## **CelebFitLife Functional Requirements Document(FRD)**

**Project:** CelebFitLife

**Author:** Aaron Hazzard | Senior Software Engineer

**Date:** January 31st, 2025

**Version:** v1.0

## **Table Of Contents**

[1. Introduction](#_r9sz0yg9mr5b)

[2. Key Terms / Glossary](#_bc45v0fz1mx1)

[3. Purpose & Scope](#_z3ymxy9ziaah)

[4. User Personas](#_1kq94sjgae0s)

[5. Requirements Overview](#_jdxzowd6y4g9)

[5.1. User (Viewer) Journey](#_ke4nwqf18cg5)

[5.2. Streamer Journey](#_9s5mho17xc95)

[6. Functional Requirements](#_9znf973j75l)

[6.1. Landing & Subscription](#_iza79ga5u59j)

[6.2. Registration & Authentication](#_bh5th9mcv7ry)

[6.3. Payment & Subscription Management](#_68zs7ksz9yfm)

[6.4. Streaming & Live Interaction](#_ceh3tcwxpvwz)

[6.5. Moderation & Access Control](#_yct5n6t5kyqo)

[6.6. Notifications & Scheduling](#_l5vl7rupv25v)

[7. Non-Functional Requirements (High-Level)](#_pof95gz2gx9j)

[8. Assumptions & Dependencies](#_xt3opzqyj93x)

[9. Open Questions](#_6xxtg39zt187)

[Conclusion](#_x8ty2mg65b3g)

[Next Steps](#_naqit4xypluq)

# 

# 

# 

# 

# Introduction

**CelebFitLife** is a live-streaming fitness platform that connects celebrities and fitness instructors (Streamers) with fans (Users) in real time. The goal is to create a fun, engaging fitness experience by allowing fans to work virtually alongside their favorite celebrities or instructors.

# Key Terms / Glossary

* **FRD (Functional Requirements Document)**

A document outlining the functions that a system or component must be able to perform, detailing business and technical requirements.

* **Firebase (**Already Setup**)**

A Backend-as-a-Service (BaaS) platform by Google that offers authentication, real-time database, cloud storage, hosting, and other services for building apps quickly.

* **Email.js(***Already Setup***)**

A cloud-based email delivery platform used to send and manage large volumes of email reliably (e.g., transactional emails, notifications).

* **Stripe (***Already Setup***)**

An online payment processing platform enabling subscription management, billing, and secure transactions.

* **Twilio (**Already Setup**)**

A cloud communications platform providing APIs for real-time communications—voice, video, messaging—often used for live streaming or SMS.

* **Zustand**  (To-Do)

A small, fast, and flexible state-management library for React applications, allowing you to store and manage global state outside of components.

# Purpose & Scope

* **Purpose**: Define the functional requirements and core user flows for both **Users** and **Streamers**, ensuring that the development team has a clear, shared understanding of system behavior.
* **Scope**: The FRD covers core features such as user registration, subscriptions, live streaming, chat moderation, content restrictions, and scheduling. It also touches on the broader ecosystem integrations with Firebase (auth, data), Twilio (streaming), Stripe (payments), and more.

# User Personas

1. **End User / Viewer**
   * Typical fitness enthusiast or fan of the celebrity.
   * Wants an engaging, interactive workout experience with minimal friction.
2. **Streamer (Celebrity / Fitness Instructor)**
   * Provides live workout content.
   * Needs to manage scheduling, interact with viewers, handle moderation, and monetize content.

# 5. Requirements Overview

## 5.1. User (Viewer) Journey

1. **Landing Page** – The user sees basic site info, pricing, and streaming previews.
2. **Sign Up** – The user registers with personal details and is offered a *1 workout free trial* so that they don’t necessarily have to workout the same day*.*
3. **Subscription Confirmation** – The user selects a package (1 streamer, 3 streamers, or unlimited), sets up payment details (Stripe), and is billed after the free trial.
4. **Stream Selection** – The user chooses a streamer to follow. If not subscribed, they get a 1-minute preview with a countdown timer.
5. **Live Stream Interaction** – Subscribed users can chat, participate in polls, and see real-time updates. Non-subscribed or free viewers see limited content.
6. **Access Restrictions** – If a user is not subscribed, after 1 minute of viewing, the stream is blurred, audio is cut, and they are redirected.

## 5.2. Streamer Journey

1. **Streamer Login** – Using a unique sub-domain email (e.g., dwan.streamer@celebfit.com).
2. **Dashboard** – Displays live view metrics, scheduling tools, chat moderation tools, and notifications for subscribed users.
3. **Scheduling** – Set the date/time for upcoming live sessions. Subscribed users receive a notification (email, possibly in-app).
4. **Live Streaming** – Launch a Twilio live stream and see real-time chat (with moderation controls, polls, upvotes).
5. **Subscriber Management** – Ability to see who is subscribed, possibly export or message subscribers.
6. **Moderation & Engagement** – Streamers or assigned moderators can ban/mute viewers, create polls, highlight comments, etc.

# 6. Functional Requirements

## 6.1. Landing & Subscription

* **Landing Page Display**
  + **FRD-L1**: The landing page shall display basic service information, including a high-level description of CelebFitLife, the streaming approach, and highlights of featured streamers.
  + **FRD-L2**: The landing page shall present 3 subscription tiers (e.g., 1-streamer, 3-streamers, unlimited) with corresponding prices (to be determined).
  + **FRD-L3**: A CTA button labeled “Join Workout” shall direct users to the Streaming page.
* **Free Trial**
  + **FRD-L4**: Upon registration, users receive a 1 workout free trial which grants access to 1 full stream session of a single streamer.

## 6.2. Registration & Authentication

* **Sign-Up Flow**
  + **FRD-A1**: Users shall enter their **Username, Email, Password, Phone Number, Country & City, and Age** during registration.
  + **FRD-A2**: Users must accept the Terms & Conditions before proceeding.
  + **FRD-A3**: If using Firebase Auth, the system shall store user credentials securely.
  + **FRD-A4**: Upon registration, a user must confirm their email address (Firebase built-in or email link).
* **Login Methods**
  + **FRD-A5**: Users shall be able to log in via **email + password**, or **social logins** (Google, Facebook).
  + **FRD-A6**: An option to log in via **email link** shall be provided (Firebase’s email link auth).
* **Twilio Integration** (Under the Hood)
  + **FRD-A7**: For streaming access, the system shall verify the user’s subscription status before providing Twilio streaming tokens.

## 6.3. Payment & Subscription Management

* **Payment Integration (Stripe)**
  + **FRD-P1**: Users shall enter credit card details via a secure Stripe form after registration for the free trial.
  + **FRD-P2**: The system shall automatically **bill** the user 1 day (next day EOD) after the 1-workout free trial, according to the selected subscription tier.
  + **FRD-P3**: Users can **cancel** or **upgrade/downgrade** their subscription(profile page).
  + **FRD-P4**: The system shall handle recurring payments monthly (or as determined by the business model).
* **Subscription Packages**
  + **FRD-P5**: Package #1: 1 streamer subscription.
  + **FRD-P6**: Package #2: 3 streamers subscription.
  + **FRD-P7**: Package #3: Unlimited streamers subscription.
* **Subscription Tracking**
  + **FRD-P8**: The system shall track how many streamers a user has subscribed to and prevent subscriptions to more streamers than allowed by their plan.
  + **FRD-P9**: On the streaming page, users who are not subscribed to a chosen streamer should see a prompt to subscribe or use the 1-minute preview.

## 6.4. Streaming & Live Interaction

* **Streaming Page Display**
  + **FRD-S1**: If **not logged in**, a user sees a grid of streamer thumbnails with categories and a search bar.
  + **FRD-S2**: If **logged in**, a user sees the same grid plus a **slider** (like Twitch) of recommended or subscribed streamers.
* **Stream Access**
  + **FRD-S3**: A non-subscribed user can watch a streamer’s live session for **1 minute** (once per month per streamer).
  + **FRD-S4**: A **countdown timer** (bottom-left corner) shows the remaining free preview time.
  + **FRD-S5**: At 5 seconds remaining, a modal appears notifying the user that their free preview is ending.
  + **FRD-S6**: After the timer hits 0, the background is blurred, the audio is cut, and the modal blocks further access (with a CTA to subscribe or close).
* **Device/IP Tracking**
  + **FRD-S7**: The system shall track the user’s IP (and possibly other device/browser fingerprints) to ensure they cannot repeatedly gain free access via VPN or incognito.
  + **FRD-S8**: Restrict free previews to once per streamer per month, for both non-logged-in and logged-in users who aren’t subscribed to that streamer.
* **Chat, Polls, Reactions**
  + **FRD-S9**: Logged-in users can join the chat. Chat messages appear in real time.
  + **FRD-S10**: Streamer or mod can create **polls**; users can vote once. Results appear after the poll ends.
  + **FRD-S11**: Reactions (e.g., emojis, quick “upvote”) are displayed in real-time in the chat window or overlay.

## 6.5. Moderation & Access Control

* **Moderation Roles**
  + **FRD-M1**: Each streamer can assign **Moderators**. Moderators can **mute** or **ban** users in the chat.
  + **FRD-M2**: A streamer can remove or add moderators from the Dashboard.
* **Ban & Mute Functionalities**
  + **FRD-M3**: If a user is banned, they cannot send chat messages in that stream or any future streams by that streamer.
  + **FRD-M4**: Mutes are temporary. The user’s chat messages are hidden for the rest of the workout.
* **Reporting**
  + **FRD-M5**: (Optional initial feature) Users or mods can report malicious chat activity. This can be escalated to an admin interface in the future.

## 6.6. Notifications & Scheduling

* **Streamer Scheduling**
  + **FRD-N1**: Streamers can set specific dates/times for upcoming live sessions from the dashboard.
  + **FRD-N2**: The system shall display upcoming streams in the user’s subscribed streamer list or slider.
* **Email Notifications**
  + **FRD-N3**: Subscribed users receive an **email notification** (via Firebase or Email.js) when their chosen streamer goes live or schedules a new session.
  + **FRD-N4**: Streamers can also send push notifications or email announcements to subscribers (e.g., “Workout delayed by 30 minutes”).

## 

## 

## 

## 7. Non-Functional Requirements (High-Level)

1. **Scalability**: Must handle up to 1,000 users per celebrity in the first year, with more expansions in year two.
2. **Reliability**: High uptime for live streaming. Twilio integration must handle concurrent streams and chat sessions.
3. **Security**: Protect personal and payment data (Stripe, JWT tokens in Firebase Auth).
4. **Performance**: Quick streaming load times; minimal latency for chat.
5. **Usability**: Clear UI/UX so users can easily view, subscribe, and watch streams without confusion.

## 8. Assumptions & Dependencies

1. **Twilio** supports real-time video streaming, chat, and possibly the creation of sub-rooms for moderation.
2. **Firebase** handles:
   * Authentication (email/password, social logins, email link)
   * Database & user data storage (FireStore, Real-Time DB)
   * Possibly hosting and serverless functions (if needed)
3. **Stripe** handles recurring billing, subscription upgrades/downgrades, and cancellation flows.
4. **Tailwind** is used for rapid UI development.
5. **Zustand** for state management in the Next.js front end.

## 

## 9. Open Questions

1. **Moderator Access** – Should moderators have global privileges or be restricted to each streamer’s session? (Likely restricted to each streamer’s session.)
2. **Multiple Streamers** – If a user has multiple subscriptions (3 or unlimited), how do they switch streams mid-session without paying extra? (Probably free to switch among subscribed channels.)
3. **Free Trial Limitations** – Should the trial allow the user to watch 1 full session from a single streamer or any streamer once?
4. **IP & Device Tracking** – Are we implementing a custom device fingerprint approach, or is IP-based tracking sufficient for MVP?
5. **Twilio Features** – Do we rely on Twilio’s built-in chat? Or integrate a separate chat solution (like Firebase’s real-time DB for chat messages) and just use Twilio for streaming?

# Conclusion

The above **Functional Requirements Document** outlines the essential features and flows for **CelebFitLife**. The goal is to provide a **live-streaming fitness platform** that leverages **Next.js**, **Firebase**, **Twilio**, **Stripe**, and **Zustand** for smooth user experience and robust streamer functionality. Future expansions may include additional monetization, more advanced recommendation algorithms, and deeper social features like follower/friend systems.

### **Next Steps**

* **Review & Sign** on these requirements by key stakeholders (business, engineering, product).
* **Create Technical Architecture** detailing the data model, service integrations, serverless functions, etc.
* **Establish a Development Roadmap** with prioritized milestones (MVP → Beta → Production).