListener(ClUp);
cop1=(Button) findViewById(R.id.cop1); b118 = (Button)
findViewById(R.id.b118);
cop1.setOnClickListener(ClassBut2);
b118.setOnClickListener(ClassBut2);
b109=(Button) findViewById(R.id.b109); tb12 = (Button)
findViewById(R.id.tb12);
b109.setOnClickListener(ClassBut2);
tb12.setOnClickListener(ClassBut2);
b111=(Button) findViewById(R.id.b111); tb13=(Button)
findViewById(R.id.tb13);
b111.setOnClickListener(ClassBut2);
tb13.setOnClickListener(ClassBut2);
b113=(Button) findViewById(R.id.b113); b120=(Button)
findViewById(R.id.b120);
b113.setOnClickListener(ClassBut2);
b120.setOnClickListener(ClassBut2);
b115=(Button) findViewById(R.id.b115); b122 = (Button)
findViewById(R.id.b122);
b115.setOnClickListener(ClassBut2);
b122.setOnClickListener(ClassBut2);
b117=(Button) findViewById(R.id.b117); b124 = (Button)
findViewById(R.id.b124);
b117.setOnClickListener(ClassBut2);
b124.setOnClickListener(ClassBut2);
b119=(Button) findViewById(R.id.b119); b126 = (Button)
findViewById(R.id.b126);
b119.setOnClickListener(ClassBut2);
b126.setOnClickListener(ClassBut2);
b121=(Button) findViewById(R.id.b121); b128 = (Button)
findViewById(R.id.b128);
b121.setOnClickListener(ClassBut2);
b128.setOnClickListener(ClassBut2);
b123=(Button) findViewById(R.id.b123); b130 = (Button)
findViewById(R.id.b130);
b123.setOnClickListener(ClassBut2);
b130.setOnClickListener(ClassBut2);
b125=(Button) findViewById(R.id.b125); b130a = (Button)
findViewById(R.id.b130a);
b125.setOnClickListener(ClassBut2);
b130a.setOnClickListener(ClassBut2);
b127=(Button) findViewById(R.id.b127); b132 = (Button)
findViewById(R.id.b132);
b127.setOnClickListener(ClassBut2);
b132.setOnClickListener(ClassBut2);
b129=(Button) findViewById(R.id.b129); b120ab = (Button)
findViewById(R.id.b120ab);
b129.setOnClickListener(ClassBut2);
b120ab.setOnClickListener(ClassBut2);

```

```

cLayout = (ConstraintLayout) findViewById(R.id.mapCL_1);
innerCL = (ConstraintLayout) findViewById(R.id.lay_1);

```

```
//шаблон нижнего бара
```

```

LayoutInflater ltInflater = getLayoutInflater();
Bview = ltInflater.inflate(R.layout.bottom_bar, cLayout, true);
BcL = (ConstraintLayout) Bview.findViewById(R.id.bottombar_layout);
BcL.setVisibility(View.INVISIBLE);
info=(TextView) Bview.findViewById(R.id.info);
close=(Button) Bview.findViewById(R.id.close);
close.setOnClickListener(clBut);

//шаблон верхнего бара для поиска
Sview = ltInflater.inflate(R.layout.search, cLayout, true);
ScL = (ConstraintLayout) Sview.findViewById(R.id.SearchLayout);
clSearch = (Button) Sview.findViewById(R.id.closeTop);
clSearch.setOnClickListener(searchClouse);
GoFind = (Button) Sview.findViewById(R.id.go);
GoFind.setOnClickListener(startSearch);
search_text = (EditText) Sview.findViewById(R.id.searchText);
lParams = (ConstraintLayout.LayoutParams) innerCL.getLayoutParams();
lParams.topMargin=400;

//адаптер для выпадающего списка и объект класса для поиска
search_obj = new activity_search(this);
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, search_obj.data());

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

//выпадающий список корпусов поиска
spin_search = (Spinner) Sview.findViewById(R.id.spinner);
spin_search.setAdapter(adapter);
spin_search.setPrompt("Корпуса");
spin_search.setSelection(3);

//шаблон верхнего бара для построения маршрута
Rview = ltInflater.inflate(R.layout.route, cLayout, true);
RcL = (ConstraintLayout) Rview.findViewById(R.id.RConLay);
clRoute = (Button) Rview.findViewById(R.id.CloseTop2);
clRoute.setOnClickListener(RouteClouse);
RcL.setVisibility(View.INVISIBLE);

//выпадающие списки корпусов маршрута
spin_route1 = (Spinner) Rview.findViewById(R.id.spinner1);
spin_route1.setAdapter(adapter);
spin_route1.setPrompt("Корпуса");
spin_route2 = (Spinner) Rview.findViewById(R.id.spinner2);
spin_route2.setAdapter(adapter);
spin_route2.setPrompt("Корпуса");
spin_route2.setSelection(3);

//шаблон поискового меню
Btnview = ltInflater.inflate(R.layout.menu_btns, cLayout, true);
BtncL = (ConstraintLayout) Btnview.findViewById(R.id.btn_layout);
findBtn=(Button) Btnview.findViewById(R.id.findmenuBtn);
findBtn.setOnClickListener(find);
searchBtn=(Button) Btnview.findViewById(R.id.searchBtn);
searchBtn.setOnClickListener(search);
routeBtn=(Button) Btnview.findViewById(R.id.routeBtn);
routeBtn.setOnClickListener(route);
ClfindBtn=(Button) Btnview.findViewById(R.id.ClmenuBtn);
ClfindBtn.setOnClickListener(clfind);

//подключаемся к базе
mDBHelper = new DatabaseHelper(this);

```

```

try {
    mDBHelper.updateDataBase();
} catch (IOException mIOException) {
    throw new Error("UnableToUpdateDatabase");
}
try {
    mDb = mDBHelper.getWritableDatabase();
} catch (SQLException mSQLException) {
    throw mSQLException;
}

Boolean goS = intent.getBooleanExtra("goS", false);
if(goS) {
    String clas = intent.getStringExtra("clasForSearch");
    search_text.setText(clas);
    GoFind.callOnClick();
}
}
@Override
protected void onResume() {
    super.onResume();
    if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
        &&
        ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
    {
        return;
    }
    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1000
* 10, 10, locationManager);
    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1000 *
10, 10, locationManager);
    locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER,
1000 * 10, 10, locationManager);
    checkEnabled();
}
private LocationListener locationManager = new LocationListener() {

    @Override
    public void onLocationChanged(Location location) {
        formatLocation(location);
    }

    @Override
    public void onProviderDisabled(String provider) {
        checkEnabled();
    }

    @Override
    public void onProviderEnabled(String provider) {
        checkEnabled();
        if (ActivityCompat.checkSelfPermission(getBaseContext(),
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
            &&
            ActivityCompat.checkSelfPermission(getBaseContext(),
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
        {
            return;
        }
        formatLocation(locationManager.getLastKnownLocation(provider));
    }

    @Override

```

```

        public void onStatusChanged(String provider, int status, Bundle extras) {
        }
    };

    private void formatLocation(Location location) {
        if (location == null)
            X=Y="";
        X=String.format("%1$.8f", location.getLatitude());
        Y=String.format("%2$.8f", location.getLongitude());
        showPos();
    }
    private void checkEnabled() {
        if (locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER) ||
locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER))
            pos.setVisibility(View.VISIBLE);
        else
            pos.setVisibility(View.INVISIBLE);
    }
    private void showPos() {
        Double temp_x=0.0, temp_y=0.0;
        temp_x=Math.abs(Double.parseDouble(X)-Start_X);
        temp_y=Math.abs(Double.parseDouble(Y)-End_Y);
        temp_x/=0.00000889;
        temp_y/=0.00000108;
        posParams.leftMargin=temp_x.intValue();
        posParams.topMargin=temp_y.intValue();
        pos_lay.setLayoutParams(posParams);
    }
    View.OnClickListener ClassBut2 = new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(btn_id!=0)
                findViewById(btn_id).setBackgroundResource(R.color.colorBuilding);
            btn_id=v.getId();
            lParams.bottomMargin=400;
            innerCL.setLayoutParams(lParams);
            BcL.setVisibility(View.VISIBLE);
            btn=((TextView)v).getText().toString();
            String [] args = {"36", btn};
            query="SELECT c.Description FROM buildings b, classes c WHERE
?=b.Building AND c.Building_ID=b.ID AND c.Class=?";
            cursor = mDb.rawQuery(query, args);
            if (cursor.moveToFirst()) {
                info.setText(cursor.getString(0));
            }
            cursor.close();
            v.setBackgroundResource(R.color.colorClickBuilding);
        }
    };
    //скрыть нижнее меню
    View.OnClickListener clBut = new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            BcL.setVisibility(View.INVISIBLE);
            lParams.bottomMargin=0;
            innerCL.setLayoutParams(lParams);
            findViewById(btn_id).setBackgroundResource(R.color.colorBuilding);
        }
    };
    //показать кнопки поиска
    View.OnClickListener find = new View.OnClickListener() {
        @Override

```

```

        public void onClick(View v) {
            searchBtn.setVisibility(View.VISIBLE);
            routeBtn.setVisibility(View.VISIBLE);
            findBtn.setVisibility(View.INVISIBLE);
            ClfindBtn.setVisibility(View.VISIBLE);
        }
    };
    //кнопка поиска
    View.OnClickListener search = new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(RcL.getVisibility()==View.VISIBLE){
                RcL.setVisibility(View.INVISIBLE);
            }
            ScL.setVisibility(View.VISIBLE);
            lParams.topMargin=400;
            innerCL.setLayoutParams(lParams);
        }
    };
    //скрыть поиск
    View.OnClickListener searchClouse = new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            ScL.setVisibility(View.INVISIBLE);
            search_obj.hide();
            lParams.topMargin=100;
            innerCL.setLayoutParams(lParams);
        }
    };
    //кнопка построения маршрута
    View.OnClickListener route = new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(ScL.getVisibility()==View.VISIBLE){
                ScL.setVisibility(View.INVISIBLE);
            }
            RcL.setVisibility(View.VISIBLE);
            lParams.topMargin=600;
            innerCL.setLayoutParams(lParams);
        }
    };
    // скрыть построение маршрута
    View.OnClickListener RouteClouse = new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            RcL.setVisibility(View.INVISIBLE);
            lParams.topMargin=100;
            innerCL.setLayoutParams(lParams);
        }
    };
    // скрыть кнопки поиска
    View.OnClickListener clfind = new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            searchBtn.setVisibility(View.INVISIBLE);
            routeBtn.setVisibility(View.INVISIBLE);
            findBtn.setVisibility(View.VISIBLE);
            ClfindBtn.setVisibility(View.INVISIBLE);
        }
    };
    //осуществление поиска
    View.OnClickListener startSearch = new View.OnClickListener() {

```

```

        @Override
        public void onClick(View v) {

if(search_obj.isexist(spin_search.getSelectedItem().toString(),search_text.getText
().toString())) {
            if(search_text.getText().toString().equals("")) {
                intent = new Intent(getBaseContext(), Main.class);
                intent.putExtra("sel",
spin_search.getSelectedItemPosition());
                intent.putExtra("goS", true);
                startActivity(intent);
            }
            else {

if(search_obj.show().equals("android.intent.action.activity_3b_2")) {
                if(btn_id!=0)

findViewById(btn_id).setBackgroundResource(R.color.colorBuilding);

btn_id=getResources().getIdentifier(search_obj.retID(), "id", "com.example");
                findViewById(btn_id).setOnClickListener();
            }
            else {
                intent = new Intent(search_obj.show());

intent.putExtra("clasForSearch",search_text.getText().toString());
                intent.putExtra("goS", true);
                startActivity(intent);
                finish();
            }
        }
    }
    else {
        Toast.makeText(getBaseContext(), "Нет совпадений",
Toast.LENGTH_LONG).show();
    }

}
};
// вверх
View.OnClickListener ClUp = new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        intent= new Intent("android.intent.action.activity_3b_2");
        intent.putExtra("goS", false);
        startActivity(intent);
        finish();
    }
};
}

```

**Activity\_search.java** - содержит класс, осуществляющий поиск и построение маршрутов в программе

```

package com.example;

import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.database.SQLException;

```

```

import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.support.constraint.ConstraintLayout;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import java.io.IOException;
import java.util.ArrayList;

public class activity_search extends AppCompatActivity {

    private ArrayList<String> data;    //массив для элементов из базы
//переменные для работы с базой
    private DatabaseHelper mDBHelper;
    private SQLiteDatabase mDb;
    private String query;
    private ArrayList<String> layouts;
    private String h_layout;
    private String [] args;
    private Cursor cursor;
    private Cursor cursor2;
    private String point;

    activity_search(Context context) {
        layouts=new ArrayList<>();
        data = new ArrayList<>();
//подключаемся к базе
        mDBHelper = new DatabaseHelper(context);
        try {
            mDBHelper.updateDataBase();
        } catch (IOException mIOException) {
            throw new Error("UnableToUpdateDatabase");
        }
        try {
            mDb = mDBHelper.getWritableDatabase();
        } catch (SQLException mSQLException) {
            throw mSQLException;
        }
        query="SELECT Building FROM buildings";
        cursor = mDb.rawQuery(query, null);
        if (cursor.moveToFirst()) {
            do {
                data.add(cursor.getString(0));
            } while (cursor.moveToNext());
        }
        cursor.close();
    }
    public boolean isexist(String building, String p) {
        hide();
        args = new String [] {building, p};
        String [] arg = {building};
        if (p.equals("")){
            query="SELECT l.Name FROM layouts l, buildings b WHERE ?=b.Building
AND b.Layout_ID=l.ID";
            cursor = mDb.rawQuery(query, arg);
            if (cursor.moveToFirst()) {
                layouts.add(cursor.getString(0));
                cursor.close();
            }
        }
    }
}

```



```

        return true;
    }
    return false;
}
query="SELECT c.Class, c.Layout_ID FROM classes c, buildings b WHERE
?=b.Building AND c.Building_ID=b.ID AND ?=c.Class";
cursor = mDb.rawQuery(query, args);
if (cursor.moveToFirst()) {
    h_layout=cursor.getString(1);
    arg[0]= h_layout;
    query="SELECT l.Name FROM layouts l WHERE ?=l.ID";
    cursor2 = mDb.rawQuery(query, arg);
    cursor2.moveToFirst();
    h_layout=cursor2.getString(0);
    layouts.add(cursor2.getString(0));
    cursor.close();
    cursor2.close();
    return true;
}
else{
    cursor.close();
    query="SELECT c.Class, c.Layout_ID FROM classes c, buildings b WHERE
?=b.Building AND c.Building_ID=b.ID AND ?=c.Description";
    cursor = mDb.rawQuery(query, args);
    if (cursor.moveToFirst()) {
        do {
            h_layout = cursor.getString(1);
            arg[0] = h_layout;
            query = "SELECT l.Name FROM layouts l WHERE ?=l.ID";
            cursor2 = mDb.rawQuery(query, arg);
            cursor2.moveToFirst();
            layouts.add(cursor2.getString(0));
            cursor2.close();
        }while(cursor.moveToNext());
        cursor.close();
        return true;
    }
}
cursor.close();
return false;
}
public String show() {
    return layouts.get(0);
}
public ArrayList<String> data() {
    return data;
}
public String retID(){
    if(args[1].equals("")){
        String arg[]={args[0]};
        query="SELECT B_ID FROM buildings b WHERE ?=b.Building";
        cursor = mDb.rawQuery(query, arg);
        cursor.moveToFirst();
        point=cursor.getString(0);
        cursor.close();
    }
    else{
        String arg[]={args[1],args[0]};
        query="SELECT c.B_ID FROM classes c, buildings b WHERE ?=c.Class AND
?=b.Building AND c.Building_ID=b.ID";
        cursor = mDb.rawQuery(query, arg);
        cursor.moveToFirst();
    }
}

```

```

        point=cursor.getString(0);
        cursor.close();
    }
    return point;
}
public void hide(){
    layouts.clear();
    point="";
    h_layout="";
}
}
}

```

## **DataBaseHelper.java** - содержит класс для работы с базой данных

```

package com.example;
import android.content.Context;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

public class DatabaseHelper extends SQLiteOpenHelper {
    private static String DB_NAME = "susu.db";
    private static String DB_PATH = "";
    private static final int DB_VERSION = 17;

    private SQLiteDatabase mDataBase;
    private final Context mContext;
    private boolean mNeedUpdate = false;

    public DatabaseHelper(Context context) {
        super(context, DB_NAME, null, DB_VERSION);
        if (android.os.Build.VERSION.SDK_INT >= 17)
            DB_PATH = context.getApplicationInfo().dataDir + "/databases/";
        else
            DB_PATH = "/data/data/" + context.getPackageName() + "/databases/";
        this.mContext = context;

        copyDataBase();

        this.getReadableDatabase();
    }

    public void updateDataBase() throws IOException {
        if (mNeedUpdate) {
            File dbFile = new File(DB_PATH + DB_NAME);
            if (dbFile.exists())
                dbFile.delete();

            copyDataBase();

            mNeedUpdate = false;
        }
    }

    private boolean checkDataBase() {

```

```

        File dbFile = new File(DB_PATH + DB_NAME);
        return dbFile.exists();
    }

    private void copyDataBase() {
        if (!checkDataBase()) {
            this.getReadableDatabase();
            this.close();
            try {
                copyDBFile();
            } catch (IOException mIOException) {
                throw new Error("ErrorCopyingDataBase");
            }
        }
    }

    private void copyDBFile() throws IOException {
        InputStream mInput = mContext.getAssets().open(DB_NAME);
        //InputStream mInput
        mContext.getResources().openRawResource(R.raw.info);
        OutputStream mOutput = new FileOutputStream(DB_PATH + DB_NAME);
        byte[] mBuffer = new byte[1024];
        int mLength;
        while ((mLength = mInput.read(mBuffer)) > 0)
            mOutput.write(mBuffer, 0, mLength);
        mOutput.flush();
        mOutput.close();
        mInput.close();
    }

    public boolean openDataBase() throws SQLException {
        mDataBase = SQLiteDatabase.openDatabase(DB_PATH + DB_NAME, null,
        SQLiteDatabase.CREATE_IF_NECESSARY);
        return mDataBase != null;
    }

    @Override
    public synchronized void close() {
        if (mDataBase != null)
            mDataBase.close();
        super.close();
    }

    @Override
    public void onCreate(SQLiteDatabase db) {

    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        if (newVersion > oldVersion)
            mNeedUpdate = true;
    }
}

```

## ПРИЛОЖЕНИЕ 2

### ПРОГРАММА ДЛЯ НАВИГАЦИИ ПО УЧЕБНЫМ КОРПУСАМ ЮУрГУ Описание программы

листов 3

2018

86

## ОГЛАВЛЕНИЕ

П2.1 Общие сведения.....	88
П2.2 Функциональное назначение.....	88
П2.3 Описание логической структуры .....	88
П2.4 Используемые технические средства .....	89
П2.5 Вызов и загрузка .....	89
П2.6 Входные и выходные данные.....	89

## П2.1 Общие сведения

Данная программа, являющаяся мобильным приложением, должна предоставлять возможность навигации пользователя по учебным корпусам ЮУрГУ.

Программа обладает следующими возможностями:

- просмотр информации об объекте на карте;
- поиск введенного места;
- построение маршрута между двумя точками;
- отображение текущего местоположения пользователя;

Платформа, на которой написано приложение – Android Studio. Языки программирования Java, XML.

## П2.2 Функциональное назначение

Приложение предназначено для навигации пользователей по учебным корпусам Южно-Уральского государственного университета. Для запуска и полноценной работы программы необходима система должна работать на устройствах с операционной системой Android.

## П2.3 Описание логической структуры

После старта приложения происходит обработка нажатий пользователя на соответствующие объекты на карте. В зависимости от типа объекта запускается обработчик событий, который выполняет нужное действие.

Для создания базы данных использовалась СУБД SQLite.

В таблице П2.1 представлены основные функции программы.

Таблица П2.1

Название	Входные данные	Выходные данные	Назначение
isExist(String build, String class)	Корпус(build), аудитория(class)	True или false	Проверка на существование в БД введенного объекта
Show()	нет	Строка названия layout объекта	Определение местонахождения объекта
Data()	нет	Массив строк	Данные для заполнения выпадающего списка корпусов
retID()	нет	Строка с id	Получение id

		элемента	искомого объекта
Hide()	нет	нет	Очистка всех полей класса

#### П2.4 Используемые технические средства

Для успешной работы приложения необходимо мобильное устройство с операционной системой Android версии 4.1 и выше, а также наличие приложения Google Service.

#### П2.5 Вызов и загрузка

Запуск данного приложения осуществляется на мобильном устройстве пользователя. Для успешного функционирования системы необходимы:

- оперативная память не менее 20 Мб;
- свободная память не менее 50 Мб.

#### П2.6 Входные и выходные данные

Действия «Поиск» и «Построения маршрута» требуют входные данные, представляющие собой текстовые строки, вводимые пользователем, в специальных полях. Остальные действия программы не получают никаких внешних данных.

Выходные данные делятся на два типа: информация из базы данных и сообщение пользователю.

Информация из базы данных подразумевает всю информацию, извлекаемую из нее программой.

Сообщение пользователю – это системное сообщение «Объект не найден», появляющееся в результате, когда пользователь вводит не существующий объект на карте.

## ПРИЛОЖЕНИЕ 3

ПРОГРАММА ДЛЯ НАВИГАЦИИ ПО УЧЕБНЫМ КОРПУСАМ ЮУрГУ  
Руководство пользователя

листов 3

2018

90



## ОГЛАВЛЕНИЕ

П3.1 Назначение программы.....	91
П3.2 Условия выполнения программы .....	91
П3.3 Выполнение программы .....	91
П3.4 Сообщения оператору .....	92

### ПЗ.1 Назначение программы

Программа предназначена для пользователей, которым необходима помощь в навигации по учебным корпусам ЮУрГУ.

Она предоставляет возможность просмотра карты корпусов, поиска объекта на ней, построение маршрута между точками, а также отображение местоположения пользователя.

Поиск объекта осуществляется после того, как пользователь ввел необходимые данные в специальные поля и нажал кнопку «Поиск». Построение маршрута начинается аналогично, после ввода данных и нажатия кнопки «Построить».

Отображение местоположения пользователя работает без прямого участия пользователя. Если на мобильном устройстве включены GPS и WI-FI программа автоматически запустит данную функцию, получая координаты от них.

### ПЗ.2 Условия выполнения программы

Для работы программы необходимо мобильное устройство на базе ОС Android версии 4.1 и выше, наличие приложения Google Service, свободной памяти не менее 50МБ и оперативной памяти не менее 20МБ.

### ПЗ.3 Выполнение программы

Для запуска приложения необходимо нажать на соответствующую иконку приложения на мобильном устройстве.

После этого пользователь увидит стартовое окно, содержащее карту всего студенческого городка Южно-Уральского государственного университета. Пример на Рисунке ПЗ.1



Рисунок ПЗ.1 – «Стартовое окно»

Пользователь может взаимодействовать с данной картой, выделяя объекты на ней для получения информации о них.

Для осуществления поиска необходимо выбрать корпус из соответствующего списка и, если необходимо, ввести номер аудитории. После чего программа откроет соответствующее окно, на котором находится искомый объект, и отобразит его пользователю.

Для построения маршрута необходимо открыть соответствующее поле из меню приложения и ввести данные в поля ввода. После чего программа начнет отрисовку пути до конечной точки в том окне, в котором находится начальная точка.

Для отображения местоположения пользователю необходимо в настройках включить GPS и WI-FI. Если таковые включены, то система автоматически начнет собирать информацию о координатах пользователя и отображать ее на карте.

#### П3.4 Сообщения оператору

В случаях, когда пользователь вводит данные в соответствующие поля, а система не может найти их соответствие с данными, хранящимися в базе данных, пользователь получит сообщение «Объект не найден» и процедура поиска или отображения маршрута не начнется.