|  |  |
| --- | --- |
| 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](#_Toc196898777)

[1. Обзор предметной области 4](#_Toc196898778)

[1.1 Основные характеристики рельефа 4](#_Toc196898779)

[1.2 Источники и спецификации цифровых моделей местности 6](#_Toc196898780)

[1.3 Критерии точности и качества профиля рельефа 9](#_Toc196898781)

[2. Математическое описание методов построения профиля местности 12](#_Toc196898782)

[2.1 Методы пространственной интерполяции 12](#_Toc196898783)

[2.3 Формализация модели вертикального разреза 17](#_Toc196898784)

[3. Организация хранилища геоданных 19](#_Toc196898785)

[3.1 Архитектура PostgreSQL и обеспечение целостности 19](#_Toc196898786)

[3.2 Индексирование и управление пространственными данными в PostGIS 20](#_Toc196898787)

[4 Проектирование приложения 24](#_Toc196898788)

[4.1 Функциональные требования 24](#_Toc196898789)

[4.2 Архитектурный дизайн системы 25](#_Toc196898790)

[4.2.1 Сервисно-ориентированная структура 25](#_Toc196898791)

[4.2.2 Взаимодействие клиентского и серверного слоёв 26](#_Toc196898792)

[4.3 Пользовательский интерфейс 28](#_Toc196898793)

[4.4 Модель «Сущность-связь» 29](#_Toc196898794)

[ЗАКЛЮЧЕНИЕ 32](#_Toc196898795)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 33](#_Toc196898796)

[Приложение A 34](#_Toc196898797)

# **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 цифровые модели местности (Digital Elevation Models) и 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.

Кроме чисто вычислительных вопросов возникает проблема согласованности исходных данных: разнородные DEM часто различаются разрешением, системами координат и систематическими смещениями. Поэтому исследование дополняется анализом методов предварительной нормализации и оценки качества входных моделей, а также формулировкой критериев точности выходного профиля. Особое внимание уделяется устойчивости алгоритмов к шуму и пропускам в исходном наборе точек, что критично в условиях полевых измерений.

# **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 Источники и спецификации цифровых моделей местности

Цифровая модель местности (ЦММ) – это обобщенное понятие для цифрового представления поверхности земной территории. На практике часто разделяют понятия цифровая модель рельефа (ЦМР) и цифровая модель поверхности:

* Цифровая модель рельефа (Digital Terrain Model, DTM) обычно подразумевает модель именно земной поверхности без учета надземных объектов, где учтены высоты рельефа (земной коры) без деревьев, построек и прочих объектов​.
* Цифровая модель поверхности (Digital Surface Model, DSM) – это модель, включающая верхнюю поверхность всех объектов, т.е. высоты по верхушкам деревьев, крышам зданий и т.д.​

Цифровая модель возвышений (Digital Elevation Model, DEM) – термин, употребляемый в литературе не строго: часто DEM используют как общее название для любой модели высот​. В ряде случаев DEM отождествляют с DTM, или наоборот с DSM, в зависимости от контекста​. Многие картографические организации (например, USGS) применяют термин DEM как общий для любой цифровой высотной модели​. В отечественной терминологии ЦМР обычно соответствует понятие цифровая модель рельефа (близкое DTM), а цифровая модель местности может включать информацию о поверхностных объектах (то есть быть аналогом DSM).

Спецификация ЦМР включает способ представления рельефа (структуру данных), а также точность и разрешающую способность модели. Основные форматы представления:

* **Регулярная сетка высот** **(GRID)** – наиболее распространенный формат DEM. Представляет собой матрицу значений высот с заданным шагом сетки. Каждый элемент матрицы (пиксел) соответствует высоте рельефа в центре этого пиксела. Пространственное разрешение модели определяется размером пикселя;
* **Нерегулярная триангуляция (TIN – Triangulated Irregular Network)** – модель в виде совокупности неравносторонних треугольников, покрывающих поверхность. Узлы треугольников задаются в характерных точках рельефа, а внутри каждого треугольника высота интерполируется по плоскости. TIN является векторной моделью и хорошо передает сложные участки рельефа с разной плотностью узлов: на ровных местах узлов мало, в пересеченной местности – больше. Однако TIN сложнее в реализации и менее совместима с растровыми алгоритмами анализа;
* **Изолинии (горизонтали)** – хранение рельефа в виде набора изолиний (замкнутых кривых на определенных высотах, обычно через заданный интервал – шаг сечения).

Этот формат характерен для классических топокарт и может рассматриваться как вид векторной модели рельефа. Для использования в вычислительных задачах изолинии часто преобразуют в регулярную сетку или TIN, так как не каждое место на модели имеет явную высоту (известны только линии равных высот).

1.3 Методы построения и применение изолиний

Изолинии – это линии равных значений некоторой величины на плоскости. В случае рельефа изолинии равных высот называют горизонталями или изогипсами. Каждая такая линия соединяет точки местности с одинаковой абсолютной высотой над уровнем моря​. На топографических картах горизонтали обычно проводятся через регулярный интервал высот. Набор таких изолиний дает двумерное изображение рельефа: по их плотности и форме можно судить о крутизне склонов и очертаниях форм рельефа. Близко расположенные линии означают крутой склон, а разреженные – пологий​.

Построение изолиний по числовым данным рельефа – задача, решаемая методами изолинирования. Если имеется непрерывная функция высоты , то изолиния для высоты есть геометрическое место точек, удовлетворяющих уравнению .

В случае цифровой модели эта функция задана дискретно. Задача построения горизонталей сводится к поиску на каждой ячейке сетки линии, соединяющей точки, где высота равна нужному значению. В традиционной картографии горизонтали чертили вручную по результатам съемки, используя экспертное понимание рельефа. В компьютерной реализации используется алгоритм «марширующих квадратов» (marching squares) [1] для растровых данных​.

Этот алгоритм проходит по клеткам регулярной сетки и, анализируя соотношение высот углов клетки с уровнем определяет конфигурацию пересечения горизонтали с границами клетки. Для каждой из восьми возможных комбинаций (углы выше/ниже уровня) в алгоритме заранее рассчитано, как пройдет линия внутри квадрата.

Марширующие квадраты позволяют построить одну изолинию заданного уровня за один проход по сетке. Для построения полного набора горизонталей алгоритм повторяется для каждого уровня, либо модифицируется для одновременного выделения множества контуров. Стоит отметить, что алгоритмы построения изолиний из GRID-данных работают автоматически, но их результаты нуждаются в сглаживании и проверке топологии. Например, метод marching squares по умолчанию порождает ломаные сегменты, соответствующие переходу через клетки, и хотя при достаточно мелкой сетке визуально получается гладкая кривая, иногда применяют сплайн-интерполяцию, чтобы сгладить изогипсы.

Кроме того, возможно появление артефактов – ложных замкнутых петель или разрывов – если в данных есть шум. Поэтому после вычислительного построения горизонталей часто применяются процедуры очистки.

Альтернативный подход – построение изолиний из TIN-модели. В этом случае пересекаются горизонтальные плоскости с каждым треугольником сети.

На каждом треугольнике линия уровня – либо отрезок, либо отсутствует (если весь треугольник выше или ниже заданного уровня). Собрав отрезки со всех треугольников, получают полный контур. Этот подход дает линейные сегменты на плоскостях треугольников, которые обычно менее гладкие, чем результаты marching squares.

Если исходная информация о рельефе представлена только горизонталями, то чтобы получить профиль вдоль произвольной линии, сначала нужно из этого набора изогипс восстановить функцию высоты.

Один из методов – провести профиль, найдя точки пересечения профильной линии с каждой горизонталью определенной высоты, и интерполировать высоту между ближайшими горизонталями.

Зачастую после создания цифровой модели рельефа строят горизонтали для проверки. По набору изолиний можно быстро оценить форму рельефа. Кроме того, изолинии, будучи векторным представлением рельефа, удобны для хранения и могут использоваться для ускорения расчета профилей.

## 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.

# **2. Математическое описание методов построения профиля местности**

## 2.1 Методы пространственной интерполяции

Для получения непрерывного профиля рельефа по дискретным исходным данным (точкам высот или изолиниям) необходимо использовать методы пространственной интерполяции. Интерполяция позволяет оценить высоту в произвольной точке на основании известных значений в окружающих точках. Существует множество методов интерполяции рельефа, различающихся предположениями о характере изменения высоты и дающих различное качество результатов. Детерминированные методы интерполяции основываются непосредственно на известных значениях без привлечения случайных моделей. К ним относятся:

* **Метод ближайшего соседа** – простейший способ, когда каждой неизвестной точке присваивается значение ближайшей известной точки. Для профиля это означало бы, что высота берется равной высоте ближайшего узла сетки или ближайшей точки измерения. Этот метод дает ступенчатую модель и редко применяется для профилей, кроме как для грубой оценки;
* **Интерполяция по TIN** – если построена триангуляционная сеть из точечных данных, то внутри каждого треугольника высота определяется линейно по плоскости, проходящей через значения в вершинах треугольника. Такой подход гарантирует, что на самих известных точках профиль будет точно совпадать с ними (метод точной интерполяции), а между ними – линейно;
* **Инверсионно-дистанционное взвешивание (IDW – Inverse Distance Weighting)** – метод, в основе которого предположение: ближайшие точки влияют сильнее, чем дальние. Для неизвестной точки вычисляется взвешенное среднее известных высот, где вес каждой точки обратнопропорционален расстоянию до этой точки (обычно в степени )​. Формула интерполяции IDW [3] для высоты в точке выглядит как:

где – известная высота в -й точке, – расстояние от точки до этой точки, – параметр степени (обычно )​. При влияние убывает пропорционально квадрату расстояния. В реализации часто берут ограниченное число ближайших точек или радиус поиска, чтобы исключить очень далекие;

* **Сплайн-интерполяция (сплайн-поверхности**) – метод, при котором через известные точки проводятся гладкие кривые (сплайны), обеспечивающие непрерывность первой и второй производных. В двумерном варианте используются билинейные или бикубические сплайны на сетке, или радиальные базисные функции. Цель – получить гладкую поверхность, максимально близкую к точкам. Сплайны хорошо подходят для профилей, т.к. дают визуально сглаженную линию без изломов;
* **Метод напряженных сплайнов (Thin Plate Spline)** – разновидность сплайна, который минимизирует изгиб поверхности, как будто натянута тонкая пластина через точки. Часто используется для рельефа, обеспечивает очень гладкую поверхность, но может сглаживать реальные острые формы (например, пик может быть снижен);
* **Геостатистические методы (стохастические)** – предполагают наличие статистической структуры в данных высот. Самый известный – кригинг (kriging). В отличие от детерминированных подходов, кригинг основывается на модели пространственной автокорреляции данных​. Он рассматривает высоту как случайную величину, имеющую некоторую структуру ковариации. Кригинг оценивает неизвестную высоту как линейную комбинацию имеющихся, подобно IDW, но веса подбираются, с учетом варьирования данных: ближние точки получают вес по принципу, зависящему от эмпирической автокорреляции на данном расстоянии. При этом метод может предоставлять не только оценку высоты, но и оценку погрешности (дисперсии ошибки) в данной точке, чего нет у детерминированных методов​.

При использовании линейной интерполяции (TIN или между изолиниями) профиль может иметь изломы на точках данных. IDW сгладит изломы, но может недооценить пик, если вокруг нет точки именно на пике. Сплайн сделает профиль гладким, проходящим через исходные точки, но потенциально с небольшими колебаниями. Кригинг обеспечит наилучшее статистически обоснованное приближение, но требует больше параметризации.

2.2 Алгоритмы извлечения профиля по изолиниям

Пусть на местности задана некоторая линия профиля – обычно пользователем на карте выбирается начало и конец, возможно ломаная линия через несколько точек. Требуется по имеющимся горизонталям (изолиниям) построить профиль – то есть определить высоты пересечения рельефа этой линией. Предположим, что в распоряжении есть набор изолиний с определенным шагом сечения и известными значениями высот этих изолиний. Алгоритм профилирования по горизонталям можно описать так:

* **Нахождение точек пересечения** – для каждой изолинии (для каждого уровня) вычисляется пересечение с линией профиля. Каждая горизонталь представлена набором полилиний (векторных сегментов). Линия профиля тоже может быть ломаной (состоящей из сегментов между заданными пользователем точками). Требуется найти все координаты , где -я изолиния пересекает -й сегмент профиля. Это сводится к пересечению отрезков на плоскости. Результатом будет набор точек, каждая из которых лежит на профиле и на некоторой горизонтали известной высоты . Таким образом, этим точкам можно присвоить высоты пересекаемой изолинии;
* **Сортировка точек по расстоянию вдоль профиля** – все найденные точки пересечения сортируются в порядке следования по профильной линии от начала к концу. Каждая точка характеризуется расстоянием от начала профиля и высотой . Также в начало профиля добавляется первая точка с известной высотой;
* **Интерполяция между пересечениями** - между найденными точками пересечения профильной линии с горизонталями существуют участки профильной линии, на которых профиль проходит между горизонталями. На каждом таком участке граничными точками являются два соседних пересечения с изолиниями высот и . Поскольку между двумя соседними горизонталями высота меняется монотонно (если на профиле нет других пересечений между ними, значит профиль не достигает какой-то третьей горизонтали между и ), можно предположить линейное изменение высоты. Тогда высоту любой точки между этими двумя можно получить линейной интерполяцией по расстоянию. Но часто возникает ситуация, что начало профиля может начинаться между горизонталями. В таком случае, стартовую высоту профиля можно оценить экстраполяцией: продолжить линейный тренд последнего сегмента или, более правильно, определить высоту начала профиля из ближайшей горизонтали. Если начало профиля точно совпадает с известной точкой, то это точка якорная;
* **Построение сегментов профиля** – cоединив последовательные точки (пересечения и крайние точки), получается ломаная линия "высота - расстояние". Эта ломаная и есть профиль. Он будет представлять собой отрезки, каждый из которых соответствует интервалу между двумя соседними горизонталями на карте. По определению линейной интерполяции, между горизонталями профиль выходит прямолинейный (уклон постоянный);
* **Учет особых случаев** – особый случай, когда профильная линия проходит точно по горизонтали на некотором протяжении. Тогда она пересекает ее не в точке, а на отрезке. В результате, множество точек пересечения с одной высотой, что не является проблемой – их можно свести к двум крайним точкам этого отрезка, а между ними профиль остается на том же уровне. Другой случай – профиль проходит через вершину или впадину, которая не отмечена горизонталью (например, пик между горизонталями). Тогда на профиле появится плавный перелом: сначала подъем до максимума, затем спад. Но если данных нет о том, что там пик, то профиль останется линейным и недооценит пик (реальная вершина выше, чем линейная интерполяция между горизонталями). Чтобы улучшить ситуацию, часто комбинируют горизонтали с дополнительными точками – отметками вершин, если они есть на карте.

Таким образом, базовый алгоритм на основе горизонталей дает ломанный профиль. Чтобы его улучшить, можно применить сглаживание. Один из приемов – провести сплайн через полученные точки пересечения, обеспечив более плавный переход, особенно если известно, что форма склона плавная.

Другой подход – построение TIN по горизонталям: изолинии представляют собой множество точек (вершин ломаных). Если взять все вершины всех изолиний и дополнительно случайные точки на самих линиях, можно построить триангуляционную сеть. Затем профиль строится как пересечение этой сети треугольников вертикальной плоскостью, заданной профильной линией.

Геометрически, линия профиля – это прямая (или ломаная) на плоскости . Если представить вертикальную поверхностную цилиндрическую призму над этой линией, то пересечение боковых плоскостей призмы с плоскостями треугольников TIN и даст профиль.

Практически, алгоритм пересечения линии с TIN: разбить профильную линию на достаточно мелкие сегменты, для каждого вычислить высоту путем интерполяции на треугольнике, в который он попадает. Это эквивалентно тому, что описано выше, но может выявлять случаи пиков лучше, если они заданы точками в TIN (например, отдельно была добавлена точка вершины).

## 2.3 Формализация модели вертикального разреза

Под моделью вертикального разреза (профиля) понимается математическое или геометрическое представление, описывающее полученный профиль.

Рассмотрим профиль как функцию высоты от пройденного расстояния. Пусть профильная линия на местности задана в плановых координатах как параметрическая кривая на плоскости , (параметр может выражать долю пройденного пути от начала до конца). Тогда каждая точка профиля имеет координаты .

Имея функцию высот , можно определить высоту на профиле как . Таким образом, профиль можно выразить как параметрическую пространственную кривую

.

Эта кривая лежит на поверхности рельефа. Однако для построения вертикального профиля нас больше интересует зависимость от пройденного расстояния вдоль профиля, чем от .

Введем длину дуги профиля – расстояние по горизонтальной проекции профиля от начала. Если линия профиля разбивается на сегменты с длинами в горизонтальной плоскости (например, по координатам ), то накопленная длина .

Нормируя, можно использовать либо физический параметр (в метрах), либо относительный , где – общая длина профиля по горизонтали.

Тогда профиль удобно представить как функцию – высота как функция длины вдоль профиля. Это уже функция одной переменной, которую можно анализировать обычными методами. В частности, уклон профиля в данной точке – это производная , которая численно равна тангенсу угла наклона местности.

Если нужно учесть истинную длину по склону, то можно ввести длину по поверхности, но обычно профили строят в проекции на горизонтальную линию. Функция получается дискретно – как набор точек после построения профиля. Далее ее можно аппроксимировать аналитически, например, многочленом или сплайном.

Для хранения профиля часто применяют формат набора точек с равным шагом по или список вершин ломаной. Если профиль очень длинный и деталей много, его можно обобщить с помощью упрощения полилинии для отбрасывания избыточных точек, сохраняя форму в пределах допуска по высоте.

# **3. Организация хранилища геоданных**

## 3.1 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.

## 3.2 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.

# 4 Проектирование приложения

## 4.1 Функциональные требования

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

* **Интерактивное построение профилей местности** – система должна принимать исходные геоданные и автоматически генерировать профиль местности на их основе. Пользователь должен иметь возможность при необходимости вносить ручные правки в сгенерированный профиль для учета экспертных корректив;
* **Кроссплатформенная мобильность** – клиентское приложение должно функционировать на распространённых мобильных платформах, обеспечивая единообразный интерфейс и набор функций на разных устройствах;
* **Офлайн-режим и синхронизация** – приложение должно поддерживать работу в условиях временного отсутствия интернета. В офлайн-режиме данные о проекте (карты, профили местности и пр.) сохраняются локально на устройстве. При восстановлении соединения система выполняет синхронизацию локальных изменений с сервером, обеспечивая консистентность данных между клиентом и серверной базой данных;
* **Безопасность и аутентификация** – необходимо реализовать надёжную систему аутентификации и авторизации пользователей. Доступ к функциям приложения должен осуществляться только после подтверждения личности пользователя. Система разграничения прав доступа должна обеспечивать защиту персональных данных и предотвращать несанкционированный доступ;
* **Экспорт и генерация отчетов** – система должна предоставлять возможность экспорта полученных результатов в распространённые форматы данных. Пользователь должен иметь возможность сохранить профиль местности и сопутствующую информацию в виде PDF отчёта;
* **Мониторинг и журналирование** – для обеспечения надежной эксплуатации система должна включать средства мониторинга и журналирования. Необходимо регистрировать ключевые действия пользователей и системы (журналы событий) и следить за состоянием серверных компонентов. Это позволит оперативно обнаруживать неполадки и получать статистику использования, что важно для последующего совершенствования приложения.

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

## 4.2 Архитектурный дизайн системы

### 4.2.1 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.

### 4.2.2 Взаимодействие клиентского и серверного слоёв

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

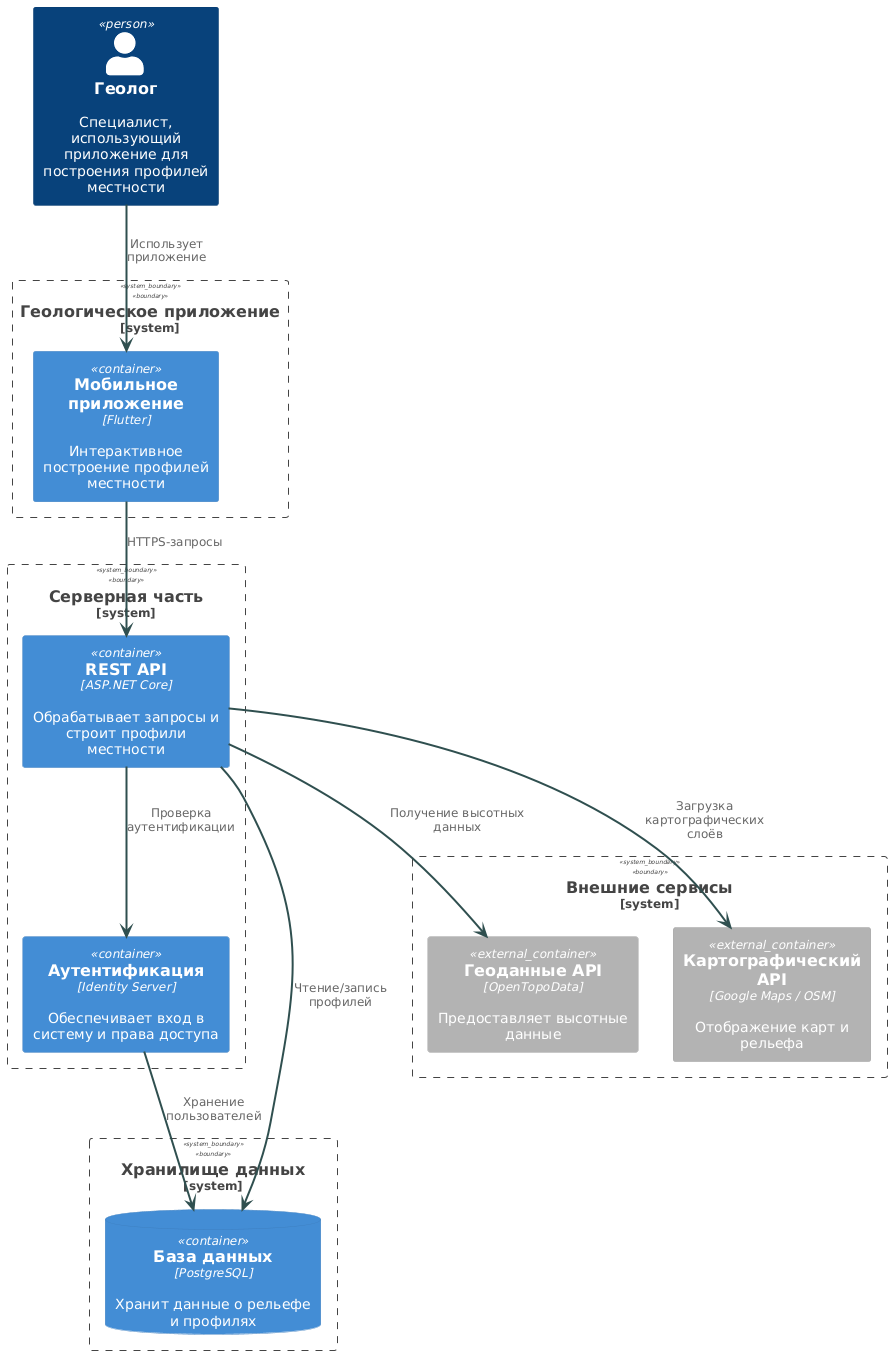


Рисунок 1 - С4 диаграмма архитектуры

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.

## 4.3 Пользовательский интерфейс

Структура 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.

# 4.4 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.

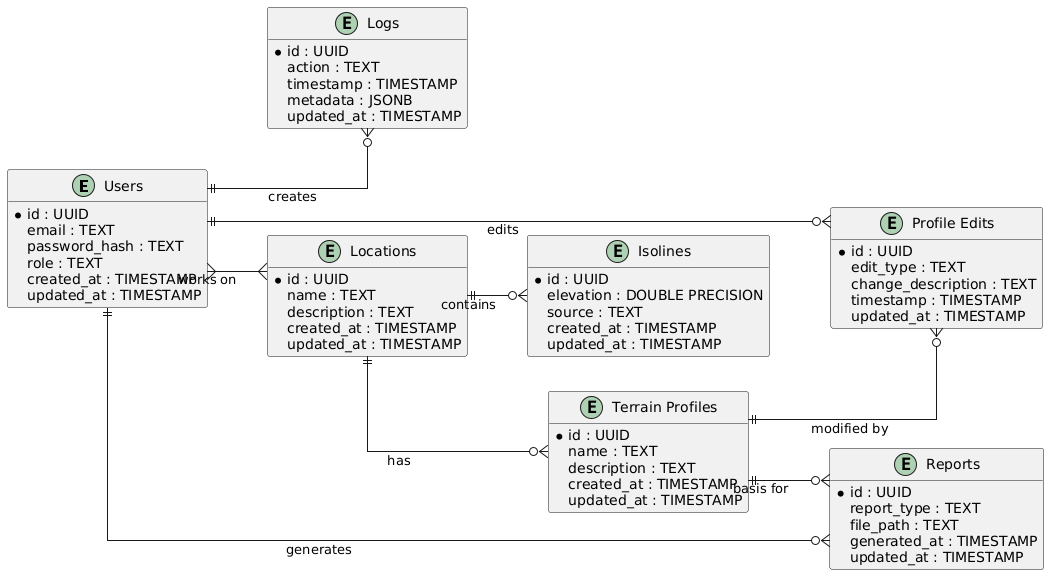


Рисунок 2 - ER модель

Таким образом, ER-модель базы данных описывает логическую структуру хранимых данных и взаимоотношения между ними. Эта модель легла в основу создания схемы базы данных и обеспечивает хранение всех необходимых сведений для функционирования приложения. Она поддерживает целостность данных (через связи и ограничения целостности на уровне СУБД) и позволяет эффективно выполнять запросы для выборки данных по проектам, профилям, пользователям и т.д.

# **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.

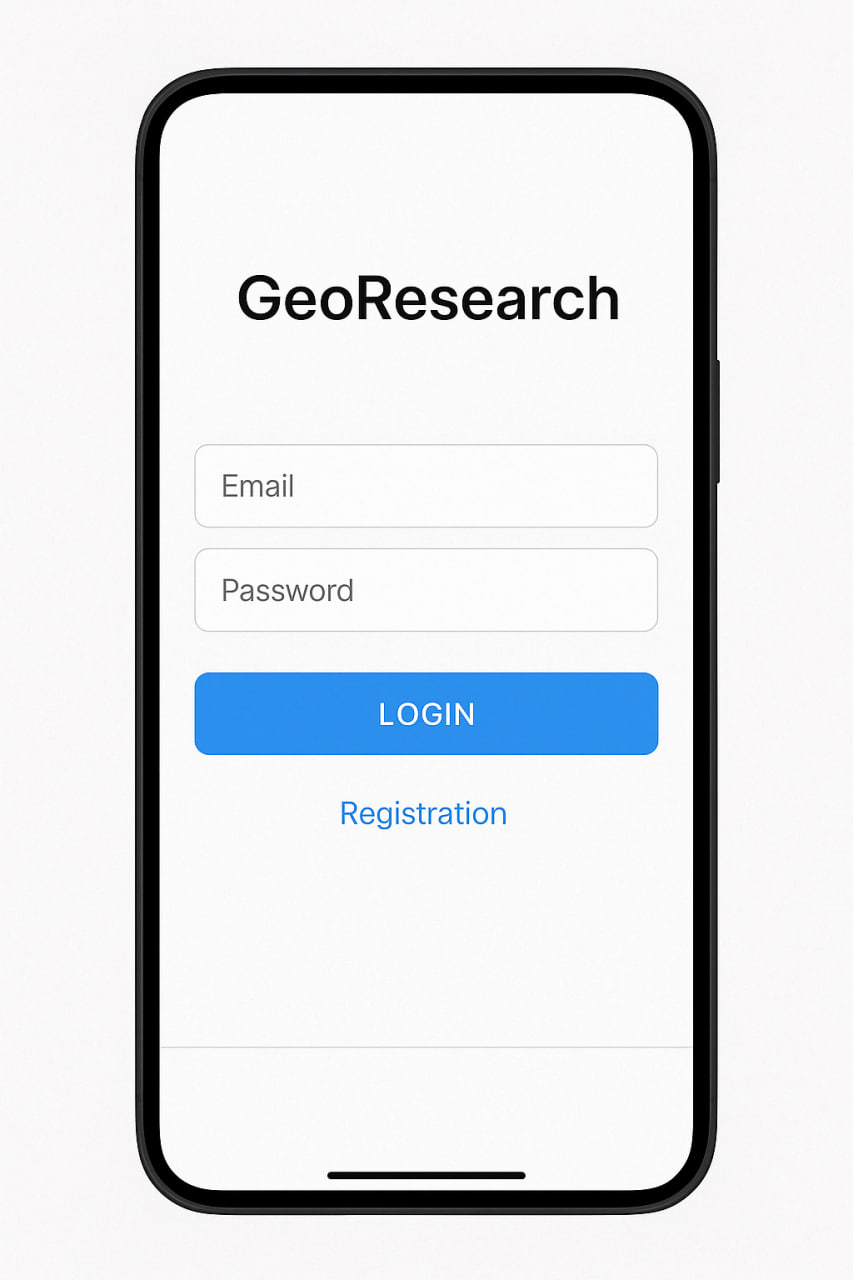


Рисунок 3 - Экран авторизации



Рисунок 4 - Экран регистрации

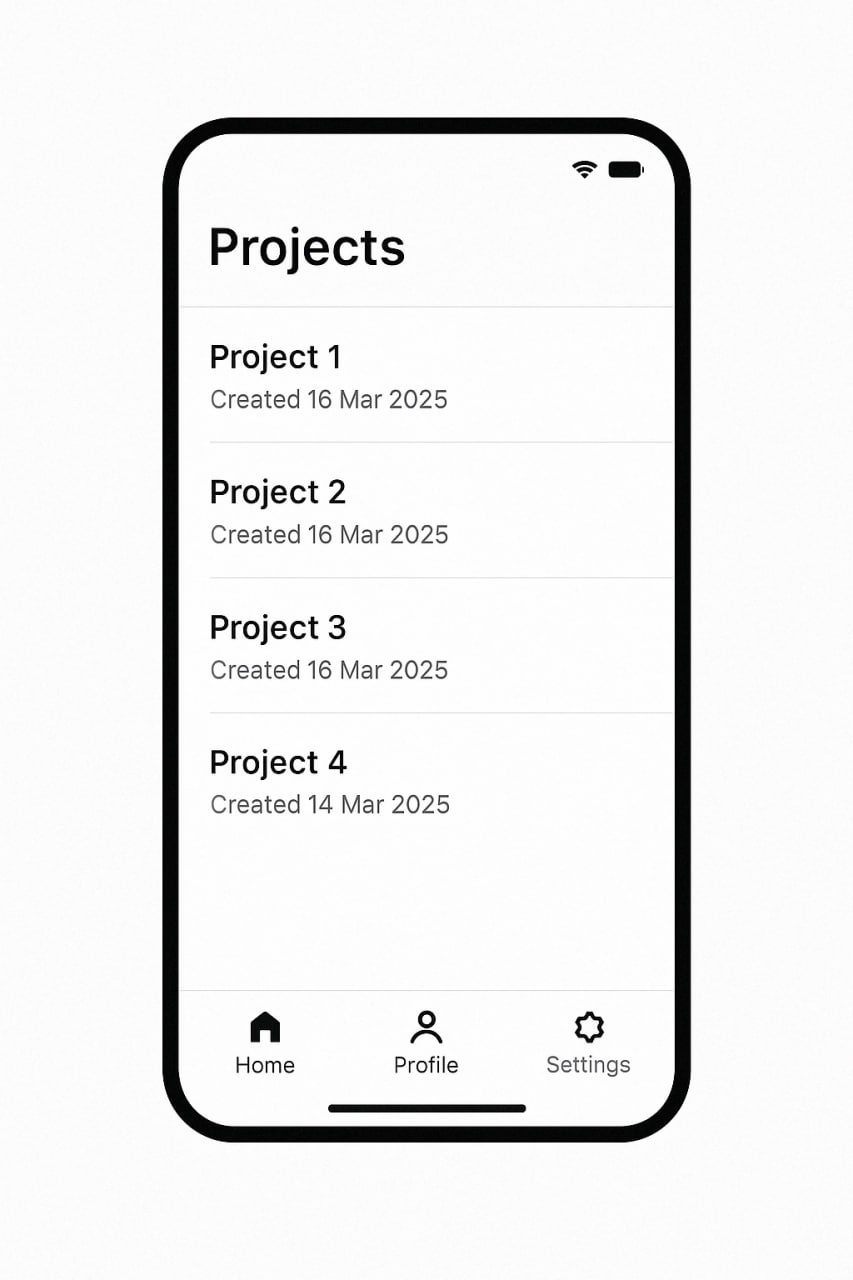


Рисунок 5 - Экран проектов

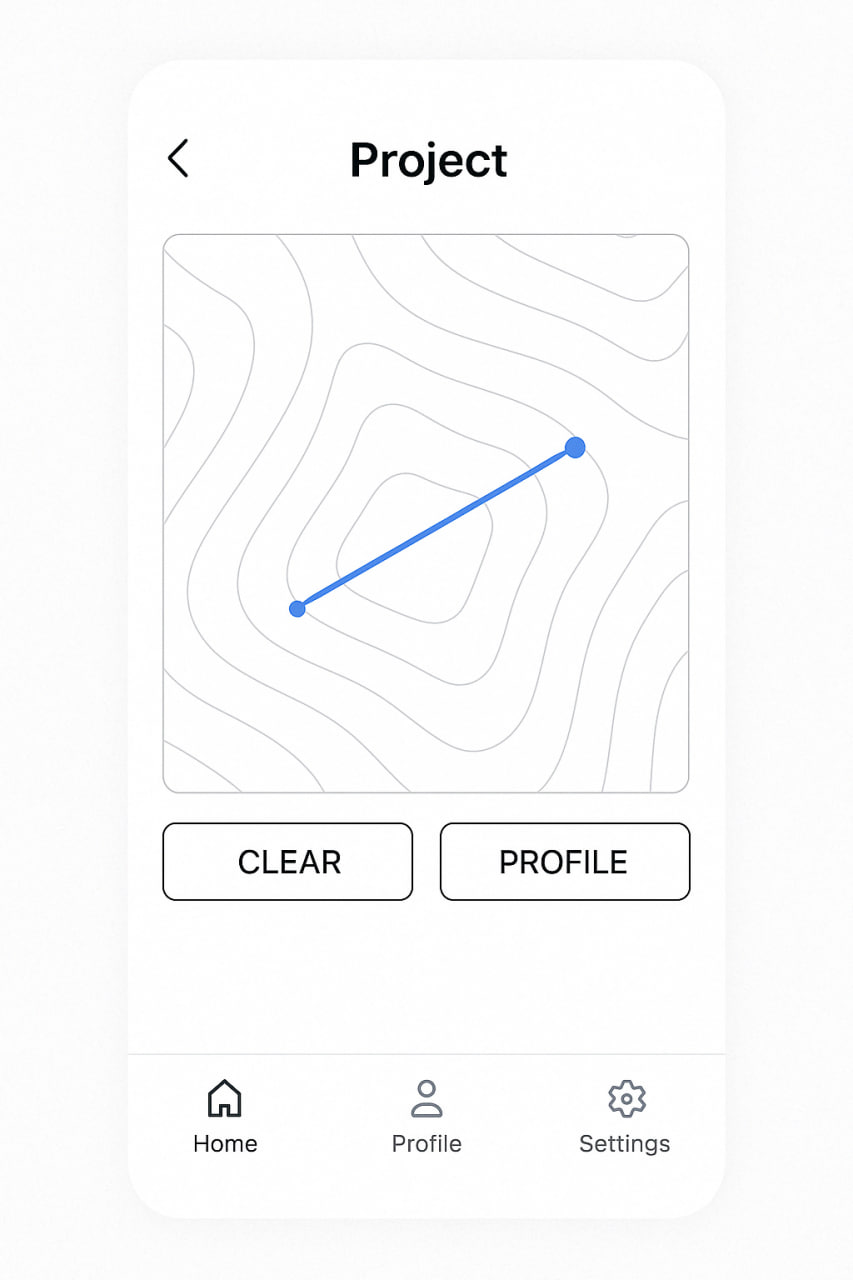


Рисунок 6 - Экран проекта

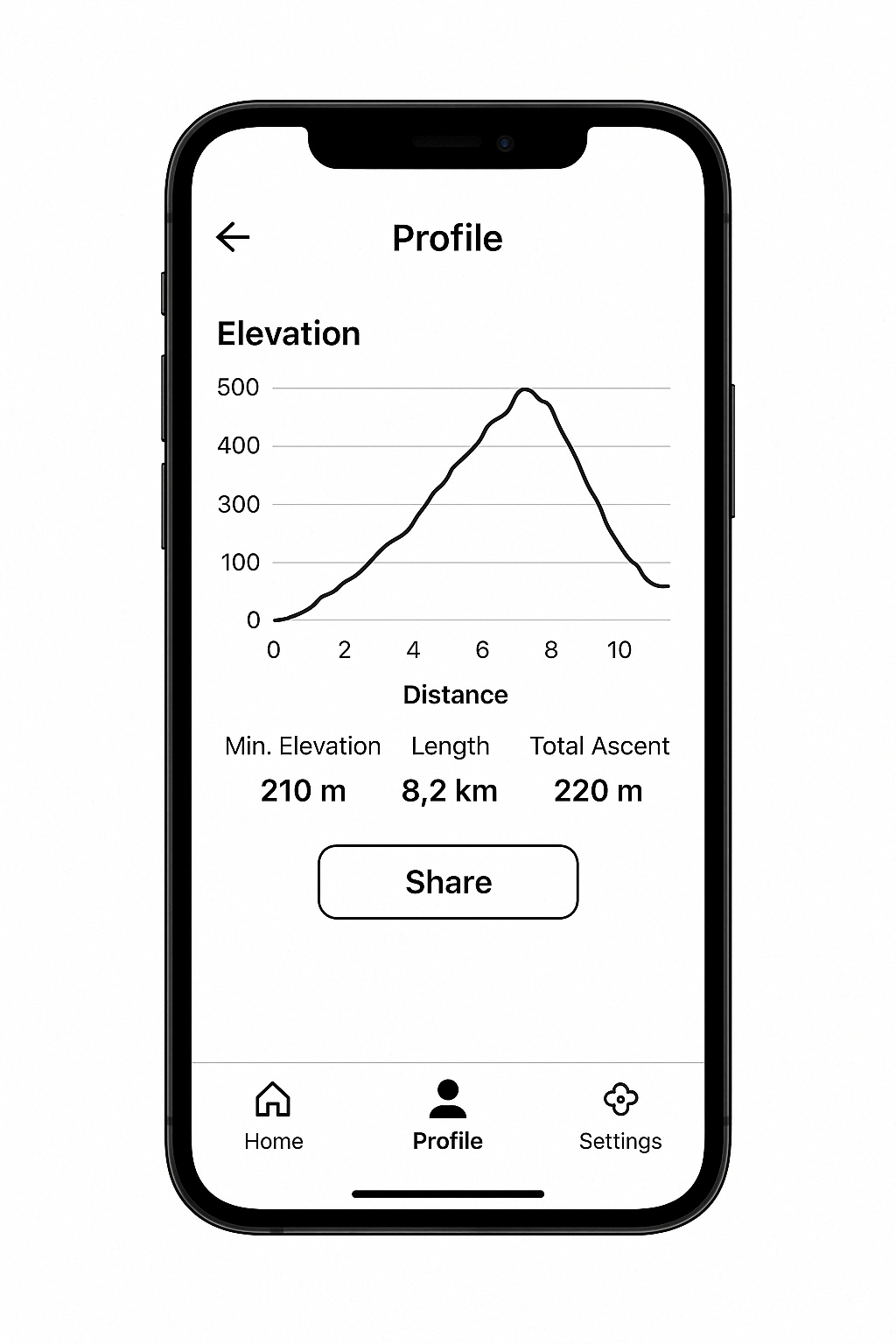


Рисунок 7 - Экран профиля рельефа

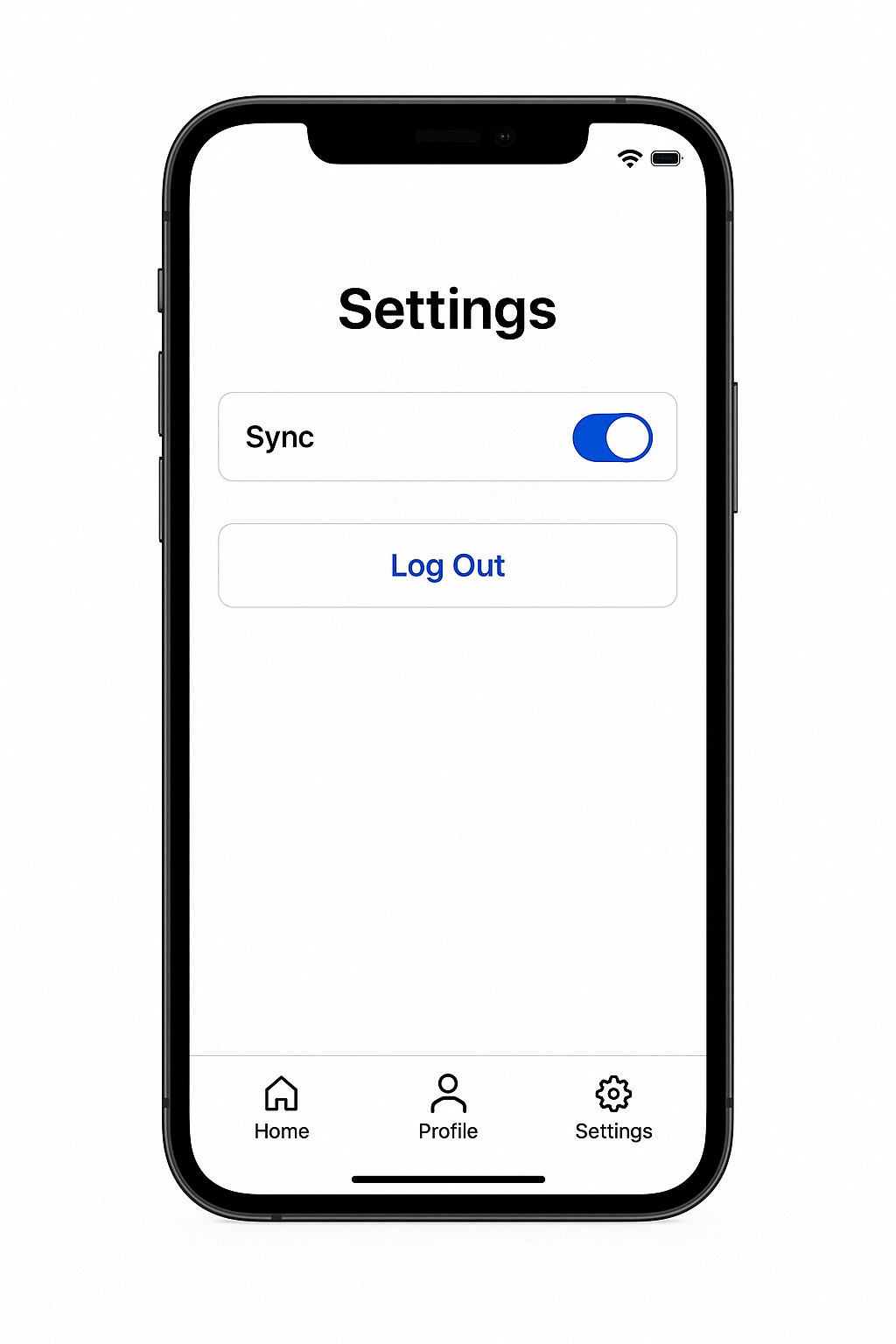


Рисунок 8 - Экран настроек

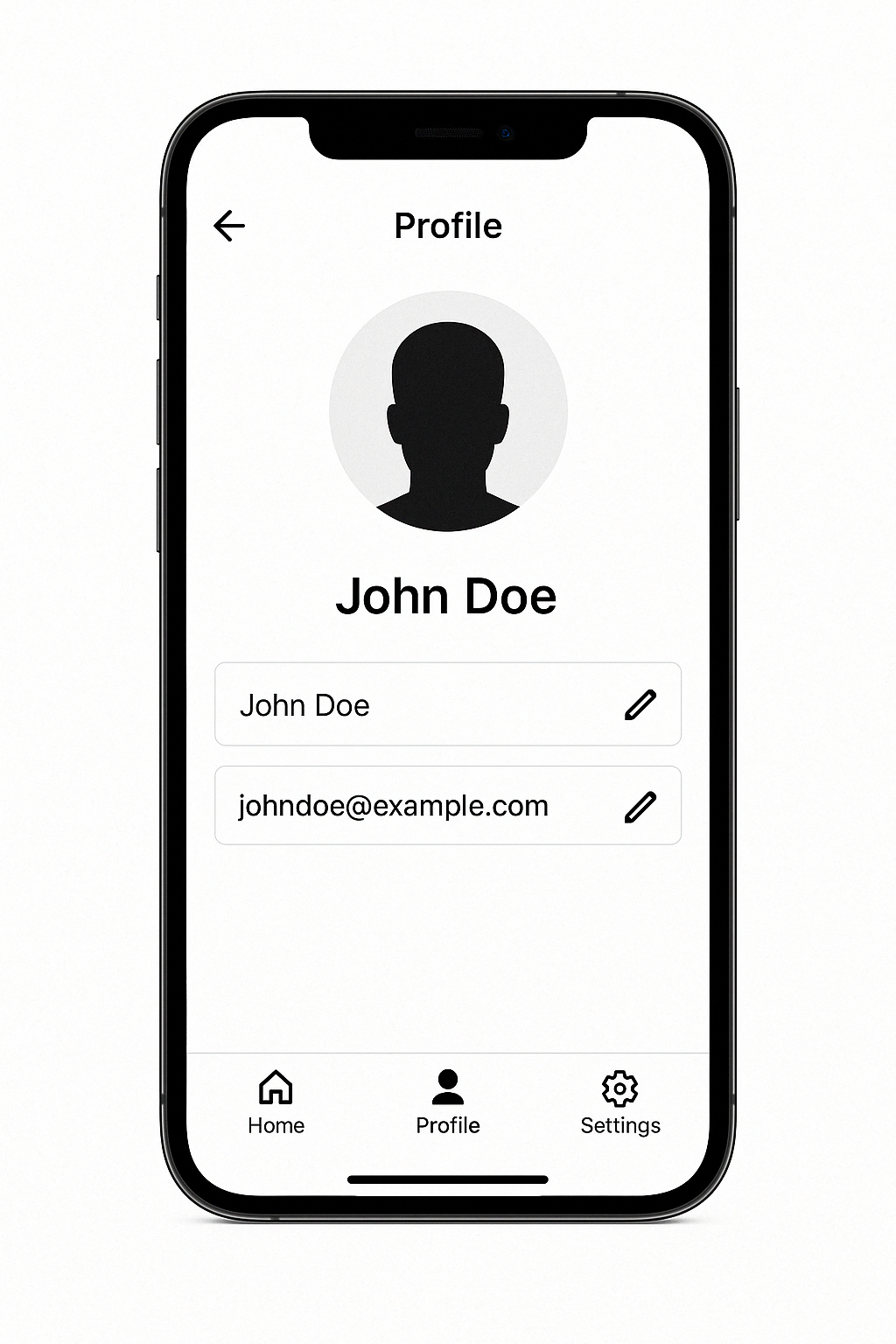


Рисунок 9 - Экран профиля пользователя