|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ***

***НА ТЕМУ:***

***Приложение для построения профиля рельефа местности***

Студент ИУ9-81Б **\_\_\_\_\_\_\_\_\_\_\_\_** А.А. Митрошкин

(Группа) (Подпись,дата) (И.О.Фамилия)

Руководитель **\_\_\_\_\_\_\_\_\_\_\_\_\_**  А. Б. Домрачева

(Подпись, дата) (И.О.Фамилия)

Консультант **\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2025 г*

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc197324882)

[1. Обзор предметной области 4](#_Toc197324883)

[1.1 Основные характеристики рельефа 4](#_Toc197324884)

[1.2 Профиль рельефа местности 6](#_Toc197324885)

[1.3 Критерии точности и качества профиля рельефа 6](#_Toc197324886)

[1.4 Алгоритмы извлечения профиля по изолиниям 9](#_Toc197324887)

[2. Организация хранилища геоданных 14](#_Toc197324888)

[2.1 Архитектура PostgreSQL и обеспечение целостности 14](#_Toc197324889)

[2.2 Индексирование и управление пространственными данными в PostGIS 15](#_Toc197324890)

[3 Проектирование приложения 19](#_Toc197324891)

[3.1 Функциональные требования 19](#_Toc197324892)

[3.2 Архитектурный дизайн системы 20](#_Toc197324893)

[3.2.1 Сервисно-ориентированная структура 20](#_Toc197324894)

[3.2.2 Взаимодействие клиентского и серверного слоёв 21](#_Toc197324895)

[3.3 Пользовательский интерфейс 22](#_Toc197324896)

[3.4 Модель «Сущность-связь» 23](#_Toc197324897)

[ЗАКЛЮЧЕНИЕ 26](#_Toc197324898)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 27](#_Toc197324899)

[ПРИЛОЖЕНИЕ A 28](#_Toc197324900)

# **ВВЕДЕНИЕ**

Автоматизированное построение вертикального профиля рельефа — одна из базовых задач современной геоинформатики. Профиль высот используется при инженерно-геологических изысканиях, оценке эрозионных процессов, моделировании поверхностного стока, проектировании дорожных и гидротехнических сооружений, а также при территориальном планировании.

Корректность этих расчётов напрямую зависит от точности получаемого разреза, что придаёт особую значимость выбору математических методов обработки исходных геоданных.

В роли исходного материала выступают изолинии — линии равной высоты, формируемые на основе картографических, спутниковых или лидарных измерений. Качественное получение профиля требует решения двух взаимосвязанных задач. Во-первых, необходимо корректно интерполировать высотные значения в местах, где прямых измерений нет. Для этого применяются такие методы, как обратное взвешивание расстояний, кригинг и сплайн-интерполяция; каждый из них по-разному учитывает пространственную корреляцию и распределение ошибок. Во-вторых, требуется построить разрез — наследовать высоты вдоль заданного направления или линии, обеспечив непрерывность и минимизацию погрешности.

# **1. Обзор предметной области**

## 1.1 Основные характеристики рельефа

Рельеф местности – это совокупность неровностей земной поверхности, различающихся по форме, размерам и происхождению. Для систематизации многообразия форм рельефа в геоморфологии используют различные классификации. Одним из подходов является деление форм рельефа по масштабу на следующие категории:

* **Планетарный рельеф (мегарельеф)** – крупнейшие формы, размеры которых сопоставимы с размерами планеты. К ним относят материки, океанические впадины, крупные горные системы и равнины планетарного масштаба;
* **Макрорельеф** – крупные формы рельефа, выделяемые в пределах материков: горные хребты, обширные низменности, плоскогорья, крупные долины. Высотная амплитуда макрорельефа измеряется сотнями и тысячами метров;
* **Мезорельеф** – формы среднего масштаба: отдельные горы, холмы, ущелья, овраги, речные долины, карстовые воронки. Вертикальные размеры мезорельефа обычно десятки – первые сотни метров;
* **Микрорельеф** – мелкие формы рельефа высотой в метры и десятки метров: бугры, ямы, барханы, террасы. Микрорельеф часто является объектом детальных инженерных изысканий;
* **Нанорельеф** – самые мелкие неровности, такие как кочки, борозды вспашки, следы эрозии на почве.

Другой важный подход – генетическая классификация рельефа, учитывающая происхождение форм. Согласно генетической классификации выделяют тектонический рельеф (возвышенности и впадины, образованные движениями земной коры), вулканический рельеф (конусы вулканов, кратеры), водно-эрозионный рельеф (овраги, долины, каньоны, сформированные текущей водой), ледниковый рельеф (карлинги, троги, морены, образованные деятельностью ледников), карстовый рельеф (воронки, пещеры, возникающие вследствие растворения горных пород), эоловый рельеф (барханы, дюны, созданные ветровыми отложениями) и др.

Генетические типы рельефа накладываются на его масштабы: например, крупная горная страна может быть тектонического происхождения, а внутри нее присутствуют мезо и микроформы разного происхождения. Основные морфометрические характеристики рельефа включают: абсолютную высоту точек (относительно уровня моря), относительные превышения, уклоны поверхностей, экспозицию склонов и кривизну поверхности.

Эти характеристики количественно описывают рельеф и используются при построении профиля. Например, крутизна склонов (уклон) напрямую влияет на форму профиля: на крутых склонах профиль показывает резкие перепады высот, на пологих – плавные изменения. В выпуклых областях (положительная кривизна) профиль становится более пологим к вершинам холмов, а в вогнутых (отрицательная кривизна) – показывает впадины и долины.

Для описания рельефа местности широко применяются топографические карты и цифровые модели. На картах рельеф обычно изображается горизонталями (изолиниями равных высот), а также дополнительными способами: отметками высот отдельных точек, шкалой гипсометрической окраски (заливкой высотных зон цветами) и методами художественной картограмметрии (штриховкой, тенью).

В совокупности эти средства дают представление о трехмерной форме поверхности на плоском чертеже карты. Таким образом, рельеф характеризуется многоуровневой системой форм различного масштаба и генезиса. Для изучения рельефа необходимо располагать данными о высоте во всех точках интересующей территории, что достигается посредством создания цифровых моделей рельефа – специальных представлений поверхности в виде, пригодном для компьютерного хранения и обработки.

## 1.2 Профиль рельефа местности

Профиль рельефа местности представляет собой вертикальное сечение поверхности Земли вдоль произвольной линии, проведённой по карте (отрезка или ломаной). С математической точки зрения он может быть описан как график зависимости высоты над уровнем моря от расстояния вдоль этой линии.

Построение такого профиля выполняется геометрически: на карте с горизонталями (изолиниями) отмечают линию разреза, затем определяют точки её пересечения с горизонталями и фиксируют соответствующие высоты местности. Эти значения высот переносятся на вертикальную плоскость, где по горизонтальной оси откладывается расстояние от начала линии, а по вертикальной — высота. В результате получается наглядный продольный разрез рельефа вдоль выбранной линии. Как правило, его иллюстрируют на схеме: в верхней части — карта с горизонталями и выделенной линией (например, отрезок AB), а в нижней — профиль рельефа вдоль этой линии, отражающий форму выпуклостей и впадин рельефа.

## 1.3 Критерии точности и качества профиля рельефа

Профиль рельефа представляет собой разрез местности, поэтому к нему предъявляются требования по точности как по горизонтали (точность положения характерных точек по расстоянию вдоль трассы профиля), так и по вертикали (точность передачи высот). Основные критерии качества, которыми оценивают профили:

* **Вертикальная точность профиля** – отклонение высот точек профиля от истинных значений. Обычно оценивается через среднеквадратическую ошибку (RMSE) [2] по высоте​, если имеются контрольные данные. Вертикальная точность профиля не может быть выше, чем точность исходной цифровой модели;
* **Горизонтальная точность** – если профиль привязан к географическим координатам, важна точность позиционирования, т.е. соответствие расстояний на профиле реальным расстояниям на местности. В большинстве случаев горизонтальные расстояния можно определить достаточно точно по карте или координатам начальной и конечной точки, поэтому основной вклад в горизонтальную ошибку дает неточное определение положения характерных точек рельефа;
* **Детализация (разрешение)** – способность профиля отображать мелкие детали рельефа. Профиль должен повторять все существенные перегибы местности. Если профиль выровнен (усреднен), то некоторая мелкая неровность не появится на разрезе, хотя есть на местности. Разрешение профиля зависит от шага выборки точек;
* **Соблюдение экстремальных точек** – на профиле должны присутствовать все важные высотные экстремумы, которые есть вдоль линии на реальной местности. Это вершины и впадины (пики и минимумы высоты). Критерием является отсутствие пропуска пика или седловины. Если исходная модель недостаточно подробна, может случиться, что небольшой холм между горизонталями не отразится и на профиле;
* **Гладкость и реалистичность формы** – профиль не должен содержать искусственных изломов, если они не обусловлены реальным рельефом. Для оценки этого можно исследовать вторую производную профиля (кривизну).

При оценке точности построенного профиля рекомендуется сравнить его с эталонными данными, если возможно. Эталоном может служить профилирование с помощью высокоточного GNSS по фактической линии местности или профиль, полученный из более точной модели рельефа (интерполяции) без значимых искажений.

1.3 Методы интерполяции профиля

Интерполяция профиля рельефа по набору дискретных точек (s, H) позволяет получить непрерывную зависимость высоты H вдоль линии профиля от расстояния s. Изначально известны лишь точки пересечения линии профиля с изолиниями, задающие пары (s, H). Задача состоит в приведении этих точек плавной кривой или функцией, обеспечивающей адекватное продолжение рельефа между точками. Основная проблема – баланс между точностью прохождения через известные точки и гладкостью полученного профиля. Ниже рассмотрены основные одномерные методы интерполяции и сглаживания данных вдоль линии профиля.

* Кусочно-линейная интерполяция – на каждом интервале высота задаётся линейной формой

Метод -непрерывен, реализуется без решения систем уравнений, но порождает «ломаный» профиль, обусловленный разрывами первой производной;

* Кубические сплайны – для каждого интервала строится кубический полином

коэффициенты которого определяются условиями

Непрерывность , в узлах обеспечивает -гладкость. Метод даёт естественный плавный профиль, но требует решения трёхдиагональной системы;

* Сглаживающие сплайны – ищется функция минимизирующая функционал

где регулирует баланс «точность – гладкость». При получаем классический кубический сплайн; при – сглаженную кривую, не обязательно проходящую строго через все ;

* B-сплайны – кривая представляется в базисе B‑сплайнов порядка p:

где – базисные функции. Гладкость контролируется порядком p (обычно p=3). Локальное влияние контрольных коэффициентов ​ обеспечивает устойчивость и предсказуемость формы, но требует настройки узлового вектора;

* Полиномиальная аппроксимация – Профиль моделируется единственным полиномом степени m:

коэффициенты ​ подбираются методом наименьших квадратов. При малой степени метод прост и даёт гладкую -кривую; при увеличении m возможны осцилляции и снижение устойчивости.

* Скользящее среднее – для каждой точки вычисляется

, ​

где r – радиус окна. Метод устраняет высокочастотный шум, но не гарантирует прохождения через исходные и сглаживает пиковые экстремумы.

## 1.4 Алгоритмы извлечения профиля по изолиниям

Дан набор векторных изолиний (горизонталей) с известными отметками высот и задана профильная линия (прямая отрезок или ломаная) на плане. Необходимо определить высоты вдоль этой линии и построить профиль рельефа – ломаную линию изменения высоты в зависимости от пройденного расстояния. На выходе должен формироваться дискретный профиль (набор точек с координатами расстояние–высота), пригодный для последующей визуализации и анализа.

Алгоритм разбивается на следующие этапы:

1. **Подготовка исходных данных** – на первом этапе подготавливаются данные для расчёта. Все исходные изолинии должны содержать атрибут высоты. Профильная линия задаётся в плановых координатах (как геометрия на плоскости ). Если профильная линия состоит из нескольких сегментов (ломаная), она разбивается на отдельные отрезки для удобства обработки. Также целесообразно упорядочить изолинии по высоте и выбрать только те, которые находятся в границах области, пересекаемой профильной линией, чтобы не обрабатывать лишние данные. После подготовки имеется: множество изолиний с высотами и профильная линия, заданная последовательностью вершин .
2. **Поиск пересечений профильной линии с изолиниями** – для каждого сегмента профильной линии вычисляются точки пересечения с каждой изолинией. Геометрически, каждый сегмент профиля – отрезок на плоскости, а изолиния – либо замкнутая кривая, либо открытая ломаная. Необходимо найти все точки, где профильный отрезок пересекает линию равной высоты. Для этого решается система уравнений для координат точек пересечения отрезков. Каждый факт пересечения дает новую точку профиля: ее плановые координаты находятся на пересечении, а высота этой точки равна отметке пересекаемой изолинии. Формально, пусть профильный сегмент задан параметрически . Изолиния – уравнением . Решая уравнение , находятся параметры и пересечения. Если они лежат в диапазоне , то сегмент профиля пересекает данный отрезок изолинии. Координаты пересечения вычисляются подставновкой найденного в , а высота точки будет , где – известная высота данной изолинии. Таким образом, после проверки всех изолиний собирается набор точек , через которые профильная линия проходит ровно в тех местах, где высота равна одной из высот изолиний. Если профильная линия содержит несколько сегментов, описанная процедура повторяется для каждого сегмента, и все найденные точки объединяются в общий список.
3. **Интерполяция высоты в начальной и конечной точках** – рассматриваются ближайшие изолинии, между которыми расположена данная точка профиля. Если точка начала или конца лежитточно на изолинии, её высота принимается равной высоте этой изолинии (интерполяция не требуется). Если же начальная точка находится между двумя горизонталями с отметками (нижайшая ниже точки) и (ближайшая выше точки), то высоту можно аппроксимировать, исходя из положения точки относительно этих линий. Предполагая, что между соседними горизонталями высота изменяется приблизительно линейно, высоту находят пропорционально расстояниям до каждой изолинии. Например, если расстояние от до линии равно , а до линии – , то:

.

После этого шага начальная и конечная точки профиля получают численные значения высоты, и их можно включить в общий набор профильных точек для построения линии.

1. **Формирование ломаного профиля** – все характерные точки профиля собраны: с высотой , точки пересечения профильной линии с изолиниями, каждая с высотой соответствующей горизонтали, с высотой . Эти точки располагаются в порядке следования вдоль профильной линии от начала к концу. Для упорядочения точек вычисляется проекционная длина пути (горизонтальное расстояние) от начала профиля до каждой точки. Если профильная линия задана в виде координат, длина дуги от начала до точки рассчитывается как сумма длины всех предыдущих сегментов плюс расстояние по текущему сегменту до точки пересечения. В результате определяется параметр – пройденное расстояние от начала профиля до каждой значимой точки. Затем точки сортируются по возрастанию . Соединив упорядоченные точки от до прямыми отрезками в порядке следования, получаем ломаную линию профиля. Формально профиль может быть представлен как функция высоты от пройденного расстояния . Эта функция определяется кусочно-линейно: на каждом интервале высота изменяется по линейному закону. Если на концах сегмента высоты и , то для любой промежуточной точки с параметром расстояния выполняется соотношение линейной интерполяции:

Таким образом, результатом является дискретный профиль в виде последовательности , задающий ломаную линию изменения высоты вдоль заданного направления.

1. **Сглаживание профиля** – полученная ломаная линия — это точная аппроксимация исходных данных, однако из‑за дискретности изолиний она содержит угловатые переломы и не отражает плавность реального рельефа. Для устранения этого эффекта применяется один из методов интерполяции, описанных в разделе 1.3, например сплайн-интерполяция: по упорядоченному набору точек  строится непрерывная сплайн‑функция . Используют кубические сплайны, обеспечивающие непрерывность первой и второй производных. Пусть узлы совпадают с координатами ; тогда на каждом интервале высота описывается полиномом третьей степени:

коэффициенты определяются из условий и равенства первых и вторых производных в соседних узлах. В результате получается сглаженный профиль, сохраняющий все исходные пересечения с изолиниями, но отображающий плавные изменения уклона между ними, что ближе к реальной форме склонов. Такой сплайн‑профиль используется для визуализации и дальнейших вычислений.

# **2. Организация хранилища геоданных**

## 2.1 Архитектура PostgreSQL и обеспечение целостности

Для построения профилей рельефа в приложении требуется эффективно хранить и обрабатывать пространственные данные – такие как наборы высот (точки, растр) и линии (изолинии, профильные линии).

Рассмотрим СУБД PostgreSQL [4] с расширением PostGIS для поддержки географических данных. Рассмотрим особенности архитектуры PostgreSQL и то, как она обеспечивает целостность данных, что критично для надежной работы приложения.

PostgreSQL – это объектно-реляционная система управления базами данных (ОРСУБД) с открытым исходным кодом. С архитектурной точки зрения PostgreSQL использует модель клиент-сервер, где сервер управляет файлами базы данных на диске, а клиенты посылают SQL-запросы. Процессы PostgreSQL обеспечивают параллельное обслуживание многих подключений, с отдельным рабочим процессом на каждое соединение. Ключевым аспектом является обеспечение целостности и надежности данных. PostgreSQL гарантирует т.н. свойства ACID-транзакций: атомарность, согласованность, изоляция, долговечность (англ. Atomicity, Consistency, Isolation, Durability)​. Это означает, что любая серия операций над базой (транзакция) либо выполняется полностью, либо не затрагивает базу вообще (атомарность); в процессе транзакции соблюдаются все ограничения целостности (согласованность); параллельные транзакции не мешают друг другу (изоляция); а после фиксации транзакции изменения надежно сохраняются даже при сбоях (долговечность).

Для геоданных это важно так же, как и для обычных: например, если добавляется новая изолиния и одновременно связанные с ней атрибуты, транзакция гарантирует, что либо и линия, и атрибуты будут добавлены вместе, либо не будет добавлено ничего (исключая несогласованность, при которой линия есть, а атрибутов нет, или наоборот).

Данные записываются с журналированием (Write-Ahead Logging), что позволяет восстанавливаться после сбоев. Также можно настраивать репликацию (дублирование на другие серверы) для отказоустойчивости.

Объектно-реляционная модель PostgreSQL позволяет определять пользовательские типы данных, что и делает PostGIS – добавляет новый тип geometry (и geography) для хранения пространственных объектов (точек, линий, полигонов и т.д.).

Но с точки зрения целостности, эти геометрические данные по-прежнему управляются PostgreSQL как частью транзакций и могут быть связаны с другими таблицами через внешние ключи.

Еще один уровень целостности – геометрическая корректность данных. PostGIS предоставляет функции для проверки валидности геометрий (ST\_IsValid и др.). Например, замкнутый полигон рельефа не должен самопересекаться, линия изолинии должна быть простая (без самопересечений). Такие требования можно контролировать на уровне приложения или с помощью триггеров в БД.

Архитектура хранения: PostgreSQL хранит данные в строках таблиц, разбитых на страницы на диске. Геометрические объекты часто занимают много места, поэтому PostGIS использует собственный бинарный формат (EWKB) для поля geometry, который может включать координаты, тип, SRID (идентификатор системы координат) и т.д.

Поле geometry выглядит для PostgreSQL как большой двоичный объект с известной внутренней структурой. Индексы позволяют ускорять поиск по этим данным.

## 2.2 Индексирование и управление пространственными данными в PostGIS

PostGIS, будучи расширением PostgreSQL, реализует специальные пространственные индексы и оптимизированные функции для управления геоданными.

Основной тип индекса для геометрических колонок в PostGIS – это индекс на основе GiST (Generalized Search Tree) [5] с использованием структуры данных R-дерево​.

R-дерево — это иерархическая структура, где каждый пространственный объект приближенно представляется прямоугольником, выровненным по осям (bounding box – ограничивающий прямоугольник). Эти прямоугольники организуются в дереве: группы близких или пересекающихся bbox объединяются на верхнем уровне в более крупные bbox и так далее. Такой индекс позволяет быстро отсеять объекты, которые гарантированно не подходят под пространственное условие. Например, при запросе «найти все изолинии, пересекающие данную профильную линию» СУБД сначала по индексу извлекает кандидатов – изолинии, чьи ограничивающие прямоугольники пересекаются с прямоугольником вокруг профильной линии. Это очень быстро, т.к. проверяются только координаты прямоугольников. Далее уже точно проверяется пересечение геометрий с помощью вычислений, но только для отфильтрованных кандидатов​.

Без индекса пришлось бы проверять пересечение со всеми линиями в таблице, что для больших наборов (крайне медленно. После этого большинство функций, типа ST\_Intersects, ST\_Within, ST\_Distance и т.д., умеют задействовать индекс на стадии предварительного отбора по bbox (PostGIS автоматически подставляет проверку bbox, называемую «индексный фильтр»). В результате запросы ускоряются на порядки.

Помимо GiST/R-дерева, PostGIS с PostgreSQL 14+ поддерживает SP-GiST для точек (еще более быстрый для очень больших облаков точек), а также BRIN индексы для случаев, когда данные на диске отсортированы пространственно (способ дают быструю грубую фильтрацию с минимальными хранилищными затратами). Но GiST является универсальным и чаще всего употребимым.

Управление пространственными данными в PostGIS включает не только индексы, но и богатый набор функций:

* **ST\_Intersection(geom1, geom2)** – возвращает геометрию пересечения двух объектов. Ее используют, чтобы вычислять точки пересечения профильной линии с изолиниями прямо в SQL. Результатом будут геометрии типа Point (или MultiPoint, если несколько пересечений с одной линией). Далее можно получить координаты этих точек, включая высоту (высоту можно хранить как атрибут изолинии или в Z-координате линии, тогда пересечение наследует Z, если 3D-линия). PostGIS поддерживает 3D и 4D (с временем) геометрии, но большинство функций оперируют только X,Y. Тем не менее, можно хранить изолинии как *LineStringZ*, но ST\_Intersection вернет 2D геометрию, игнорируя Z;
* **ST\_Distance(geom1, geom2)** – определяет расстояние между объектами. Может использоваться для вычисления длины профиля (суммарное расстояние между последовательными точками;
* **ST\_Length(geom)** – длина геометрии (для LineString – длина линии; для горизонтальных координат в проекции в метрах, если СК метры);
* **ST\_SampleLine(geom, fraction)** – функция, позволяющая взять точку на линии на заданной доле ее длины. Например, для профиля можно получать точки через равные интервалы доли;
* **ST\_Union** – объединение геометрий, можно применять для слияния данных, например построения мультиленты из всех изолиний.

Для больших моделей рельефа PostGIS предлагает тип Raster – можно хранить растровый DEM прямо в базе данных. Однако с точки зрения организации данных это не является оптимальным: хранение больших растров в базе данных приводит к большим объемам и нагрузке. Часто вместо этого хранят растр вне, а в базе данных – только ссылки или метаданные. Но PostGIS Raster позволяет делать запросы типа: ST\_Value(raster, geom\_point) – получить значение высоты растра в заданной точке; или ST\_Profile – функция, которая может взять растр и линию и вернуть набор значений – фактически построить профиль.

Отдельно стоит сказать об управлении проекциями: PostGIS имеет таблицу spatial\_ref\_sys со всеми стандартными SRID. Данные рельефа могут храниться в географической координатной системе (широта/долгота).

Таким образом, связка PostgreSQL и PostGIS обеспечивает надёжную и эффективную работу с геоданными рельефа, служа прочным фундаментом для функциональности приложения.

# 3 Проектирование приложения

## 3.1 Функциональные требования

Основываясь на целях проекта и предполагаемых сценариях использования, были сформулированы следующие ключевые функциональные требования к разрабатываемому приложению:

– **Построение профиля «одним действием»** – приложение принимает готовый слой изолиний и автоматически рассчитывает профиль рельефа по указанной линии без дополнительных параметров; ручная правка узловых точек допускается по требованию пользователя.

– **Автоматический расчёт характеристик** – после подтверждения линии система вычисляет протяжённость, перепады высот и уклоны между опорными точками без ручного ввода формул или коэффициентов.

– **Графическое отображение результатов** – построенный профиль выводится в виде наглядного графика с отметками минимальной, максимальной и средней высоты, пригодного для визуальной оценки рельефа.

– **Локальное сохранение данных** – все созданные профили, координаты и рассчитанные параметры сохраняются во внутреннем хранилище устройства для последующего просмотра без подключения к сети.

– **Экспорт профиля** – пользователь может выгрузить профиль в распространённые форматы «PNG» и «PDF» единым действием «Экспорт / Поделиться».

– **Фоновая синхронизация** – при восстановлении связи приложение автоматически передаёт локально созданные профили и обновления в центральное хранилище, не отвлекая пользователя дополнительными запросами.

– **Защита данных пользователя** – результаты измерений и настройки приложения сохраняются при неожиданном отключении питания или сети, исключая потерю данных.

Перечисленные функциональные требования обеспечивают соответствие системы ожиданиям пользователей и заложенным целевым показателям, таким как удобство использования, надежность и расширяемость.

## 3.2 Архитектурный дизайн системы

### 3.2.1 Сервисно-ориентированная структура

Система состоит из нескольких основных компонент, каждая из которых выполняет определённую роль в общей сервисно-ориентированной структуре:

* **Клиентское приложение** – мобильное приложение, устанавливаемое на устройство пользователя, отвечает за интерфейс взаимодействия с пользователем. Оно предоставляет пользователю визуальные средства для авторизации, ввода данных и отображения результатов. Клиентское приложение реализует логику отображения и часть прикладной логики, связанную с взаимодействием с пользователем. Также на клиентском уровне реализуется локальное хранение данных для офлайн-режима и механизм синхронизации с сервером при восстановлении подключения;
* **Сервер приложений** – серверная часть системы представляет собой централизованный компонент, реализующий бизнес-логику приложения. Сервер принимает и обрабатывает запросы от клиентского приложения, выполняет вычисления для построения профилей местности на основе полученных геоданных. На сервере располагается логика аутентификации и авторизации: при обращении клиента сервер проверяет учетные данные пользователя, выдает разрешения и токены доступа, тем самым обеспечивая безопасность взаимодействия. Серверный компонент экспонирует функциональность через стандартный веб-интерфейс (REST API), чтобы мобильное приложение могло вызывать необходимые операции удалённо.
* **База данных** – для хранения данных используется объектно-реляционное хранилище, спроектированное с учётом предметной области. В базу данных сохраняются учетные записи пользователей, описания проектов, загруженные исходные данные, результаты расчетов профилей местности, истории изменений профилей, сгенерированные отчеты, а также логи действий пользователей.

Описанная сервисно-ориентированная структура делает систему масштабируемой. Каждый компонент может развиваться независимо.

При росте нагрузки возможно горизонтальное масштабирование серверного слоя и оптимизация базы данных, в то время как обновление клиентского приложения не затрагивает серверный код, и наоборот. Чёткое разделение ответственности между компонентами также упрощает тестирование и отладку системы на этапе разработки.

### 3.2.2 Взаимодействие клиентского и серверного слоёв

Диаграмма на уровне контейнеров отображает основные компоненты системы и взаимодействия между ними, представлена на рисунке 1.

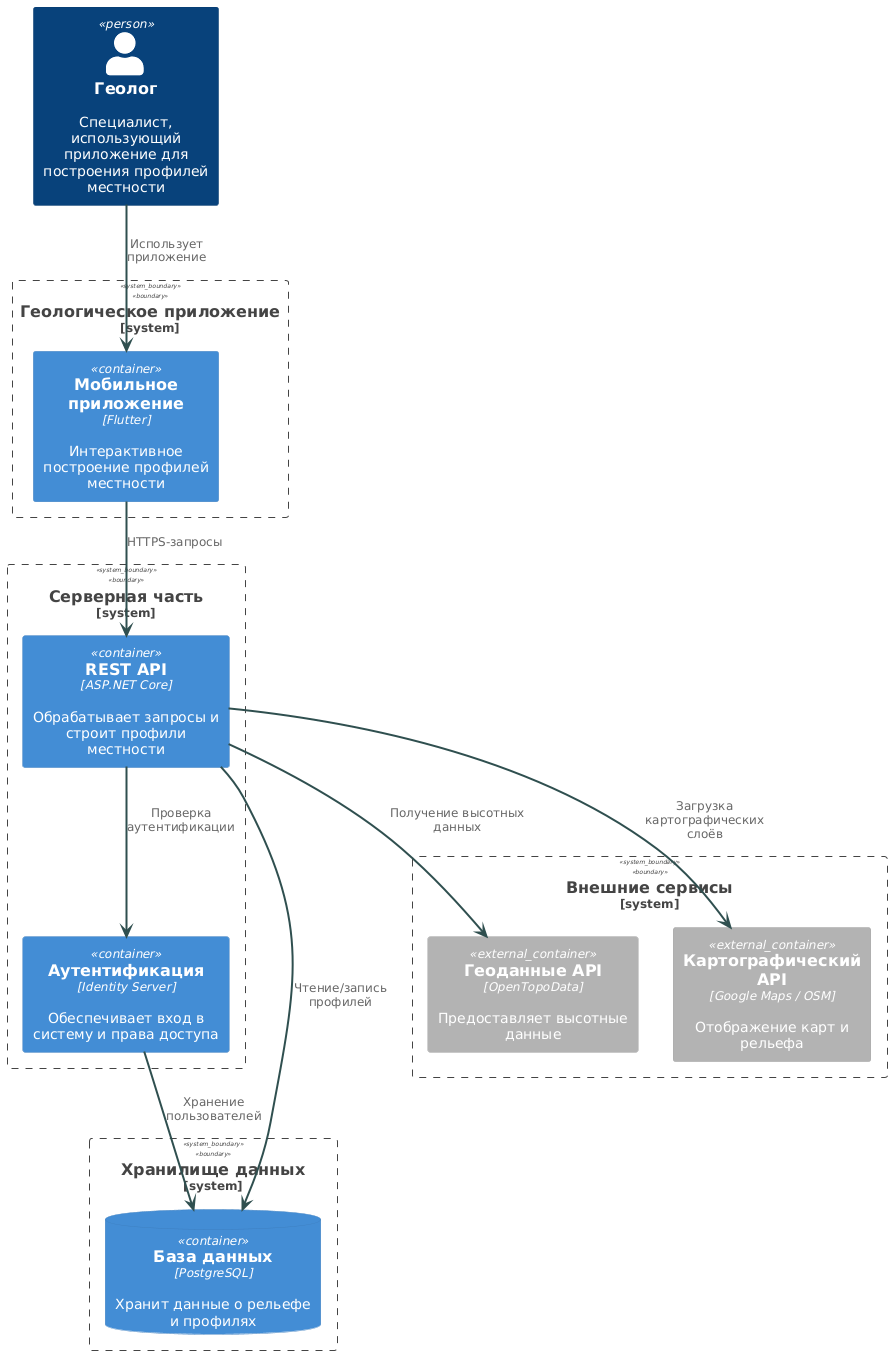


Рисунок 1 - С4 диаграмма архитектуры

На стороне клиента находится мобильное приложение (Mobile App), с которым непосредственно работает пользователь.

Мобильный клиент взаимодействует с сервером приложений, отправляя запросы через сеть и получая от сервера необходимые данные и ответы. Сервер приложений, в свою очередь, подключается к базе данных для чтения и записи информации.

Таким образом, взаимодействие клиентского и серверного уровней основано на четком разграничении ролей: **клиент** отвечает за представление данных и взаимодействие с пользователем, а **сервер** – за вычисления, хранение данных и интеграцию с внешними источниками. Подобная схема обеспечивает надёжность и безопасность: все тяжеловесные операции и хранение чувствительных данных выполняются на сервере, контролируемом разработчиками, тогда как на устройство пользователя поступают только необходимые результаты. Кроме того, использование стандартного протокола взаимодействия (веб-API) позволяет потенциально подключать к серверу и другие типы клиентов в будущем без изменения ядра системы – для этого достаточно реализовать новый интерфейс, отправляющий запросы к тому же серверу приложений.

## 3.3 Пользовательский интерфейс

Структура мобильного приложения включает несколько основных экранов, представленных в приложении А, отражающих ключевые функции системы.

**Экран авторизации** предоставляет форму входа пользователя в систему и ссылку на регистрацию нового аккаунта. **Экран регистрации** содержит поля для ввода необходимых данных нового пользователя и создания учетной записи. После успешного входа пользователь попадает на **главный экран**, который представляет обзор проектов пользователя – список сохранённых местностей. С этого экрана можно выбрать конкретный проект либо создать новый.

При выборе проекта открывается **экран с картой** – центральный интерфейс для работы с геоданными. На нем отображается карта местности с нанесенными исходными данными. Пользователь может выполнять навигацию по карте, добавлять или корректировать изолинии, вызывать функцию генерации профиля местности. Результат генерации – профиль высот – отображается на соответствующем **экране профиля**. Этот экран показывает график профиля и сопровождающую информацию.

Кроме основных рабочих экранов, предусмотрен **экран настроек**, где пользователь может управлять параметрами приложения. Также есть **экран профиля пользователя**, на котором отображаются сведения об учетной записи, возможности смены пароля и другие персональные настройки.

## 3.4 Модель «Сущность-связь»

В центре модели находятся сущности, отражающие предметную область: **Locations** и связанные с ними данные рельефа. Сущность **Users** представляет зарегистрированных пользователей приложения и хранит информацию о них.

Пользователи могут создавать и управлять проектами, поэтому между **Users** и **Locations** установлена связь, обозначающая, что пользователь работает над одной или несколькими локациями.

Сущность **Locations** описывает конкретные проекты или участки местности, над которыми ведется работа. Для каждой локации хранится уникальный идентификатор, название, описание и служебные временные метки. **Locations** находятся в центре нескольких связей: во-первых, с сущностью **Isolines**, во-вторых, с сущностью **Terrain Profiles**.

**Isolines** (изолинии) – сущность, хранящая данные о линиях равной высоты, используемые для моделирования рельефа. Каждая запись изолинии содержит координатную привязку и относится к конкретной локации. Связь между **Locations** и **Isolines** показывает, что каждой локации может соответствовать множество изолиний, описывающих ее рельеф.

**Terrain Profiles** – сущность, представляющая сгенерированные продольные профили рельефа. Профиль местности включает идентификатор, название или номер разреза, описание и пр. Каждый профиль связан с одной локацией, то есть создается в контексте определенного проекта местности. На основе данных изолиний и дополнительных параметров для локации может быть построено несколько профилей.

С профилями связана сущность **Profile Edits** (правки профиля). Она предназначена для хранения истории изменений профилей местности. Если пользователь вручную редактирует сгенерированный профиль, система фиксирует такую правку как отдельную запись. В **Profile Edits** сохраняется тип изменения, описание правки, временная отметка и другие метаданные. Эта сущность связана с профилями, отражая, что каждый профиль может иметь множество правок (версий) в истории.

**Reports** (отчеты) – сущность, отвечающая за хранение сведений о сформированных по результатам работы отчетах. Отчет представляет собой массив байтов с визуализацией профиля и сводной информацией. В таблице Reports сохраняются идентификатор отчета, тип или формат отчета, а также временные метки генерации и обновления. Каждый отчет связывается с конкретным профилем местности и может формироваться на основе текущего состояния профиля или выбранной версии правок. На диаграмме показано, что **Terrain Profiles** через связь могут порождать отчеты, а отчеты, в свою очередь, могут отражать изменения профиля.

Сущность **Logs** предназначена для записи событий и действий, происходящих в системе. Сюда относятся записи о том, какие действия выполнял пользователь: создание новой локации, генерация профиля, экспорт данных и т.п. В таблице **Logs** хранятся идентификатор события, описание действия или его тип, временная метка, а также поле для дополнительных метаданных. **Logs** связаны с сущностью **Users**, указывая на то, что каждое событие инициируется определенным пользователем. Журналирование позволяет отслеживать историю использования системы, проводить аудит изменений и при необходимости отлаживать работу приложения.

На рисунке 2 представлена сформированная ER модель.

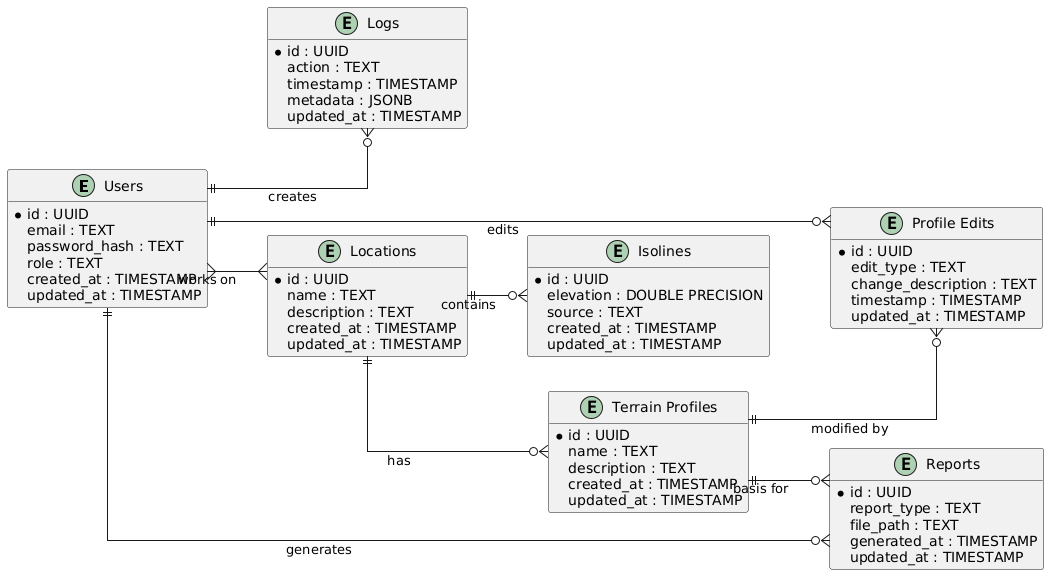


Рисунок 2 - ER модель

Таким образом, ER-модель базы данных описывает логическую структуру хранимых данных и взаимоотношения между ними.

# **ЗАКЛЮЧЕНИЕ**

В работе последовательно разобраны все ключевые элементы, необходимые для надёжного построения профиля рельефа. Проанализированы типы цифровых моделей высот, их происхождение и влияющие на точность факторы. Это позволило определить, какие исходные данные подходят для разных допусков по высоте.

Исследованы алгоритмы, преобразующие исходные высоты в непрерывный профиль: сравнивались интерполяционные методы и способы извлечения профиля через изолинии, оценивались их погрешности и вычислительная сложность, после чего выбран набор приёмов, обеспечивающий баланс между скоростью и точностью. Завершающий блок посвящён организации данных: обоснован выбор реляционного хранилища с расширением геопространственной функциональности; рассмотрены механизмы индексирования, позволяющие выполнять пространственные запросы к изолиниям и профилям с минимальными задержками, и показано, как транзакционные свойства системы гарантируют целостность данных при параллельных операциях.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] An algorithm for Contour lines [Электронный ресурс] – URL: <https://grgrdvrt.com/works/contour-lines>

[2] Marwaha N., Duffy E. Everything you need to know about Digital Elevation Models (DEMs), Digital Surface Models (DSMs), and Digital Terrain Models (DTMs) – UP42 Blog, 12 марта 2021 [Электронный ресурс] – URL: <https://up42.com/blog/everything-you-need-to-know-about-digital-elevation-models-dem-digital>

[3] ISGeography.com. Inverse Distance Weighting (IDW) Interpolation [Электронный ресурс] – URL: <https://gisgeography.com/inverse-distance-weighting-idw-interpolation/>

[4] PostgreSQL Global Development Group. PostgreSQL Official Website: About [Электронный ресурс]. – URL: <https://www.postgresql.org/about/>

[5] NextGIS. Пространственные индексы в PostGIS – раздел 1.5 курса «Веб-ГИС» [Электронный ресурс]. –

URL: [http://nextgis.github.io/webgis\_course/1/postgis\_spatial\_indexes.html](https://grgrdvrt.com/works/contour-lines)

# ПРИЛОЖЕНИЕ A

На рисунках 3-9 представлены экраны пользовательского интерфейса.

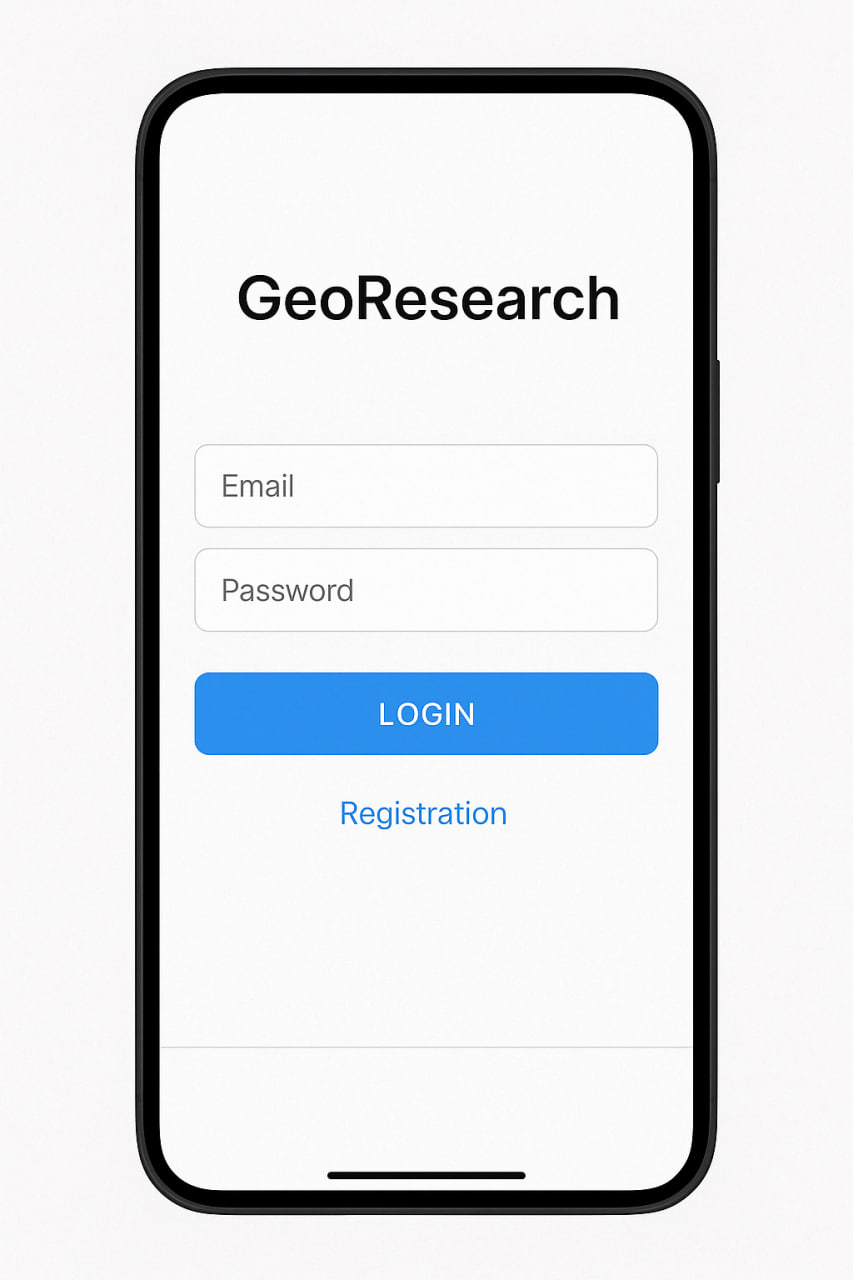


Рисунок 3 - Экран авторизации



Рисунок 4 - Экран регистрации

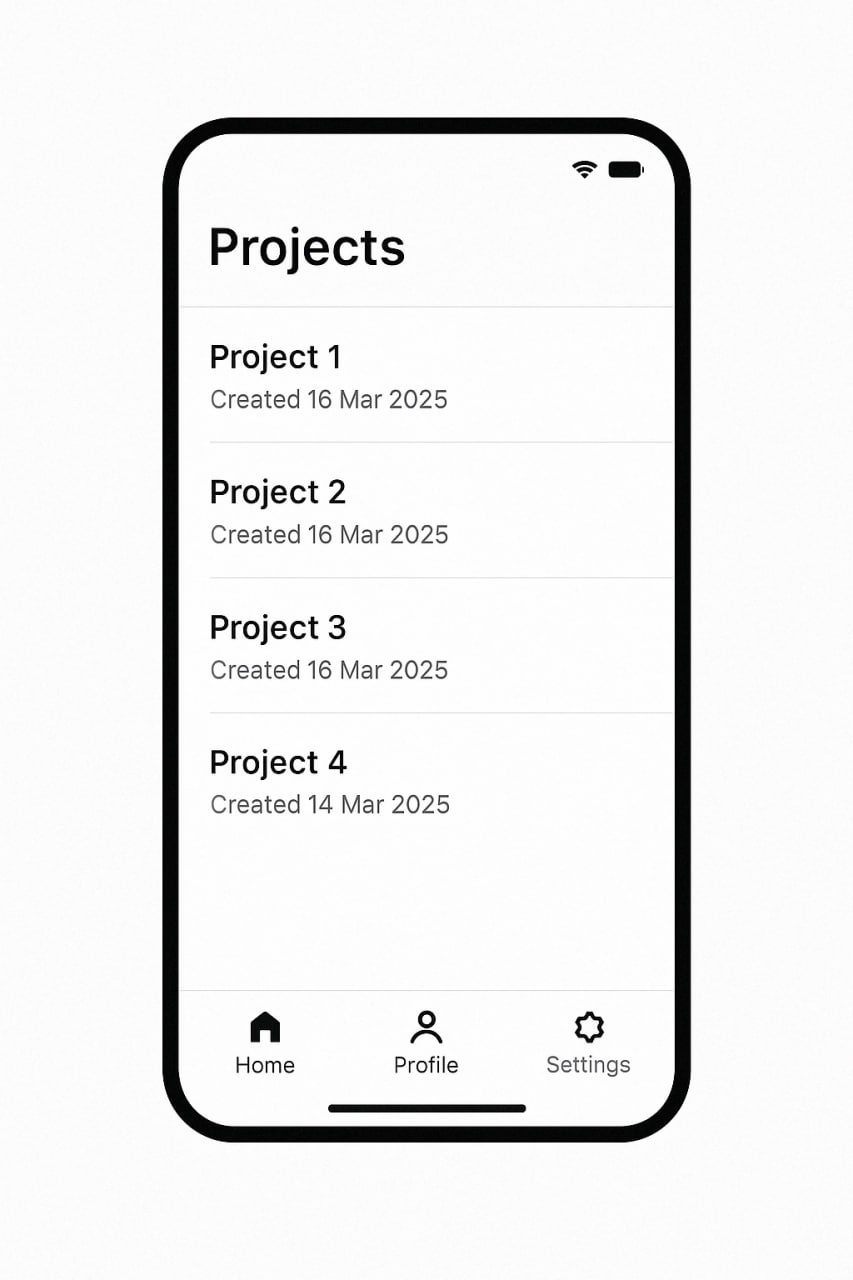


Рисунок 5 - Экран проектов

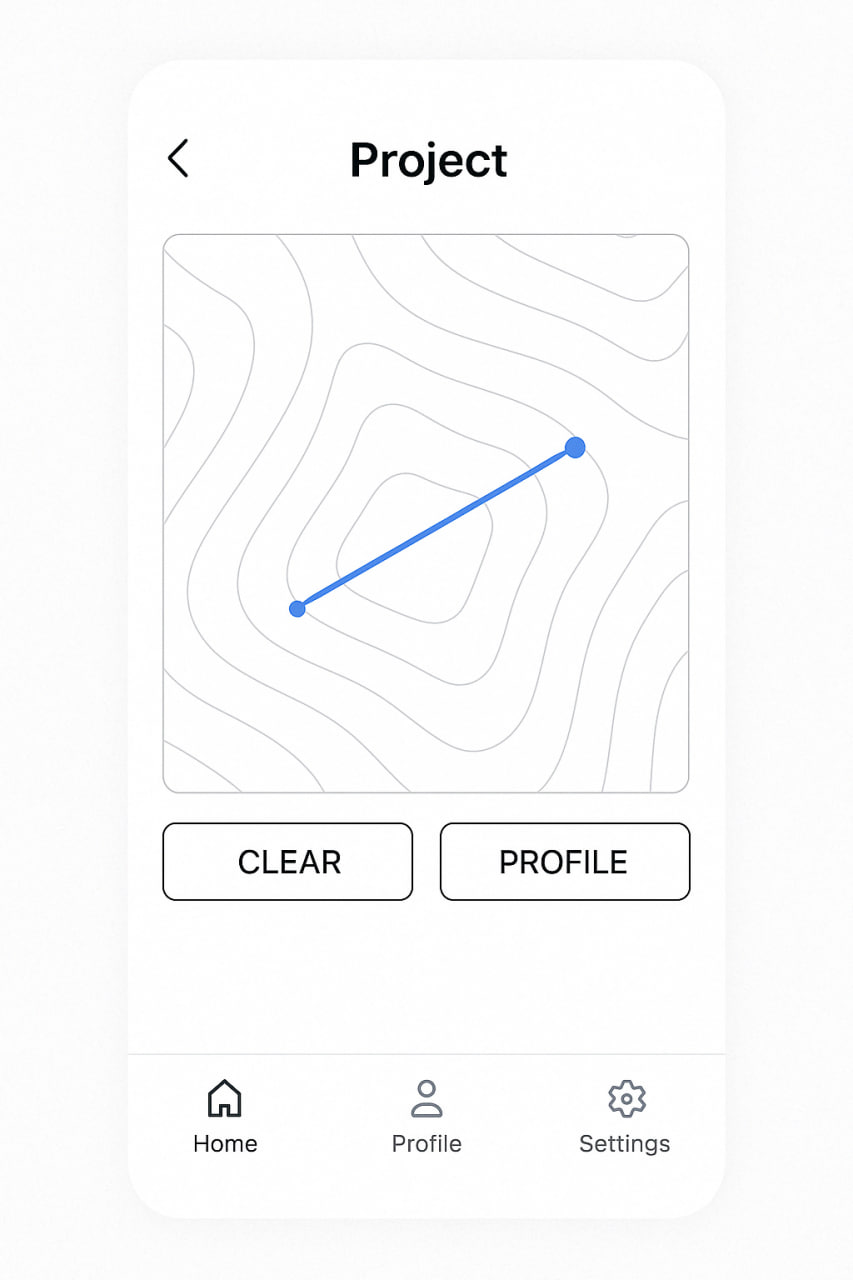


Рисунок 6 - Экран проекта

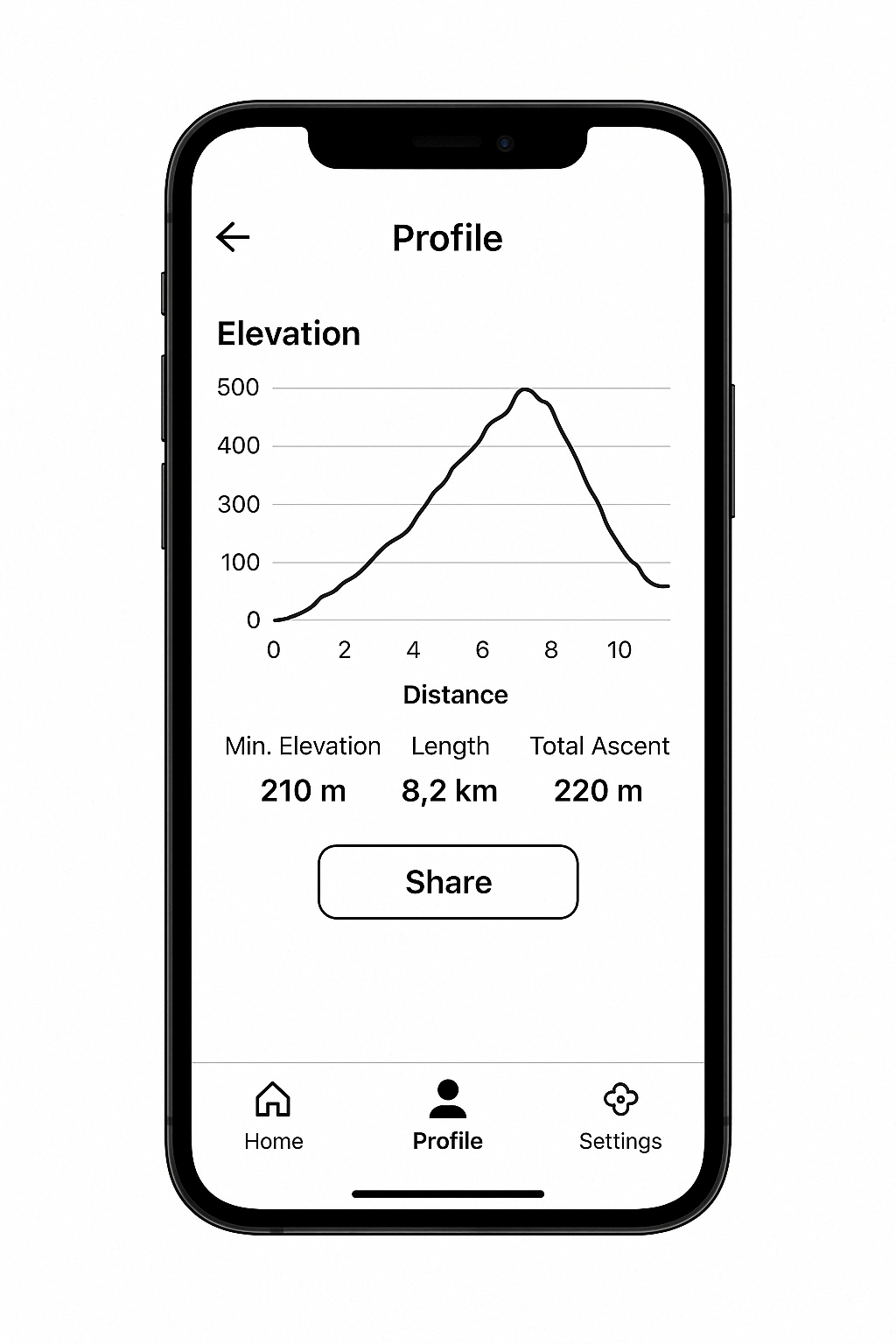


Рисунок 7 - Экран профиля рельефа

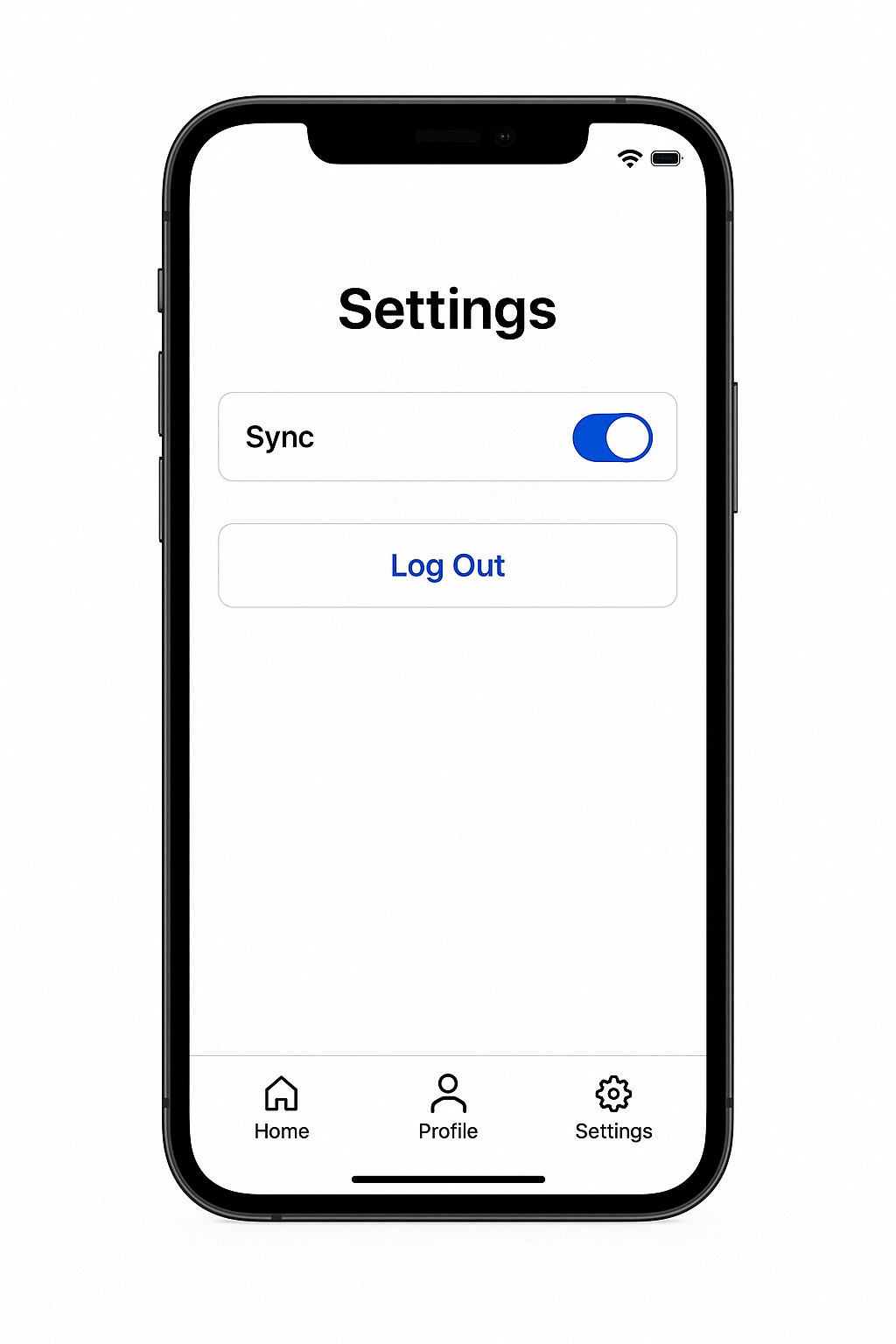


Рисунок 8 - Экран настроек

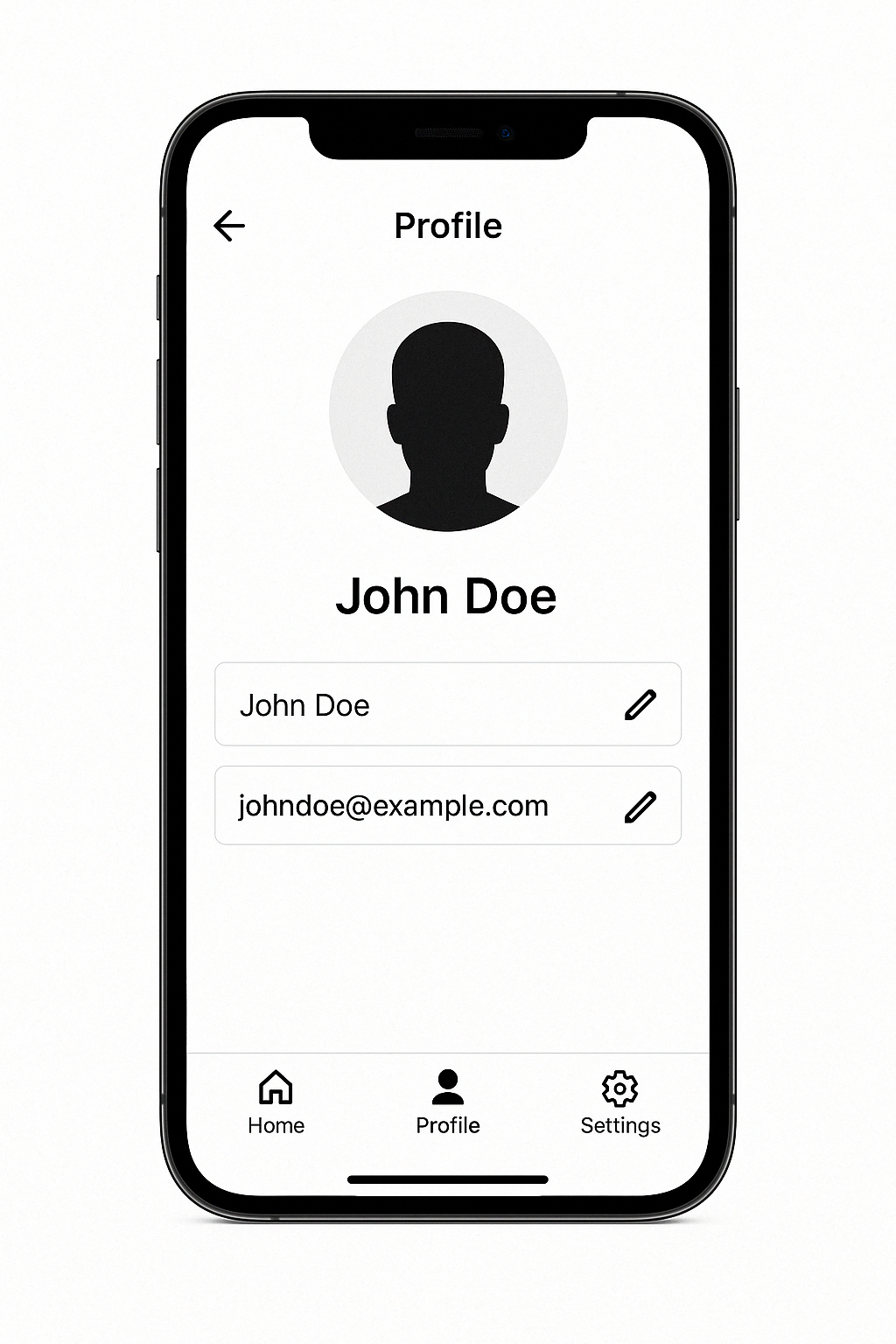


Рисунок 9 - Экран профиля пользователя