

Orgive: Product Requirement Document (PRD) 1

Feature: Community Page (Stories | Reels | Articles)

Version: v0.1 – June 9 2025

1. Purpose & Goals

Goal	Why it matters
Give users a safe, engaging place to share health-related experiences (organ donation journeys, diet tips, sleep hacks, etc.).	Drives daily active usage (DAU), builds community trust, surfaces organic content that models healthy behaviors.
Provide personalized, scrollable feeds curated by topic & user preference.	Keeps users returning; complements the app’s AI insights with peer-generated content.
Ensure HIPAA-aware privacy, easy reporting, and medical accuracy safeguards.	Maintains compliance, avoids misinformation, protects brand reputation.

Success Metrics (90 days post-launch)

- ≥ 25 % of monthly active users post or react at least once/week.
- Session length on Community Page ≥ 3 min (average).
- < 1 % content removal rate due to policy violations.

2. Personas & Core User Stories

Swati (28, kidney donor): “I want to post a reel of my recovery, so others feel confident donating.”

Raj (45, father of diabetic teen): “I want to bookmark credible diet articles and discuss them with my son.”

#	User Story	Priority
US-1	As a user, I can create a story (15-60 sec video or image + caption) and tag topics.	P0
US-2	As a user, I can swipe vertically through a personalized feed of community content filtered by topics I follow.	P0

US-3	As a user, I can like, comment, share, or save a post.	P0
US-4	As a user, I can report or flag misinformation or offensive content.	P0
US-5	As a user, I can follow other users & see their posts in my feed.	P1
US-6	As a user, I can view “Trending” and “Staff-Curated” tabs for high-quality health content.	P1

3. Functional Requirements

Content Types

- Stories: ephemeral (24 h), full-screen, tappable progress bar.
- Reels: ≤ 60 s vertical video, looping, audio controls.
- Articles: Markdown-formatted text, hero image, 600-word limit.

Feed Logic (MVP)

1. Fetch top 200 posts by topic tag \rightarrow order by posted_at DESC.
2. Re-rank on device with simple collaborative-filter score: likes + $(2 \times \text{comments})$ + follow_score.
3. Future: plug in RAG-based AI recommender (*out of scope*).

CRUD APIs (REST over HTTPS; prefix /v1/community)

- POST /posts – upload media (pre-signed S3 URL), captions, tags.
- GET /feed?topics=&page= – paginated JSON.
- PATCH /posts/{id} – edit caption/tags ≤ 15 min after posting.
- POST /reports – body {post_id, reason}.

Moderation Pipeline (MVP)

- Client-side: nudity & violence ML model (TensorFlow-Lite).
- Server-side: queue to moderation table; admin dashboard review ≤ 12 h.
- Auto-remove if flagged by ≥ 5 users.

4. Non-Functional Requirements

Category	Specification
Performance	Initial feed load ≤ 1.5 s on 4G; lazy-load images & videos.
Scalability	Support 50 k DAU with < 70 ms p95 API latency.
Security & Compliance	OAuth2 (Google/Apple), JWT, encrypted S3 buckets; store only non-PHI user-generated content; follow HIPAA guidelines for any incidental health mentions.
Accessibility	WCAG 2.1 AA: captions for videos, alt-text for images, VoiceOver labels.
Analytics	Firebase Analytics events: post_create, feed_scroll, post_like, report_submit.

5. UI / UX Guidelines

- Visual Style – Flexible palette (suggestions: indigo #1D4ED8, teal #14B8A6, warm gray backgrounds), rounded corners (12 px), soft drop-shadows.
- Navigation – bottom tab “Community”. Stories carousel at top; infinite vertical feed below.
- Micro-Interactions – heart animation on like, haptic feedback on save, pull-to-refresh.
- Empty States – friendly illustration + “Follow topics to see posts”.
- Content Composer – three-step wizard (Media \rightarrow Caption & Tags \rightarrow Preview).
- Design Tools – create high-fidelity mockups for approval before coding.

6. Technical Stack & Architecture

Layer	Choice (Flexible)	Rationale
Front-end	Flutter 3.22 + Riverpod state mgmt.	Matches existing app; fast cross-platform.
Media Processing	Flutter video_player, image_picker, flutter_ffmpeg	Compresses and transcodes media under 10 MB.
Backend	Supabase (Postgres, Row Level Security)	Already used by core app; real-time subscriptions for comments.
Storage	Supabase Storage (S3-compatible)	Secure, CDN edge caching.
Functions	Supabase Edge Functions (TypeScript)	Feed ranking & moderation enqueue; low-latency.
CI/CD	GitHub Actions → Test → Supabase → TestFlight	Fast iteration.
Testing	Flutter integration_test, Mockito	80 % coverage target.

7. Privacy, Safety & Compliance Checklist

- No PHI stored in public posts; encourage users to avoid personally identifying medical details.
- Age-gating: users < 13 denied access (COPPA).
- Terms of Use & Community Guidelines link at composer screen.
- GDPR: add “Delete my posts” endpoint (hard delete within 30 days).

8. Milestones & Tentative Timeline (8 weeks)

Week	Deliverable	Notes
1 (Jun 10–14)	Kick-off; clarify scope; review PRD & tech stack.	
2–3	Design phase – wireframes & Figma prototypes; review & sign-off.	
4	Backend: create DB tables, storage buckets, auth rules; stub APIs.	
5	Front-end: implement composer UI; upload flow w/ progress bar.	
6	Feed view: infinite scroll, pull-to-refresh, basic ranking.	
7	Engagement features (likes, comments, save, follow). Basic moderation queue.	
8 (Jul 29)	QA & bug bash; lighthouse audit; prepare demo; merge to main branch.	

9. Action Items

- Read: Supabase docs on Row Level Security & Flutter real-time.
- Set up: local dev env (Flutter 3.22, Android Studio, iOS simulator).
- Fork: orgive/community-page repo; create feature/community-feed branch.
- Bi-weekly Updates: 5-slide sprint review every other Saturday (progress, blockers, next steps).
- Demo Day: record 2-min walkthrough by Week 8.

10. Risks & Mitigations

Risk	Likelihood	Impact	Mitigation
Misinformation spread	Med	High	Strict moderation + authoritative health links badge
Video upload size bloat	High	Med	Transcode to 720p; limit duration & bitrate
Bandwidth (conflicts)	Med	Med	Early timeline buffer; stand-up sync 3×/wk

11. Approval

Product and Design Leads: Avik, Nitin, Mohit

Tech Lead: TBD