# Назначение разработки

Поиск и подбор недвижимости по заданным параметрам, а также автоматизация процесса построения маршрута просмотра возможных вариантов.

Назначением реализации проекта может служить:

1. автоматизация расчета оптимального маршрута просмотров вариантов;( выполнение расчетов и выдача результатных документов с маршрутом и расписанием;)
2. сбор статистики посещений уникальных страниц объектов(выдача отчета посещений страницы)
3. автоматизация контроля информации
4. скрытие дублирующийся информации
5. выдача отчета посещений страницы
6. просмотр квартир в виде панорам
7. выполнение расчетов и выдача результатных документов с маршрутом и расписанием;
8. выдача справочной информации:
9. по регламентированным запросам;
10. по нерегламентированным запросам.

# Цели и задачи

Цель проекта - создать ресурс, предназначенный для поиска недвижимости по заданным параметрам с учетом дублирующейся информации и возможностью составления маршрута, расписания обзора вариантов, просмотров панорам квартир, а также создание пользовательского и администраторского интерфейсов.

Задачи, решаемые в процессе достижения цели:

1. Анализ поставленной задачи.
2. Исследование предметной области.
3. Сбор и формулирование функциональных требований.
4. Проектирование пользовательского и администраторского интерфейсов
5. Проектирование системы контроля информации
6. Проектирование пользовательского интерфейса.
7. Программная реализация редактора тестов.
8. Программная реализация сервера.
9. Программная реализация клиента.
10. Выбор методов тестирования программного продукта.
11. Проведение тестирования системы.
12. Анализ и исправление выявленных ошибок и недочетов.

# Требования к разработке

## Обеспечивать регистрацию и авторизацию пользователей;

## Поддержка многопользовательского режима работы - администратор и клиент;

## Наличие личного профиля – возможность клиента сохранять карты маршрутов, добавлять варианты для дальнейшего просмотра;

## Создать интерактивную карту, выполняющую следующие функции:

## Возможность просмотра недвижимости на определенную дату, с указанием времени и местом проведения

## Вывод аналогичных вариантов в данном районе;

## Просмотр панорам квартир в реальном режиме

## Органайзер просмотров - простое дополнение для агентов, отображающее на странице каждого варианта текущий список желающих, и позволяющее записаться в очередь.

## Учет дублирующейся информации – виджет, отображающийся рядом с каждым вариантом, и показывающих негативную оценку других пользователей. При этом дополнительной проверке администратора будут подвергаться варианты, находящиеся друг от друга в радиусе 1 километра, и имеющие схожие характеристики.

# Cоставления маршрута

На сегодняшний день чтобы составить маршрут просмотра необходимо звонить по номеру указанному в объявлении, уточнить время показа(если вообще это актуально на данный момент), после чего возникают сложности с состыковкой времени различных вариантов. На это все тратится около 2-3 часов бесполезного времени(часто при «прозвоне» выясняются интересные подробности несоответствия квартиры описанию или вообще ее отсутствию).

При все при этом КПД такой работы 3, максимум 4 просмотра в день. На самом деле за время потраченное на подбор этих вариантов их можно было бы уже посмотреть и принять решение о дальнейших действиях.

# Панорамы квартир

Чтобы сложилось наиболее полное представление о предстоящих просмотрах недостаточно одного текстового описания. Фотографии так же не дают полного представления, тем более многие агенты делают их некачественно. Поэтому часто люди идут смотреть наугад, надеясь на правдивость текстового описания или со слов агента, заинтересованного в направлении потока клиентов на просмотр. Это делается зачастую для того чтобы хозяева оценили их «непосильный труд». Конечно же в итоге все просто впустую тратят свое время.

## # Органайзер просмотров

## Это простое дополнение, отображающее на странице каждого варианта позволит сразу увидеть текущий список желающий, а так же записаться в очередь. Для агентов же это будет большим подспорьем, ведь часто при огромном потоке звонков некоторые номера теряются, а это возможно именно те люди, которым бы понравился вариант.

## # Поиск дублирующейся информации

## Многие агенты выставляют квартиру по соседним адресам, получаются две три одинаковые квартиры, которые при поиске отнимают время на их выявление. Предлагается ввести кнопку рядом с каждым вариантом, нажав на которую можно отметить вариант как ложный

## # Управление агентством

Эта функция обеспечивает возможность простого добавления и удаление агентов в агентство, так же распределение вариантов агентства между агентами, с возможностью потом их перераспределить.

## Добавление агентства и агентов, администраторов

## Просмотр вариантов агентства

## Посмотреть все офисы на карте

## Оставить заявку на сдачу или съем квартиры

## Заказать звонок

Возможность передать вариант другому агенту(или просто расшарить ему доступ на редактирование или на что еще например на публикацию своего номера короче отмечаем галочками).

## Описание входных и выходных данных

Входные и выходные данные представлены в Таблице П.Б.1.

Таблица П.Б.1

Входные и выходные данные

|  |  |  |
| --- | --- | --- |
| **Функция** | **Входные данные** | **Выходные данные** |
| наглядное представление результатов | результаты поиска | табличное и графическое(на карте) представление данных |
| Составление маршрута | Координаты объектов на карте и время | Графическое представление данных в виде марщрута |
| многопользовательский режим работы | идентификатор пользователя | права доступа |
| авторизация | логин и пароль | идентификатор пользователя |
| регистрация | ФИО, почта, номер телефона, агенство | логин и пароль, идентификатор пользователя |
| наличие личного кабинета пользователя | идентификатор пользователя | личная информация, объекты и результаты запросов |

## Модель работы продукта разработки

Для наглядного отображения функционала каждого типа пользователей были разработаны три диаграммы UML Use-Case, которые представлены на рисунках П.А.1 – П.А.3.

Общая структура системы представлена на рисунке П.А.4 в виде UML Component Diagram.

Описание работы системы отображено на диаграммах IDEF0 на рисунках П.А.5 – П.А.10.

## Требования к надежности

Аппаратно-программные компоненты системы должны осуществлять выполнение процедур резервирования и восстановления системы после сбоев. Будет реализовано средствами веб-хостинга.

## Требования к защите информации от несанкционированного доступа

Разрабатываемая система должна обеспечивать разделение прав доступа между пользователями.

## Требования к аппаратным и/или программным средствам разработки

При разработке программного продукта было решено использовать связку php + html/css + javascript для клиентской части, а для разработки сервера был выбран язык С++.

Аппаратных требований к разработке не предъявляется.

## Требования к технологиям разработки

Программный продукт будет реализован с использованием технологии «клиент-сервер». Компонентами системы являются клиент, сервер и редактор. Взаимодействие компонентов происходит через локальную сеть. На сервере будет размещено хранилище, а также на нем будет происходить обработка данных. В клиенте будут отображаться такие данные, как содержание тестов и профили пользователей. Изменение данных (редактирование, добавление и удаление) будет происходить через редактор, к которому будет иметь доступ только администратор.

## Требования к составу и параметрам технических средств, применяемых при использовании системы

Требования к аппаратным и программным средствам: наличие сетевого соединения.

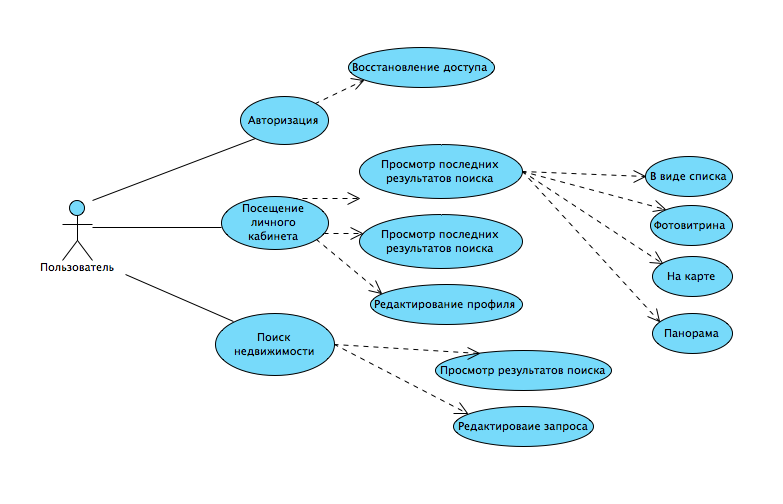
## Методы тестирования соответствия разработки предъявленным требованиям

Выбор и согласование с заказчиком методик тестирования будет проводится в процессе разработки.

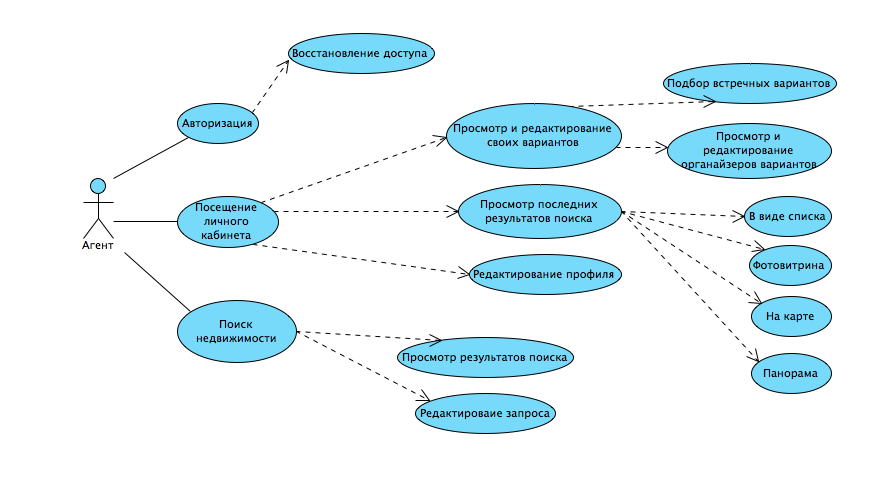
# ПРИЛОЖЕНИЕ А

# Модель разработки

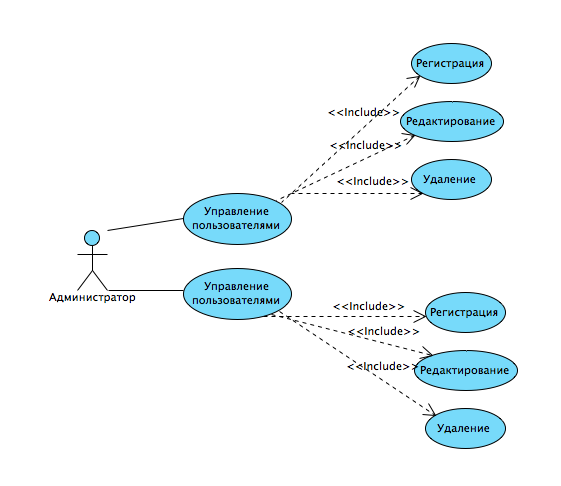
## UML Diagram



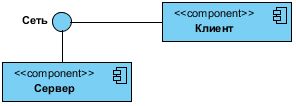
*Рисунок А.1 UML Use-Case Diagram для пользователя*



*Рисунок А.2 UML Use-Case Diagram для агента*

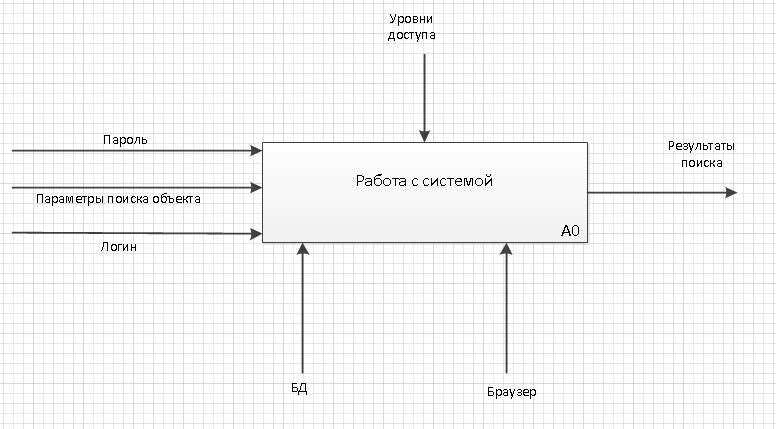


*Рисунок А.3 UML Use-Case Diagram для администратора*

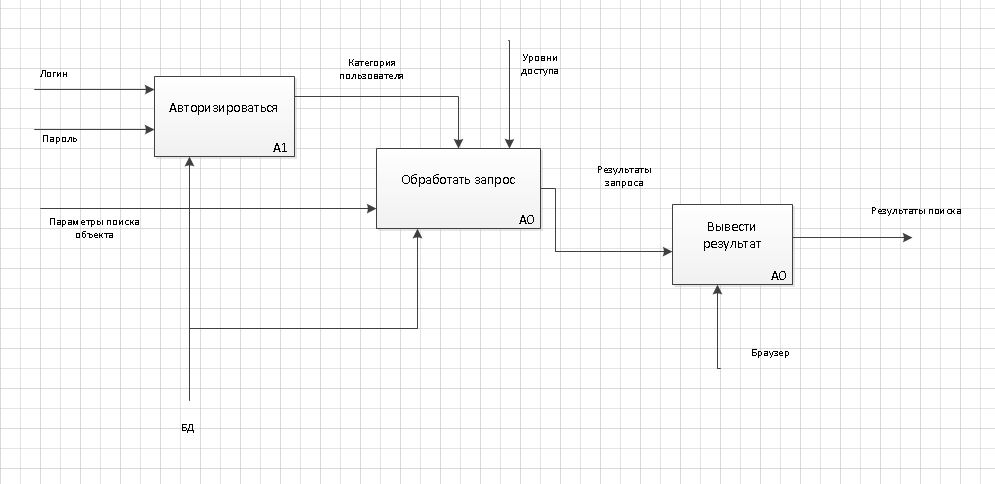


*Рисунок А.4 UML Component Diagram*

**IDEF0**



*Рисунок А.5 IDEF0: уровень А-0*



*Рисунок А.6 IDEF0: уровень А0*