

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ
Школа бакалавриата

ОТЧЕТ

По проекту
«Разработка сайта для создания застройки территории с подгрузкой откры-
тых картографических библиотек/баз данных»
по дисциплине «Проектный практикум»

Заказчик: ООО "ИнПАД"

Куратор: Кузнецов Д.С.

Студенты команды RadikULitki

Воронцов Г.Т.

Колмаков А.Д.

Бурдук К.Е.

Комов А.Е.

Максутов А.Д.

Екатеринбург, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Основная часть	5
1.1 Информация о работе каждого участника в отдельности	5
1.2 Разбор требований заказчика и пользователей к программному продукту и составление плана действий для достижения цели (backlog)	7
1.2.1 Требования заказчика	7
1.2.2 План действий (backlog).....	7
1.3 Анализ и сопоставление аналогов разрабатываемого продукта	9
1.4 Обзор архитектуры программного продукта, описание основных компонентов и связей между ними, обоснование выбора архитектурного решения	9
1.4.1 Архитектура.....	9
1.4.2 Алгоритмы	10
1.5 Описание методологии разработки, информация о процессе разработки, отчет о результатах тестирования на промежуточных этапах, разбор выявленных ошибок.....	11
1.5.1 Методология разработки.....	11
1.5.2 Тестирование проекта.....	11
1.6 Информация о планировании деятельности в ходе разработки и распределении задач между участниками команды разработчиков.....	22
ЗАКЛЮЧЕНИЕ	25
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	27
ПРИЛОЖЕНИЕ А Техническое задание по проекту	28
ПРИЛОЖЕНИЕ В Ссылки на источники, иллюстрации	32

ВВЕДЕНИЕ

Целью проекта является разработка веб-приложения, предназначенного для моделирования и оценки условий инсоляции в условиях плотной городской застройки. Основной функционал системы включает создание инсоляционной горки — инструмента, который позволяет определить максимально допустимую этажность зданий с учетом требований по естественному освещению, а также спроектировать инсоляционные каналы для обеспечения нормативных значений освещенности.

В ходе работы над проектом ставится задача разработать модуль инсоляционной горки, который позволит:

- Определить максимальную этажность при условиях плотной застройки;
- Эффективно разместить инсоляционные каналы для максимального получения полезных площадей;
- Получить информативную схему ограничений для застройщика.

Актуальность проекта обусловлена необходимостью оптимизации проектирования городской среды с учетом нормативных требований по инсоляции. В условиях роста плотности застройки и ограниченности пространства особенно важно правильно учитывать факторы естественного освещения, поскольку от этого зависит комфорт и здоровье будущих жителей, а также соответствие проектных решений градостроительным нормам.

Разрабатываемый программный продукт будет полезен в сфере архитектурного проектирования, урбанистики и градостроительства. Он найдет применение как в эскизном проектировании на ранних этапах, так и в экспертной оценке уже разработанных концепций застройки. Инструмент также может быть востребован государственными и частными заказчиками, планирующими освоение территорий.

Ожидается, что по завершении проекта будет создан инструмент, позволяющий:

- Быстро и корректно определить ограничения по инсоляции;
- Выявить оптимальные параметры размещения зданий;
- Повысить точность и эффективность градостроительного планирования.

Таким образом, результатом проекта станет современное средство поддержки архитектурных решений, направленное на повышение качества городской застройки и рациональное использование территории.

1 Основная часть

1.1 Информация о работе каждого участника в отдельности

Таблица 1 – Работа каждого участника по отдельности

Участник	Задача	Время (часы)
Комов Алексей	- Разбиение json на классы и его чтение; - Поиск более подходящей библиотеки для расчета положения солнца и ее изучение; - Создал метод для расчета положения солнца, его тестирование и проверка на работоспособность; - Начальная реализация разбиения на вертикальные и горизонтальные фасады без учета глубины окна; - Добавление эндпоинтов, докстрингов и ручное тестирование;	1 2 5 4
	- Исправление ошибок расчета инсоляции;	2
	- Исправление ошибок с построением сетки;	5
	- Доработка модуля инсоляционной горки.	3
		6

Продолжение таблицы 1

Участник	Задача	Время (часы)
Колмаков Александр	<ul style="list-style-type: none"> - Реализовал логику создания вертикальной сетки; - Доделал логику создания горизонтальной ветки; - Сделал определение точки расчета инсоляции центра окна; - Инициализация проекта; - Доработка алгоритмов отслеживания позиции солнца из окна; - Создание нового модуля инсоляционной горки. 	3 3 3 1 3 15
Бурдук Константин	<ul style="list-style-type: none"> - Подготовка ПО и данных для тестирования сервиса; - Тестирование сервиса на имеющихся данных; - Анализ результатов тестирования и составление аналитики для команды на их основе; - Углубленный разбор тестов и поиск возможных улучшений для плагина; - Тестирование отдельных модулей и методов сервиса. 	1 1 4 2 4
Воронцов Григорий	<ul style="list-style-type: none"> - Оформил презентацию проекта: структура, слайды, визуальный стиль; - Подготовил графику, схемы, акценты для облегчения восприятия; - Собрал все материалы команды в единый презентационный формат; - Помог в аналитике, предоставив данные и выводы для более глубокого понимания результатов проекта. 	4 2 2 1
Максутов Айтур	<ul style="list-style-type: none"> - Составил и оформил презентационный текст для отчетной презентации; - проанализировал данные и выделил ключевые достижения команды за спринт; 	2 2

	<p>- обеспечил соответствие отчета установленному формату, уделив внимание детали и ясности изложения.</p>	3
--	--	---

1.2 Разбор требований заказчика и пользователей к программному продукту и составление плана действий для достижения цели (backlog)

1.2.1 Требования заказчика

На основании технического задания заказчик ожидает, что программный продукт будет выполнять следующие функции:

- Загрузка и анализ JSON-файла с геометрией зданий (окружающей застройки и проектируемого здания);
- Определение инсоляционной горки на основе геометрии участка и параметров этажности;
- Автоматическая корректировка формы проектируемого здания с учетом требований по инсоляции;
- Разбиение фасадов на ячейки (горизонтальные и вертикальные), определение инсоляции каждой ячейки;
- Учет нормативов по инсоляции при расчете – количество часов солнечного освещения, не затеняя окружающие объекты;

1.2.2 План действий (backlog)

ЭТАП 1. Анализ библиотек и технологий

Выбрать подходящие инструменты для расчета инсоляции и обработки геометрии.

Задачи:

- Изучить библиотеки для геометрического моделирования.
- Изучить библиотеки для расчета солнечной инсоляции.
- Определить способ взаимодействия с картой.
- Решить, будет ли расчет производиться на клиенте или сервере.

ЭТАП 2. Обработка входных данных (JSON)

Построить базовую сетку над территорией и интерпретировать JSON-файл в объекты.

Задачи:

- Загрузить данные по зданиям в формате JSON (контуры, высоты).
- Построить сетку над исследуемой территорией.
- Разбить территорию на ячейки с фиксированным шагом (задаваемым пользователем).

ЭТАП 3. Расчет инсоляции

Определить освещенность каждой ячейки сетки на основе положения Солнца.

Задачи:

- Реализовать расчет траектории солнца на каждый день в заданный период (например, по часам).
- Для каждой ячейки сетки определить попадает ли на неё солнце с учётом окружающих зданий с учетом координаты и ориентации территории, временного диапазона (например, с 9:00 до 18:00), допустимое минимальное время солнечного освещения в часах (норма по России).
- Получить массив данных по инсоляции (в часах/днях).

ЭТАП 4. Проверка новой формы застройки

Оценить влияние новой формы (например, инсоляционной горки) на окружающую застройку.

Задачи:

- Добавить в сцену новую форму здания, изменённого инсоляционной горкой, провести повторный расчет инсоляции, сравнить с предыдущим результатом и в итоге подготовить отчет по результатам.

ЭТАП 5. Финализация и тестирование

Задачи:

- Отладка, создание документации и тестирование проекта.

1.3 Анализ и сопоставление аналогов разрабатываемого продукта

Для анализа конкурентов было взято 4 сервиса для расчета инсоляции

Таблица 2 – Анализ аналогов

Название аналога	Плюсы	Минусы
Project Sunroof	имеет 3д анализ затемнения и интегрирует с гугл мапс	отсутствие российский городов и расчета по российским нормативам
SunCalc	простота и открытый код	отсутствие учета окружающей застройки и возможности автоматического расчета инсоляционной горки
Ladybug Tools	дает точные расчеты с учетом всех нюансов	для работы с ним требуется специальная подготовка и дополнительные платные программы
RusKEO	сервис полностью соответствует российским нормативам	требует значительных финансовых затрат на лицензию и мощного компьютера для работы

1.4 Обзор архитектуры программного продукта, описание основных компонентов и связей между ними, обоснование выбора архитектурного решения

1.4.1 Архитектура

Архитектура представляет собой монолитное серверное приложение на C#, разработанное с помощью фреймворка ASP.NET Web API. Наш продукт на данном этапе включает в себя три дочерних модуля - построение сетки, расчет инсоляции, построение инсоляционной горки. В данный момент все модули работают единым оркестром в рамках одного сервиса, общаясь посредством вызова методов друг друга, это обусловлено скоростью разработки и простотой понимания. В перспективе архитектура будет работать эффективнее, если осуществить переход на микросервисную архитектуру. Но в данном семестре это успеть нам не представляется возможным.

Что касательно задействованных библиотек:

1. AutoMapper используется для упрощенного взаимодействия с моделями. Позволяет унифицировать модели при расчете и выводе информации, путём инкапсуляции расчетных атрибутов классов.
2. CosineKitty.AstronomyEngine используется для получения информации о позициях Солнца по времени и координатам, о восходах и закатах.
3. Newtonsoft.Json используется для более гибкого взаимодействия с получаемым и отправляемым JSON.
4. Swashbuckle.AspNetCore используется для ведения автоматической (Swagger) документации. Упрощает тестирование и демонстрацию работы.

1.4.2 Алгоритмы

Для реализации инсоляции, в частности обработки солнечных лучей использована структура BVH дерева для оптимизации. Внутри нее реализована коллизия на основе ААВВ и алгоритм пересечения - алгоритм для пересечения луча с оси-выравненным ограничивающим параллелепипедом.

Для построения сетки работа с Numerics и внутренними методами C#, векторная геометрия.

1.5 Описание методологии разработки, информация о процессе разработки, отчет о результатах тестирования на промежуточных этапах, разбор выявленных ошибок

1.5.1 Методология разработки

Методология нашей разработки основана на гибком подходе Scrum. Поскольку изначальные знания команды в предметной области были ограниченными, мы выработали собственную адаптацию процесса:

1. Формируется backlog проекта с задачами, разбитыми на недельные спринты. Дополнительно заранее выделяется время на устранение возможных ошибок
2. Каждый спринт длится от одной до двух недель.
3. По завершении каждого спрингта проводится анализ проделанной работы и выявляются недостатки.
4. В специально отведённое время вносятся доработки и исправления.

Основное время уходило не столько на написание кода, сколько на анализ требований и продумывание архитектурных решений.

1.5.2 Тестирование проекта

Пример входного JSON:

```
{  
    "buildings": [  
        {  
            "id": "building1",  
            "sections": [  
                {  
                    "id": "section1",  
                    "coordinates": [[0, 0], [10, 0], [10, 10], [0, 10]],  
                    "name": "Section 1"  
                }  
            ]  
        }  
    ]  
}
```

```

        "floors": [
            {
                "floorStart": 1,
                "floorEnd": 2,
                "type": "commercial",
                "floorHeight": 4
            },
            {
                "floorStart": 3,
                "floorEnd": 10,
                "type": "residential",
                "floorHeight": 3
            }
        ],
        "startHeight": 0
    }
]
}

],
"osm_buildings": [
{
    "id": "osm_building1",
    "coordinates": [[0, 0], [10, 0], [10, 10], [0, 10]],
    "height": 10
}
],
"parameters": {
    "window": {
        "width": 1.5,
        "depth": 0.3
    },
    "room": {
        "width": 4
    }
}

```

}

Было проведено тестирование основных ручек, используемых в проекте:

1) Реализация нахождения позиции солнца (рисунки 1, 2).

При одинаковых параметрах, беря в учет погрешность нескольких секунд, и более точные вычисления библиотеки AstronomyEngine, получился результат, практически точно совпадающий с аналогичным сервисом для расчёта позиции солнца, отсюда следует вывод, что метод работает корректно.

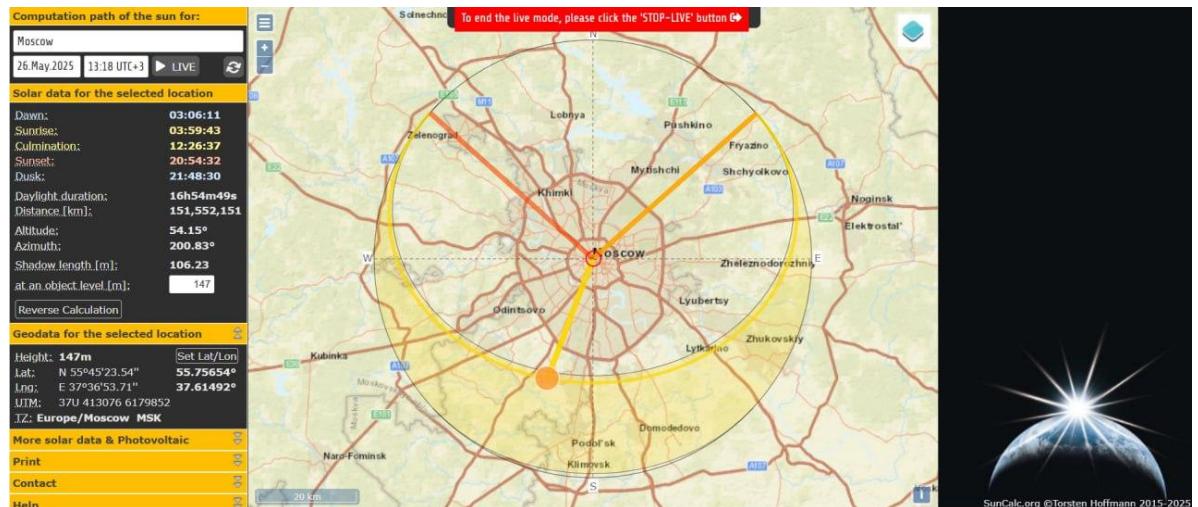


Рисунок 1. Реализация нахождения позиции солнца

The screenshot shows a user interface for testing an API endpoint. At the top, there's a section for 'Parameters' with four fields:

- latitude**: number(\$double) - value: 55.75654 (query)
- longitude**: number(\$double) - value: 37.61492 (query)
- height**: number(\$double) - value: 147 (query)
- dateTime**: string - value: dateTime (query)

Below the parameters are two buttons: 'Execute' (highlighted in blue) and 'Clear'. Under the 'Responses' section, there's a 'Curl' block containing the command to make the API call. The 'Request URL' is listed as <http://localhost:5057/api/SunPos/sun-position?latitude=55.75654&longitude=37.61492&height=147>. The 'Server response' section shows a status code of 200 and a JSON response body:

```
{
  "azimuth": 200.8193684760233,
  "altitude": 54.143556735231215
}
```

At the bottom of the response area, there are 'Copy' and 'Download' buttons.

Рисунок 2. Реализация нахождения позиции солнца

2) Метод нахождения множества позиций солнца на протяжении заданного промежутка времени с определенным интервалом.

Так как метод определения позиции солнца уже проверен, то считаем, что он работает корректно. Дополнительной проверкой к этому методу было то, что ни один промежуток времени не был пропущен. Для этого на короткий промежуток были вручную проверены все нужные данные. Как итог - все данные присутствуют.

При вводе некорректных данных методы обрабатываются конструкцией try-except, которая выдает ошибку при неверных данных, то же самое касается и предыдущего метода.

Помимо этого, данные проверяются инструментами ASP.NET Web API (рисунок 3). Это же будет касаться и всех последующих ручек.

GET /api/SunPos/sun-positions

Cancel

Parameters

Name	Description
latitude number(\$double) (query)	<input type="text" value="lat"/>
longitude number(\$double) (query)	<input type="text" value="37.6173"/>
height number(\$double) (query)	<input type="text" value="147"/>
startDate string (query)	<input type="text" value="startDate"/>
endDate string (query)	<input type="text" value="endDate"/>
intervalMinutes integer(\$int32) (query)	<input type="text" value="1"/>

Please correct the following validation errors and try again.

- For 'latitude': Value must be a number.

Execute **Clear**

Рисунок 3. Проверка данных инструментами ASP.NET Web API

Пример получаемых данных:

```
[
{
  "time": "2025-05-26T00:00:00+05:00",
  "position": {
    "azimuth": 325.8779426312861,
    "altitude": -6.5104618226011866
  }
},
{
  "time": "2025-05-26T00:01:00+05:00",
  "position": {
    "azimuth": 326.0992562182038,
    "altitude": -6.589602929164485
  }
},
{
  "time": "2025-05-26T00:02:00+05:00",
  "position": {
    "azimuth": 326.3206363518815,
    "altitude": -6.628333729164485
  }
}
```

```

    "azimuth": 326.32077000641414,
    "altitude": -6.6682894804214925
}
},
{
    "time": "2025-05-26T00:03:00+05:00",
    "position": {
        "azimuth": 326.5424837649006,
        "altitude": -6.7465198876393515
    }
},
...

```

3) Методы, связанные с сеткой.

Сетка была проверена на разных данных, с разными координатами и разной высотой, при этом все данные высчитываются корректно, что было проверено вручную.

Она была проверена на зданиях разной длины, высоты, а также на возможность работы с отрицательными координатами. Все тесты прошла успешно. Ничего не ломалось, работает стабильно.

Пример ответа сетки:

```

"buildingId": "building1",
    "sections": [
{
    "sectionId": "section1",
    "floors": [
{
        "type": "commercial",
        "floorNumber": 1,
        "height": 4,
        "facadeSides": [
{

```

```

        "normal": "(0, -1, 0)",
        "cells": [
            {
                "coordinates": [
                    "(0, 0, 0)",
                    "(4, 0, 0)",
                    "(4, 0, 4)",
                    "(0, 0, 4)"
                ],
                "window": {
                    "facadeCoordinates": [
                        "(1.25, 0, 1.25)",
                        "(2.75, 0, 1.25)",
                        "(2.75, 0, 2.75)",
                        "(1.25, 0, 2.75)"
                    ],
                    "depthCoordinates": [
                        "(1.25, 0.3, 1.25)",
                        "(2.75, 0.3, 1.25)",
                        "(2.75, 0.3, 2.75)",
                        "(1.25, 0.3, 2.75)"
                    ],
                    "insolationPoint": "(2, 0.15, 2)"
                }
            }
        ]
    }
}

```

```

        } ,
        {
            "coordinates": [
                "(4, 0, 0)",
                "(8, 0, 0)",
                "(8, 0, 4)",
                "(4, 0, 4)"

            ] ,
            "window": {

        "facadeCoordinates": [
            "(5.25,
0, 1.25)",
            "(6.75,
0, 1.25)",
            "(6.75,
0, 2.75)",
            "(5.25,
0, 2.75)"

        ] ,

        "depthCoordinates": [
            "(5.25,
0.3, 1.25)",
            "(6.75,
0.3, 1.25)",
            "(6.75,
0.3, 2.75)",
            "(5.25,
0.3, 2.75)"

        ] ,
        "insolationPoint": "(6, 0.15, 2)"

    ...

```

4) Тестирование инсоляции.

Были проведены следующие тесты (рисунок 4):

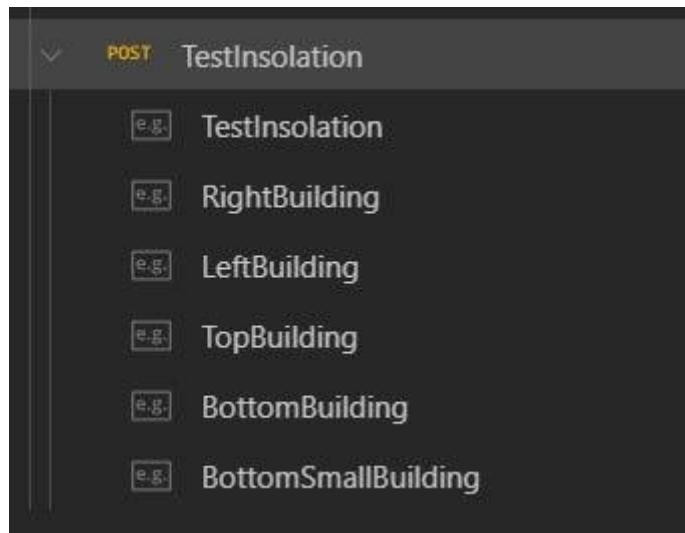


Рисунок 4. Тесты для инсоляционного модуля

Каждый из них проверяет адекватность результатов показания инсоляции при нахождении здания в упор с разных сторон, тесты направлены на проверку, что все стороны правильно обрабатывают затенение от зданий.

Так же был проведен подобный тест на высоту, когда 1 здание ниже другого. Все тесты показали корректные результаты. Здание не соответствует норме только в той стороне, в которой соприкасается с другим зданием - у него все аналогично.

Тест на высоту так же пройден, все обрабатывается корректно.

Пример одного теста:

```
"buildingId": "building1",
    "sections": [
        {
            "sectionId": "section1",
            "floors": [
```

```

{
  "type": "commercial",
  "floorNumber": 1,
  "height": 4,
  "facadeSides": [
    {
      "normal": "(0, -1, 0)",
      "cells": [
        {
          "coordinates": [
            "(0, 0, 0)",
            "(4, 0, 0)",
            "(4, 0, 4)",
            "(0, 0, 4)"
          ],
          "insolationCompliance": "Complies"
        },
        {
          "coordinates": [
            "(4, 0, 0)",
            "(8, 0, 0)",
            "(8, 0, 4)",
            "(4, 0, 4)"
          ],
          "insolationCompliance": "Complies"
        },
        {
          "coordinates": [
            "(8, 0, 0)",
            "(10, 0, 0)",
            "(10, 0, 4)",
            "(8, 0, 4)"
          ],
          "insolationCompliance": "Complies"
        }
      ]
    }
  ]
}

```

```

"insolationCompliance": "Complies"
    }
]
},
{
"normal": "(1, 0, 0)",
"cells": [
{
    "coordinates": [
        "(10, 0, 0)",
        "(10, 4, 0)",
        "(10, 4, 4)",
        "(10, 0, 4)"
    ],
}

"insolationCompliance": "Complies"
},
{
    "coordinates": [
        "(10, 4, 0)",
        "(10, 8, 0)",
        "(10, 8, 4)",
        "(10, 4, 4)"
    ],
}

"insolationCompliance": "Complies"
},
{
    "coordinates": [
        "(10, 8, 0)",
        "(10, 10, 0)",
        "(10, 10, 4)",
        "(10, 8, 4)"
    ],
}
,
```

```
        "insolationCompliance": "Complies"
    }
]
},
{
    "normal": "(0, 1, -0)",
    "cells": [
        {
            "coordinates": [
                "(10, 10, 0)",
                "(6, 10, 0)",
                "(6, 10, 4)",
                "(10, 10, 4)"
            ...
        }
    ]
}
```

1.6 Информация о планировании деятельности в ходе разработки и распределении задач между участниками команды разработчиков

Работа участников проекта была разделена по секторам ответственности, каждый участник отвечал за то, в чем он больше всего разбирался.

Воронцов Григорий - отвечает за все что касается дизайна, структуры проекта, а также является руководителем, следящим за выполнением работы группы, отвечающей за написания текста, отчетов по тестированию, проделанной работе, содержания презентации и итогов. Отвечает за коммуникацию в своей группе, докладывает о проделанной работе тимлиду, вносит правки по его указаниям

Бурдук Константин - отвечает за тесты проекта, проводит тестирование, используя ручки, предоставленные разработчиками, пишет для них отчеты, чтобы они понимали, что необходимо редактировать и исправлять. Входит в состав группы Григория и выполняет работу, данную им. Помогает

формулировать задачи для доски. Также принимает участие в составлении отчетов.

Максутов Айтур - отвечает за аналитику, сравнительный анализ четырех онлайн-сервисов для расчета инсоляции, а также за составление и оформление презентационного текста для отчетной презентации ко второй контрольной точке. Им проанализированы командные результаты спринта, выделены ключевые достижения и отражены в отчетной документации, подготовлен и оформлен отчет по проделанной работе всех участников проекта, с описанием вклада каждого, подготовлен итоговый отчет и текст презентации.

Колмаков Александр - отвечает за разработку проекта, в частности реализация алгоритмов считывания попадания луча в инсоляционную точку на сетке, а также за методы обрезки здания и возвращения новой структуры, скорректированной по данным, полученным при расчёте инсоляции зданий с учетом окружающей застройки

Комов Алексей - отвечает за управление командой проекта и разработку. Занимается проверкой выполнения заданий, порученных команде Григория, а также следит за выполнением задач по разработке. Ведет задачи на доске, проводит код ревью. Ответственен за создание сетки зданий по параметрам, присланных в JSON-файле, учет зоны, проведения расчета, вычисления, связанные с позицией солнца, объединение работы с Александром в один общий рабочий продукт.

Для отслеживания работы командой было использованы следующие инструменты:

- Для задач – доска, представленная на Teamproject (рисунок 5), а также доска, привязанная к проекту на GitHub (рисунок 6);
- Git и GitHub репозиторий - для взаимодействия между разработчиками;
- Figma - для работы дизайнера и проверки этой работы тимлидом
- Google docs - для письменной документации и отчетов

- Для коммуникации - каждые 2 недели проводились личные встречи команды, а также постоянно велось общение в группе команды телеграмм.

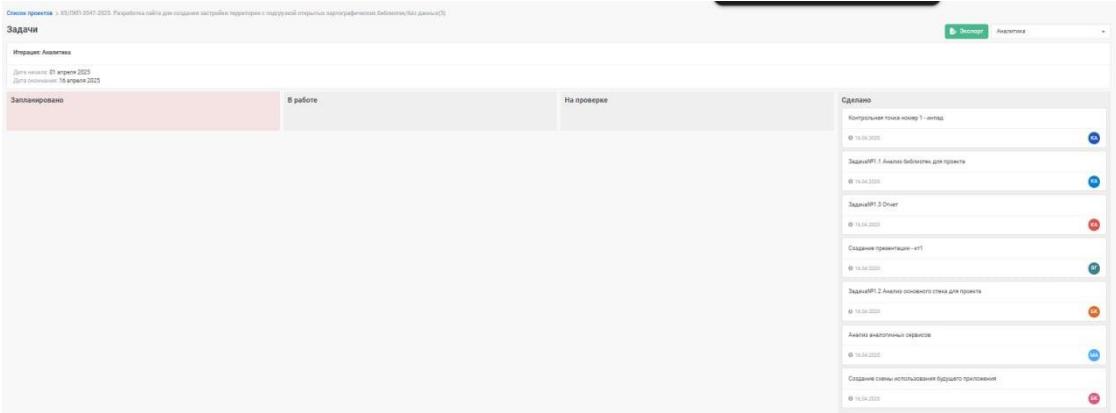


Рисунок 5. Доска задач на Teamproject

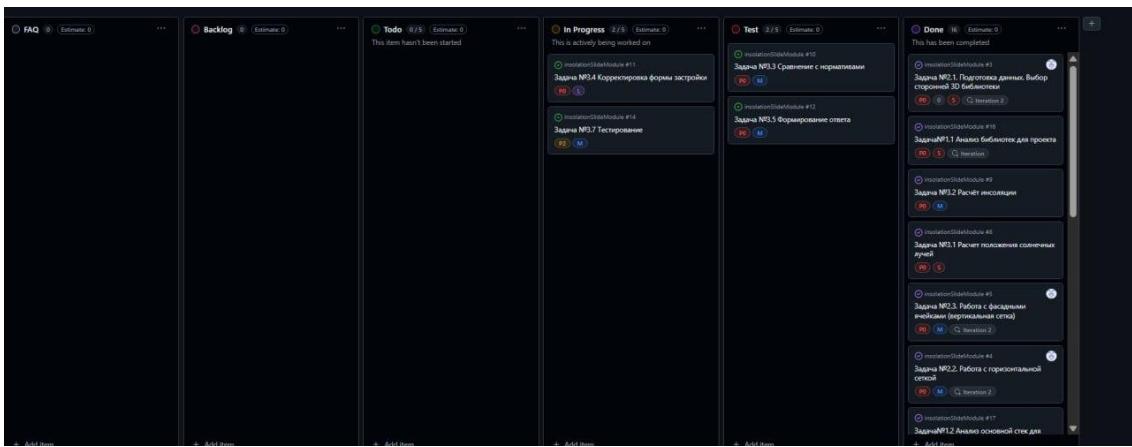


Рисунок 6. Доска задач на GitHub

ЗАКЛЮЧЕНИЕ

Разработанный модуль инсоляционной горки в целом удовлетворяет ключевым требованиям, сформулированным в техническом задании. Он способен анализировать геометрию застройки на основе JSON-данных, рассчитывать инсоляцию с учетом нормативов, и предлагать корректировки формы здания, минимизируя затенение.

Отдельного внимания заслуживает гибкость модуля: он позволяет адаптировать параметры этажности, использовать различные типы фасадных ячеек и визуализировать расчет в удобной форме.

Проведенное тестирование показало, что модуль корректно обрабатывает входные данные: строит правильную модель застройки и выполняет расчет инсоляции в большинстве сценариев. Выявленные ошибки касались в основном частных случаев: неунифицированных форматов JSON, ошибок в геометрии участков и некорректной ориентации зданий в пространстве.

Влияние ошибок на работоспособность оценено как умеренное — сбои происходят лишь при некорректном вводе, что может быть устранено дополнительной валидацией и защитой от ошибок.

На основании анализа проекта и его текущего состояния, предлагаются следующие направления развития:

- Повышение устойчивости к ошибкам во входных данных: внедрение системы валидации JSON и автоматической коррекции типовых структур.
- Поддержка пользовательских сценариев: расчет инсоляции для отдельных помещений, квартир, этажей с выбором точки наблюдения.

Проект показал, что даже в рамках ограниченного объема задач возможно создание функционального аналитического инструмента, способного значительно упростить принятие решений при проектировании плотной городской застройки. Успешная реализация инсоляционной горки

демонстрирует практическую ценность аналитики на основе геопространственных данных и подчеркивает потенциал дальнейшего развития программного продукта в сторону профессиональных градостроительных платформ.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация по .NET [Электронный ресурс] – URL: [Документация по .NET | Microsoft Learn](#) (дата обращения: 27.03.2025).
2. INPAD.STORE [Электронный ресурс] / Инсоляция– URL: [Что такое Инсоляция? Нормы](#) (дата обращения: 20.03.2025).
3. Федеральный закон «О санитарно-эпидемиологическом благополучии населения» № 52-ФЗ от 30.03.99 [Электронный ресурс] / Санитарные правила и нормы СанПиН 2.2.1/2.1.1.1278-03 – URL: [СанПиН 2.2.1/2.1.1.1278-03](#) (дата обращения: 30.03.2025).
4. Правила проектирования естественного и совмещенного освещения [Электронный ресурс] – URL: [СП 367.1325800.2017 Здания жилые и общественные. Правила проектирования естественного и совмещенного освещения](#) (дата обращения: 01.04.2025)

ПРИЛОЖЕНИЕ А

Техническое задание по проекту

Исходный код файла с входными данными для сервиса:

```
{  
    // Застройка пользователя  
    "buildings": [  
        {  
            "id": "building1",  
            "sections": [  
                {  
                    //Можно заменить на GeoJson  
                    "id": "section1",  
                    "coordinates": [[0, 0], [10, 0], [10,  
10], [0, 10]],  
                    "floors": [  
                        {  
                            // Отрезок с 1 по 2 этаж -  
                            коммерческий с высотой 4 метра  
                            "floorStart": 1,  
                            "floorEnd": 2,  
                            "type": "commercial",  
                            "floorHeight": 4  
                        },  
                        {  
                            // Отрезок с 3 по 10 этаж - жилой  
                            с высотой 3 метра  
                            "floorStart": 3,  
                            "floorEnd": 10,  
                            "type": "residential",  
                            "floorHeight": 3  
                        }  
                    ]  
                }  
            ]  
        }  
    ]  
}
```

```

        ] ,
        // Начальная высота секции от земли
        "startHeight": 0
    }
]
}

],
// Застройка с OSM
// С OSM проблемы, см. сообщение в чате
"osm_buildings": [
{
    "id": "osm_building1",
    "coordinates": [[0, 0], [10, 0], [10, 10], [0,
10]],
    "height": 10
}
],
"parameters": {
    "window": {
        "width": 1.5,
        "depth": 0.3
    },
    "room": {
        "width": 4
    }
}
}

```

Копия технического задания

**Разработка модуля инсоляционной горки на сайте с учетом
окружающей застройки**

Вам будет предоставлен JSON файл с геометрией здания, который необходимо будет проанализировать. После проведения анализа вам необходимо определить геометрию всех зданий. В файле будет указана информация о форме участка, которую нужно скорректировать. Инсоляционную горку необходимо сделать на основе геометрии. Процесс создания инсоляционной горки заключается в следующем: Вы получаете определенную геометрию, которая окружает участок. В данном участке вы получаете геометрию, которую необходимо отредактировать на основе инсоляционных лучей. Так же задаем параметры этажности и так далее, и наблюдаем окружающие здания.

С помощью платформы мы видим и понимаем какие окружающие здания. Нам необходимо узнать параметр полученной инсоляции.

Далее полученную информацию необходимо проанализировать, разложить объект на высоту типового этажа, коммерческого этажа. Наложить на форму здания сетку. И после этого выполнить расчет солнца, который попадает внутрь нашего объекта.

Если простыми словами, в JSON файле есть здания окружающей застройки и здания проекта. Вам необходимо показать конечную информацию до и после.

Фасад должен разбиваться на фасадные ячейки. Рассчитываем инсоляцию с учетом зданий, которые влияют на наши объекты. Ячейка должна состоять из типовых параметров.

После появления горизонтальной сетки нужно сделать вертикальную сетку - окнами.

После передачи и получения параметров, нужно сделать расчет. Есть квартира с габаритами, в ней нужно найти точку, куда сделать расчет солнца и куда будет попадать солнце. Если получится сделать на основе MapBox, то делаем на ее основе.

В окружающей застройке нельзя ничего затенять.

Нужно учитывать все нормативы и смотреть в каком месте объект затеняется. Если все хорошо, то оставляем. Если не хватает часов, срезаем определенные части нашей грани. Тем самым получаем нормативы.

Когда выполняем данный процесс, получаем плотность застройки.

Инсоляционные канал могут быть с разных сторон. После определения стороны, накидываем информацию на наше здание, форму и участок и далее срезаем. После срезки получаем не параллелепипед, а усеченную форму. Из этого получаем какую форму имеет участок.

Изучите библиотеки (открытые, закрытые) на каком языке программирования это сделано.

Если расчет будет на C#, создаете отдельную виртуальную машину.

Фасад разбиваете на ячейки и получаете данные по анализу, который вы сделаете.

После этого накидываете библиотеку, где будет произведен расчет солнца, делаете расчет солнца в каждой ячейке.

Когда выполните расчеты, как результат вы получаете форму и геометрию нашего здания, которую необходимо скорректировать по нормативам.

Задача №1. Какие Библиотеки по расчету солнца и инсоляции, считывают локацию и выполняют расчет солнца.

Задача №2. Работа с json геометрией. Необходимо накидывать сетку, накидывать ячейки и делать расчеты в ячейке.

Задача №3. Выполнить расчет инсоляции и понять будет ли новая форма затенять окружающую застройку. Вернуть результат новой геометрии.

ПРИЛОЖЕНИЕ В

Ссылки на источники, иллюстрации

Ссылки

1. Ссылка на наш GitHub с сервисом: [Hallo-Kostya/insolationSlideModule: Разработка модуля инсоляционной горки на сайте с учетом окружающей застройки](#)
2. Ссылка на Google Disk с артефактами: [Артефакты – Google Диск](#)
3. Библиотека Astronomy Engine на GitHub: [cosinekitty/astronomy: Astronomy Engine: multi-language calculation of Sun, Moon, and planet positions. Predicts lunar phases, eclipses, transits, oppositions, conjunctions, equinoxes, solstices, rise/set times, and other events. Provides vector and angular coordinate transforms among equatorial, ecliptic, horizontal, and galactic orientations.](#)

Иллюстрации

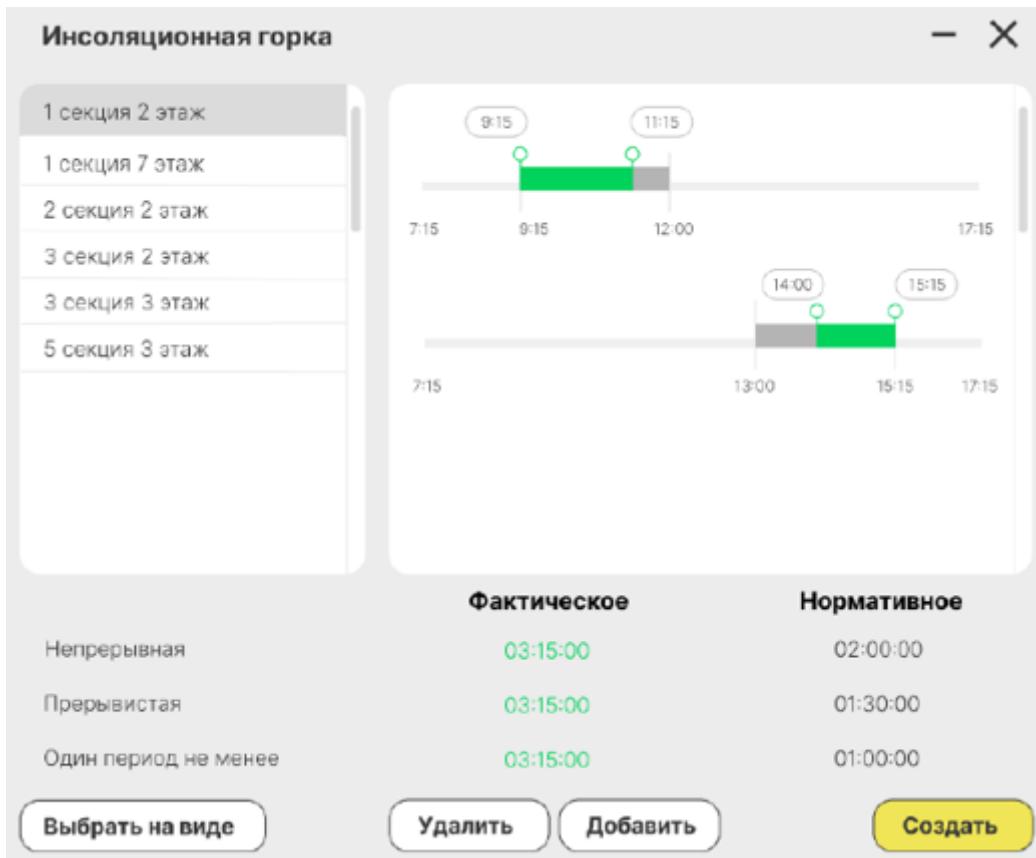


Рисунок 7. Макет интерфейса ввода

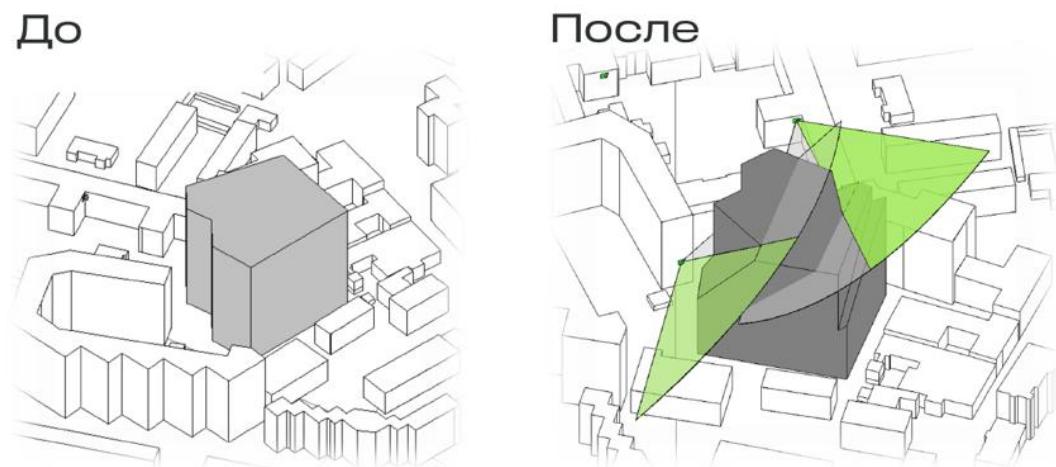


Рисунок 8. Макет работы инсоляционной горки

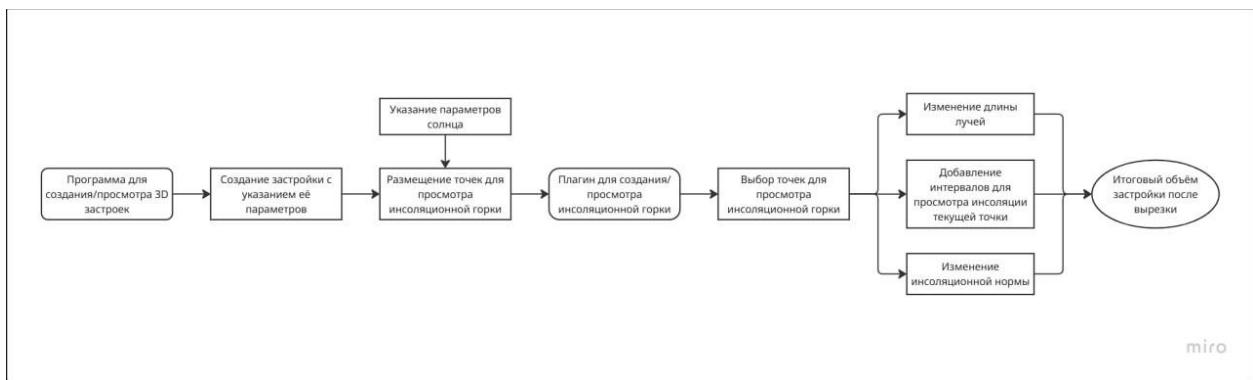


Рисунок 9. Use-case диаграмма работы сервиса