

# 1) Цель

Дать пользователю единое место для обзора жизненных сфер (Areas): список сфер (иконка + название + краткое описание + статус) и детальная страница каждой сферы с планами по горизонтам, Next Actions, метриками, историей активности и предупреждениями о стагнации. Данные — Supabase/Postgres с RLS. Интеграция с задачами уже реализована.

---

## 2) Пользовательские сценарии

1. Открыть страницу **Areas** → увидеть список сфер с приоритетами и быстрыми показателями (сколько задач сегодня, последний шаг, предупреждение «не двигалось N дней»).
  2. Нажать на сферу → открыть **страницу сферы**: обзор состояния, планы по горизонтам (день/неделя/месяц/год), Next Actions, активные/завершённые задачи, метрики (health score, активность), журнал событий.
  3. Быстро добавить Next Action/задачу внутри сферы; отметить прогресс; «коснуться сферы» (touch) чтобы сбросить стагнацию.
  4. Получить мягкий нудж при стагнации или падении health score.
  5. В боте: `/areas` (список), `/area <имя>` (короткая карточка), кнопки «Next Action», «Добавить задачу», «Открыть в mini-app».
- 

## 3) UX / UI

### 3.1 Список сфер (List View)

- **Карточка сферы** (row):
  - Иконка (emoji или SVG), название, краткое описание (1 строка, усечение).
  - Бейдж приоритета (1–5) и цветовая полоса статуса (зел/желт/красн).
  - Быстрые показатели: `Сегодня: <активные задачи>` · `Последний шаг: <дата>` · `Стагнация: <N дн>`.
  - Индикатор health score (точка/бар 0–100).
  - Тар → детальная страница.
  - Верхняя панель: поиск/фильтр (по статусу, приоритету), кнопка «+ Сфера» (модал).
  - Сортировка: по приоритету (по умолчанию), затем по стагнации убыв.
  - Состояния: пустой список (обучающая плашка), loading-skeleton, ошибка/ретрай.

### 3.2 Детальная страница сферы (Detail View)

**Header:** иконка, название, краткое описание, приоритет, health score, стагнация. Кнопки: «+ Next Action», «+ Задача», «Touch» (зафиксировать активность), меню (редактировать/архивировать).

**Вкладки:** 1) **Overview** (по умолчанию): - «One Thing» — главный фокус недели (editable). - «Next Actions» — 3 ближайших шага (из задач со статусом todo/in\_progress, horizon ∈ {day, week}). Быстрое завершение/перенос в Сегодня. - **Планы по горизонтам:** карточки «Неделя/Месяц/Год» с короткими целями и %-прогрессом. - «Сигналы» — предупреждения: стагнация > N дн, нет Next Action, слишком много in\_progress, просроченные. 2) **Tasks:** список задач сферы с фильтрами (статус, горизонт), drag-and-drop (локальный порядок в сфере), быстрые статусы. Виджет «Сегодня» вверху. 3) **Metrics:** health score (формула ниже), активность по дням (bar-sparkline), «время с последнего шага», «выполнено за 7/30», «удельная концентрация» (done/(todo+in\_progress)). 4) **Timeline:** лента действий (создания, завершения, комментарии, touch), возможность добавить заметку. 5) **Notes & Files** (лайт): список заметок/файлов (Vault-линки), кнопка «/pack <сфера>» (собрать PDF-пакет в боте).

Микровзаимодействия: свайп по Next Action → done; долгий тап → изменить горизонт/приоритет.

## 4) Модель данных (Supabase)

Примечание: в текущих задачах `task.area` — строка. Для связности вводим справочник `area` и миграцию: `task.area_id` (nullable → затем обязательный), триггер на синхронизацию.

```
create table if not exists area (
  id uuid primary key default gen_random_uuid(),
  user_id uuid not null references app_user(id) on delete cascade,
  name text not null,
  slug text generated always as (regexp_replace(lower(name), '^[a-z0-9]+', '-', 'g')) stored,
  icon text, -- emoji или ключ SVG
  short_desc text,
  priority smallint not null default 3 check (priority between 1 and 5),
  health_score smallint not null default 70 check (health_score between 0 and 100),
  last_activity_ts timestamptz,
  warning_flag boolean not null default false,
  archived_at timestamptz,
  created_at timestamptz not null default now(),
  updated_at timestamptz not null default now(),
  unique(user_id, name)
);

alter table task add column if not exists area_id uuid references area(id) on delete set null;
-- (временный) sync-триггер: если пришёл task.area (текст) и нет area_id — попытаться матчить по имени

create table if not exists area_goal (
```

```

id uuid primary key default gen_random_uuid(),
area_id uuid not null references area(id) on delete cascade,
user_id uuid not null references app_user(id) on delete cascade,
horizon text not null check (horizon in ('week', 'month', 'year')),
title text not null,
progress numeric not null default 0 check (progress between 0 and 1),
due_date date,
created_at timestamptz default now(),
unique(area_id, horizon)
);

create table if not exists area_event (
  id uuid primary key default gen_random_uuid(),
  area_id uuid not null references area(id) on delete cascade,
  user_id uuid not null references app_user(id) on delete cascade,
  ts timestamptz not null default now(),
  type text not null check (type in
('touch', 'note', 'status_change', 'goal_update')),
  payload jsonb
);

-- Материализованное представление для быстрой ленты и списка
create materialized view if not exists mv_area_rollup as
select a.id as area_id, a.user_id,
       coalesce(a.health_score,70) as health_score,
       a.priority,
       a.last_activity_ts,
       (select count(*) from task_occurrence o join task t on t.id=o.task_id
where t.area_id=a.id and o.day=today_local(a.user_id) and o.status!='done') as
tasks_today,
       (select max(o.created_at) from task_occurrence o join task t on
t.id=o.task_id where t.area_id=a.id) as last_occurrence_ts
from area a;

```

RLS:

```

alter table area enable row level security;
create policy area_owner on area for all using (auth.uid()=user_id) with check
(auth.uid()=user_id);

alter table area_goal enable row level security;
create policy area_goal_owner on area_goal for all using (auth.uid()=user_id)
with check (auth.uid()=user_id);

alter table area_event enable row level security;

```

```
create policy area_event_owner on area_event for all using (auth.uid()=user_id)
with check (auth.uid()=user_id);
```

## 5) Функции/Edge Functions

### 5.1 RPC: создать/обновить сферу

```
create or replace function rpc_area_upsert(p_user uuid, p_id uuid, p_name text,
p_icon text, p_short_desc text, p_priority smallint)
returns uuid as $$
begin
    if p_id is null then
        insert into area(user_id,name,icon,short_desc,priority)
values(p_user,p_name,p_icon,p_short_desc,p_priority) returning id into p_id;
    else
        update area set name=p_name, icon=p_icon, short_desc=p_short_desc,
priority=p_priority, updated_at=now() where id=p_id and user_id=p_user;
    end if;
    return p_id;
end; $$ language plpgsql;
```

### 5.2 RPC: touch сферы (сброс стагнации, событие)

```
create or replace function rpc_area_touch(p_user uuid, p_area uuid, p_note text
default null)
returns void as $$
begin
    update area set last_activity_ts=now(), warning_flag=false where id=p_area
and user_id=p_user;
    insert into area_event(area_id,user_id,type,payload)
values(p_area,p_user,'touch', jsonb_build_object('note',p_note));
end; $$ language plpgsql;
```

### 5.3 Edge Function: nightly rollout `area_rollups`

- Пересчёт `health_score` по формуле (см. ниже),
- Проставление `warning_flag` если `now() - coalesce(last_activity_ts, created_at) > N дней` или `tasks_today=0` при наличии активных целей,
- Refresh `mv_area_rollup`.

## 5.4 Health Score (v1)

```
HS = clamp( 40% * activity_7d + 40% * completion_rate_7d + 20% * focus_ratio ) -  
activity_7d = min(1, events_7d/5) -  
completion_rate_7d = done_7d / max(1, (todo_7d + inprog_7d + done_7d)) -  
focus_ratio = min(1, next_actions_active / 3) Выводить расшифровку «как повысить +10».
```

---

## 6) Mini-app реализация

- **Маршруты:** `/areas` (список), `/areas/:id` (деталь). State-менеджер — React Query; optimistic updates на touch/Next Action/изменение целей.
  - **Компоненты:** AreaCard, AreaHeader, NextActionsList, HorizonsCard (Week/Month/Year), MetricsWidget, Timeline, NotesList.
  - **Иконки:** emoji по умолчанию; позже — SVG-пак.
  - **Цветовые токены:** `--area-ok`, `--area-warn`, `--area-bad` для статусов; градация по health score.
  - **Пустые состояния:** «Нет Next Action» → CTA «Добавить шаг».
- 

## 7) Интеграция с задачами

- В модале «+ Задача» внутри сферы подставлять `area_id`.
  - Фильтрация списка Tasks по `area_id`.
  - На Overview показывать задачи горизонта `day/week` как Next Actions (до 3 шт.).
  - При выполнении Next Action → запись в `area_event` (type='status\_change').
- 

## 8) Бот

- `/areas` — список сфер (иконка, приоритет, стагнация, tasks\_today). Инлайн-кнопки: «Открыть», «Next Action», «Добавить задачу».
  - `/area add <имя> [#описание] [!приоритет1-5]`.
  - `/area <имя|id>` — карточка сферы (HS, стагнация, 3 next actions) + кнопка «Открыть в mini-app».
  - Кнопка «Touch» в карточке — вызывает `rpc_area_touch`.
- 

## 9) Нефункциональные требования

- Рендер списка  $\leq 50$  мс при 20 сферах; детальная страница  $\leq 150$  мс.

- Все изменения отражаются в реальном времени через Supabase Realtime (подписка на `task_occurrence`, `area`, `area_goal`, `area_event`).
  - Политики RLS на всех таблицах; бэкэнд бота — через сервис-роль.
- 

## 10) Acceptance Criteria

1) Список сфер показывает корректные счётчики (`tasks_today`, стагнация по дате последней активности) и `health score`. 2) Детальная страница отображает «One Thing», Next Actions (до 3), планы по Week/Month/Year, метрики, таймлайн. 3) Добавление Next Action/задачи из детальной страницы создаёт задачу с `area_id` и отражается в Today. 4) Нажатие «Touch» обновляет `last_activity_ts`, снимает предупреждение и пишет событие в `area_event`. 5) Ночной `rollup` корректно обновляет `health score` и `warning_flag`. 6) Бот корректно отдаёт `/areas` и `/area <имя>`, работает Touch.

---

## 11) Тест-кейсы (Given-When-Then)

- **Стагнация:** Given `last_activity` 10 дней назад, When открываю список, Then вижу бейдж «10 дн» и `warning`.
  - **Touch:** Given `warning`, When нажимаю Touch, Then `warning` исчезает, в таймлайне событие.
  - **Next Action:** Given сфера без Next Action, When добавляю задачу горизонта week, Then появляется в блоке Next Actions.
  - **Health Score:** Given 5 событий за 7 дней и высокая доля `done`, Then `HS` > 80, формируется подсказка «как улучшить».
  - **Бот:** Given `/area <имя>`, When жму «Next Action», Then открывается mini-апп на вкладке Tasks сферы.
- 

## 12) Риски и меры

- **Несоответствие строкового** `task.area` и `area_id` → миграционный скрипт маппинга по `name`; временный триггер-синк; по завершении поле `task.area` объявить `deprecated`.
  - **Перегруз Overview** → ограничить Next Actions до 3; остальное во вкладке Tasks.
  - **Latency Realtime** → `optimistic updates` + фоновая синхронизация.
- 

## 13) Следующие шаги

1) Миграции DDL (`area`, `area_goal`, `area_event`, `mv_rollup`, RLS). 2) Edge `area_rollups` + cron. 3) API мини-аппа (React Query) и компоненты UI. 4) Команды бота `/areas`, `/area`, Touch. 5) QA по AC + полевые тесты с 5–10 сферами и 100+ задачами.