

---

# SOFTWARE REQUIREMENTS SPECIFICATION

for

OuterCircl Ltd.

Version 1.0

Prepared by :

Fracto a W3Blendr Ltd Product

Submitted to : David Lau  
Founder & CEO (OuterCircl™)

**November 22, 2025**

**Author: Harsh Gupta  
Chief Architect  
Fracto a W3Blendr Ltd product**

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Purpose . . . . .	4
1.2	Intended Audience and Reading Suggestions . . . . .	4
1.3	Project Scope . . . . .	4
1.4	Definitions, Acronyms, and Abbreviations . . . . .	6
<b>2</b>	<b>Overall Description</b>	<b>8</b>
2.1	Product Perspective . . . . .	8
2.1.1	System Interfaces . . . . .	8
2.1.2	Memory Constraints . . . . .	9
2.1.3	Operations . . . . .	9
2.1.4	Site Adaptation Requirements . . . . .	11
2.2	User Classes and Characteristics . . . . .	11
2.2.1	Users . . . . .	11
2.2.2	Hosts (User Subclass) . . . . .	13
2.2.3	Administrators . . . . .	14
2.3	Product Functions . . . . .	15
2.4	Operating Environment . . . . .	19
2.4.1	Web Browser Environment . . . . .	19
2.4.2	Operating Systems . . . . .	19
2.4.3	Backend Environment . . . . .	20
2.4.4	Network Requirements . . . . .	20
2.4.5	Device Capabilities . . . . .	21
2.4.6	Security Environment . . . . .	21
2.4.7	Administrative Operating Environment . . . . .	21
2.4.8	Future Mobile Application Environment (Planned) . . . . .	22
2.5	Design . . . . .	22
<b>3</b>	<b>System Features</b>	<b>23</b>
3.1	Functional Requirements . . . . .	23
<b>4</b>	<b>Other Nonfunctional Requirements</b>	<b>27</b>
4.1	Performance Requirements . . . . .	27
4.2	Security Requirements . . . . .	27
4.3	Software Quality Attributes . . . . .	28
4.4	Business Rules . . . . .	29
<b>5</b>	<b>Other Requirements</b>	<b>30</b>

# 1 Introduction

## 1.1 Purpose

This Software Requirements Specification provides a detailed description of all the functional and the non-functional requirements for a new web software OuterCircl™ (henceforth also stated as the **application** or the **app**).

## 1.2 Intended Audience and Reading Suggestions

This SRS is for developers, project managers, users and testers. Further the discussion will provide all the internal, external, functional and also non-functional informations about "OuterCircl™Application".

## 1.3 Project Scope

The **OuterCircl** project aims to deliver a production-ready **Fast-Feedback Product (FFP)** — a web-first social discovery platform for all browsers designed to foster small, daily in-person, interest-based connections among adults in Ireland. The initial rollout will begin in Galway, followed by expansion to other Irish cities.

This phase focuses on the **core technical foundation and essential user experiences** required for a successful market launch.

### In-Scope Deliverables

#### 1. Technical Architecture & Infrastructure

- End-to-end design of web and mobile application architecture.
- Backend services with database, authentication, and secure APIs.
- Cloud hosting setup with data protection and security configurations.
- Future-ready payment architecture supporting easy integration with Stripe or similar gateways.
- Integration-ready APIs (Google Ads, Google Analytics, Notifications).

#### 2. Frontend Development

- Responsive web interface accessible through all major browsers.
- Mobile-responsive frontend optimized for smaller screens.

3. **Activity Creation & Host Tools**
  - Multi-step activity creation and management flow for hosts.
  - Configuration of type, capacity, location, participant criteria, and approvals.
4. **Notifications & Engagement**
  - Email and in-app notification systems with user preferences and settings.
5. **User Authentication & Profile Management**
  - Email/password and social login authentication.
  - Role-based access control, SMS/Email verification, and profile customization.
6. **Onboarding Experience**
  - Guided onboarding flow covering interests, intent selection, and community guidelines.
7. **Activity Discovery & Management**
  - “Discover” feed to browse curated activities with filters for date, category, and distance.
  - RSVP and waitlist management features.
8. **Real-Time Messaging**
  - Group chat and pre-activity messaging for attendees with notification support.
9. **Spontaneous Plans Timeline**
  - “Spontaneous” feed showing short-notice meetups with quick-create and proximity filters.
10. **Post-Activity Experience**
  - Review and feedback mechanism for hosts and participants.
11. **Freemium Model Foundation**
  - User tier management (Free vs Premium) with feature-based access control.
  - In-app upgrade prompts and messaging.
12. **Compliance & Security**
  - GDPR compliance, data encryption, SSL/TLS, and consent management mechanisms.
13. **Testing & Quality Assurance**
  - Performance, security, and user acceptance testing.
14. **Documentation & Handover**
  - Comprehensive SRS and codebase handover with repository access.

## 1.4 Definitions, Acronyms, and Abbreviations

This section defines key terms, acronyms, and abbreviations used throughout this Software Requirements Specification (SRS) document for the **OuterCircel** application.

- **User:** The term “User” refers to any individual using the OuterCircel platform. Users can be categorized as:
  1. **Member:** General users who register to discover, join, and participate in activities.
  2. **Host:** Verified users who can create and manage group activities or spontaneous meetups.
  3. **Admin:** System-level users with access to moderation, compliance, and configuration controls.
- **Activity:** A scheduled or spontaneous event created within the platform, where users can join based on shared interests and availability.
- **Activity Instance:** A unique environment representing one occurrence of an activity. Each instance maintains independent participant lists, chats, and feedback data.
- **Discover Feed:** A curated list of activities and meetups presented to users based on their preferences, location, and availability.
- **Spontaneous Plan:** A real-time, short-notice meetup opportunity (2–24 hour window) posted by users in the “Spontaneous” timeline.
- **Host Tools:** A set of functionalities available to hosts for creating, editing, and managing activities, including capacity limits, approval modes, and activity details.
- **Freemium Model:** A business model where users can access core platform features for free, with optional premium tiers that unlock additional benefits and tools.
- **FFP (Fast-Feedback Product):** The initial production-ready version of OuterCircel focused on rapid deployment and user validation in the Irish market.
- **UAT (User Acceptance Testing):** A formal testing process where end-users validate whether the implemented system meets their requirements and expectations.
- **GDPR (General Data Protection Regulation):** EU legislation governing data protection and privacy, applicable to all users and data handled by the OuterCircel platform.
- **API (Application Programming Interface):** A defined set of rules and protocols that allow different software systems to communicate, used for features like authentication, notifications, and analytics.

- **UI/UX (User Interface / User Experience):** The design and interaction components that determine how users visually engage and interact with the application.
- **SRS (Software Requirements Specification):** The document outlining the complete functional and non-functional requirements for the OuterCircl software project.
- **OuterCircl:** The mobile-first social discovery platform designed to help individuals form genuine, interest-based connections through curated and spontaneous meetups.
- **W3Blendr Ltd.:** The supplier company responsible for the design, development, and delivery of the OuterCircl application under this agreement.
- **URL (Uniform Resource Locator):** The address used to access web resources such as activity links, images, and user profiles.
- **GUI (Graphical User Interface):** The visual part of the application through which users interact using icons, buttons, and menus.

## 2 Overall Description

### 2.1 Product Perspective

OuterCirc1™ is a mobile-first social discovery platform developed to address increasing levels of social isolation and loneliness in Ireland. The system is designed for adults seeking authentic, platonic interactions through structured small-group meetups and spontaneous social activities, providing an alternative to traditional bar-centric social environments.

The platform enables safe, verified, in-person engagement by matching users based on shared interests, geographic proximity, and real-time availability. OuterCirc1™ supports three primary interaction models:

- **Curated small-group activities** (4–8 participants)
- **Spontaneous micro-meetups**
- **Interest-based community clubs**



Figure 2.1: High-level overview of the OuterCirc1™ platform

The diagram above provides a high-level overview of how the system delivers its core purpose. The platform prioritizes an intuitive user experience, enabling users to discover, join, evaluate, and communicate around social activities. Additionally, an administrative layer supports activity moderation and safety management to ensure a secure and trustworthy environment.

#### 2.1.1 System Interfaces

OuterCirc1™ will expose multiple system interfaces, each supporting a distinct set of use cases. The interface architecture shall be modular and extensible, allowing future interfaces to be integrated without impacting existing components.



## User Interface

A graphical user interface (GUI) will serve as the primary interaction layer between OuterCirc1™ and end users. All user-facing functionality will be delivered through responsive mobile-first screens.

## Hardware Interface

The application has no specific hardware interface requirements. It operates entirely on standard mobile device capabilities.

## Software Interface

The system will provide internal and external software interfaces via a set of RESTful APIs. These APIs will be managed through cloud-hosted backend services responsible for processing business logic, handling data transactions, and managing user intents.

## Communication Interface

Communication between the mobile client and backend services will use secure HTTPS protocols. All service interactions will follow REST-based communication patterns to ensure interoperability, scalability, and secure data exchange.

### 2.1.2 Memory Constraints

OuterCirc1™ does not introduce any explicit memory constraints beyond those managed by the underlying mobile operating system and cloud infrastructure. The application relies on standard device memory for UI rendering and local caching, while all heavy processing (matching, recommendations, verification checks, and activity coordination) is performed on cloud-hosted services. No additional memory allocation or optimization requirements exist at the application level.

### 2.1.3 Operations

The following operational flow diagrams illustrate the primary interactions between users and the OuterCirc1™ platform. These flows represent how users create, discover, and participate in social activities, as well as how the system manages verification and group coordination.

- **Activity Creation:** Users initiate a new meetup or group activity by specifying details such as interest category, group size, date, time, and location.
- **Activity Discovery and Joining:** Users browse or receive recommendations for relevant activities, verify attendance, and join curated or spontaneous meetups.

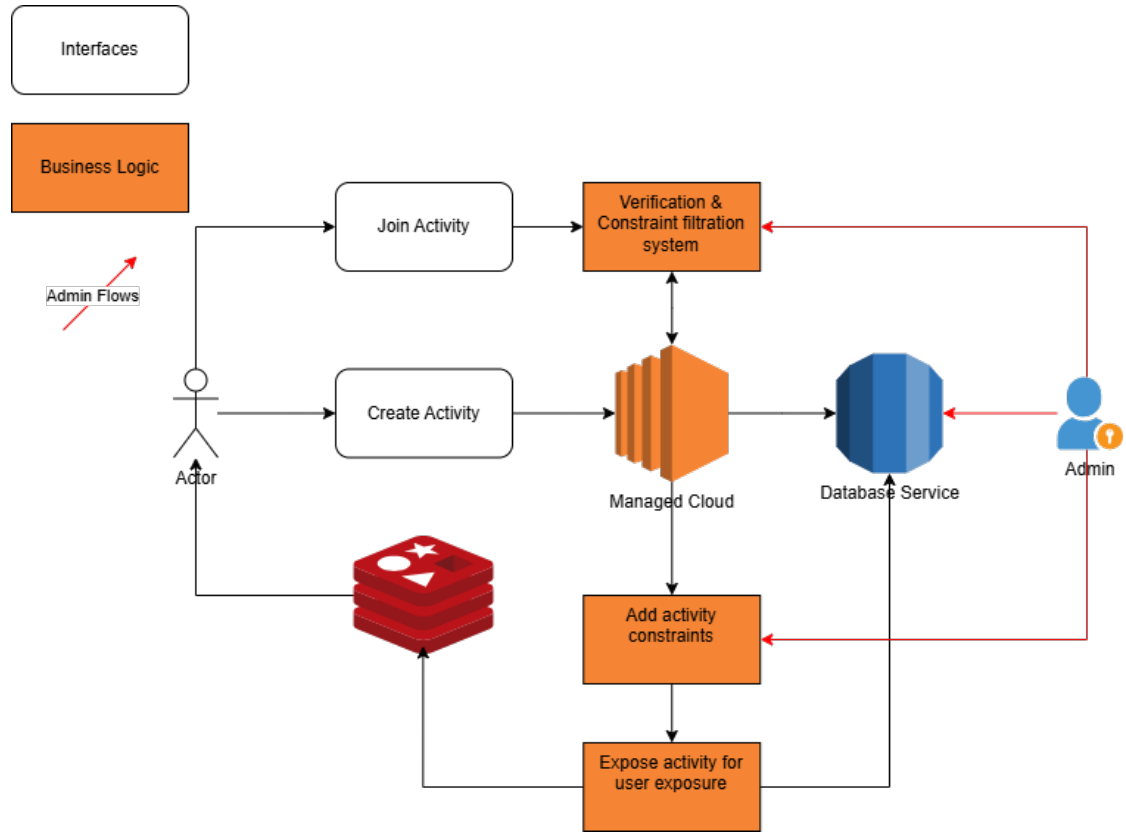


Figure 2.2: High-level operational flow for activity creation and participation

- **Participation and Feedback:** After attending a meetup, users can communicate within the group, exchange feedback, and receive recommendations for future activities.

*Note: User roles (e.g., regular users vs. admin) referenced in the operational workflows are detailed in Section 2.2 (User Classes and Characteristics).*

Once the required user inputs are provided, the system autonomously handles matching, notifications, verification processes, and group coordination without further user intervention.

### User Inputs

The system is driven primarily by user-provided inputs during setup and participation. The following represents the core inputs required for OuterCircle™ to operate:

- **Activity Title:** A short, descriptive name for the meetup or event.
- **Interest Category:** The social theme or activity type (e.g., hiking, board games, coffee meetups).

- **Group Size:** Desired number of participants; supports curated small groups (4–8 users), micro-meetups, and community club sessions.
- **Location:** A specific venue or general geographic area.
- **Schedule:** The date and time of the activity.
- **User Verification Inputs:** Identification and safety verification inputs as required by the system.
- **Availability Status:** User-specified availability used for matching and recommendations.
- **Join Activity:** Confirmation input from users to join an existing activity.
- **Feedback and Ratings:** Post-event evaluations to improve recommendations and trust scores.

Once these inputs are collected, the system automates user matching, notifications, safety checks, and event coordination workflows.

### 2.1.4 Site Adaptation Requirements

OuterCircI™ requires no installation or configuration on the hosting environment beyond standard deployment of the mobile and backend applications. All functionality is delivered as part of the platform’s mobile app and cloud services. No additional hardware, local server setup, or user-side configuration is required. Updates, security patches, and backend maintenance are handled centrally by the operations team.

## 2.2 User Classes and Characteristics

OuterCircI™ supports two primary user classes: **Users** and **Administrators**. Users may also take on a secondary functional role known as a **Host** when creating activities. The characteristics, responsibilities, and interactions of these classes are described below.

### 2.2.1 Users

A **User** represents any registered individual engaging with the OuterCircI™ mobile platform. Users form the core participant base and may join or create social meetups depending on their preferences and engagement style.

#### Characteristics:

- Adults seeking interest-based, platonic group interactions.
- Mobile-first individuals comfortable with smartphone interfaces.

## User

- Registered app user
- Joins activities (datafeed/search)
- Provides feedback/ratings

## Host

- User acting as organiser
- Creates Spontaneous activities
- Creates Scheduled/Recurring activities



user



admin

## Admin

- Moderates activities
- Handles safety & reports
- Oversees platform health



host

inherits

Figure 2.3: High-level User Classes and their characteristics

- Exhibit diverse social interests, availability patterns, and geographic ranges.
- May participate in multiple spontaneous or scheduled activities.
- May optionally undergo identity verification for safety and trust.

### Primary Responsibilities:

- Browse and join activities recommended through the personalized datafeed.
- Interact with other participants via in-app messaging.
- Provide feedback and rate activities after participation.

### Datafeed-Based Recommendations:

Users can join activities through a personalized **datafeed**, which is a proprietary recommendation algorithm developed and owned by OuterCircel™. The datafeed curates activities based on:

- User interests and behavioural patterns
- Geographic proximity
- Availability windows

- Previously attended activities
- Social compatibility metrics
- Accepted friend's activity/ interest data

This feature enables non-host users to continuously discover relevant meetups without manually searching for activities.

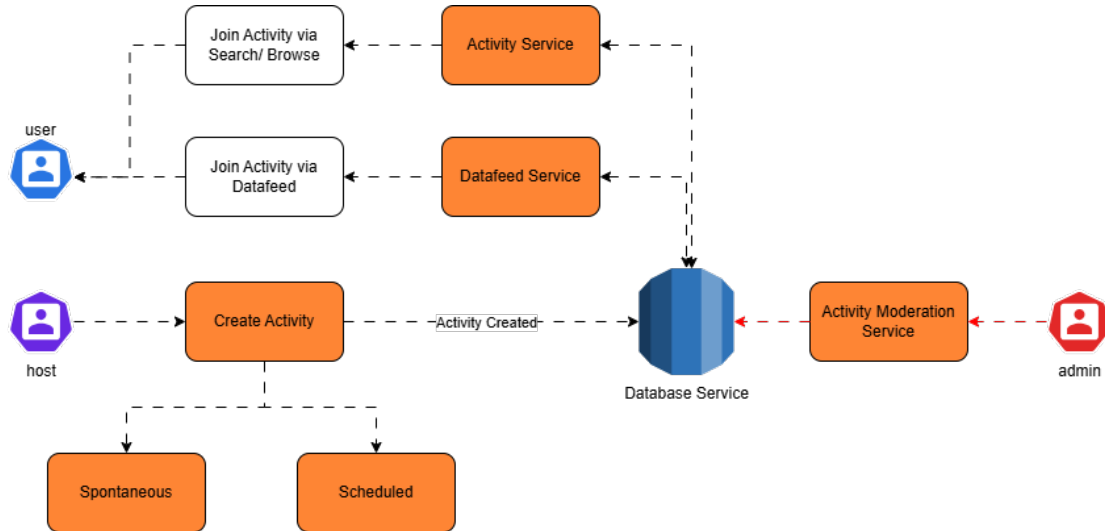


Figure 2.4: High-level Use Cases per User

### 2.2.2 Hosts (User Subclass)

A **Host** is a user who chooses to create and manage social activities within the platform. All Hosts are Users, but not all Users become Hosts. Hosting privileges are available to all verified users.

#### Characteristics:

- Users are comfortable organising group interactions.
- Can manage both spontaneous and scheduled activities.
- Responsible for defining activity parameters and maintaining group etiquette.

#### Primary Responsibilities:

- Create and manage meetup activities.
- Communicate with participants.

- Provide clear activity details (location, schedule, group size, interest type).
- Ensure respectful and safe participation during hosted activities.

Hosts may create the following two types of activities:

1. **Spontaneous Activities** Short-notice meetups initiated for immediate or near-term participation (within minutes or hours). These facilitate quick coordination and real-time socializing.
2. **Scheduled Activities** Structured meetups planned for a future date or recurring over time. These support long-term engagement and allow users to plan ahead.
3. **Own Private/ Public Groups** Hosts as premium user can manage a curated group of friends either privately or publicly where users within that group can see the specific group activities.

### 2.2.3 Administrators

An **Administrator** is responsible for monitoring platform health, ensuring safety, and enforcing community guidelines. Administrators operate primarily through a dedicated oversight dashboard with elevated privileges.

#### Characteristics:

- Trained staff with moderation authority.
- Familiar with compliance, verification, and safety processes.
- Typically work from desktop interfaces rather than the mobile app.

#### Primary Responsibilities:

- Review and moderate user-created activities.
- Handle user reports and safety escalations.
- Remove inappropriate or unsafe content.
- Oversee suspicious behaviour, fraudulent profiles, or abusive conduct.
- Coordinate with technical teams for platform reliability and compliance.

## 2.3 Product Functions

### F1

**User Registration and Authentication Description:**

Handles user onboarding and secure access. Supports registration via email, phone number, or third-party identity providers. Enables optional identity verification for increased trust and to unlock hosting capabilities.

**Dependencies:**

None (entry point function).

**Requirements to Fulfill:**

**F7** (profile setup), **F10** (membership activation)

### F2

**Personalized Datafeed and Activity Discovery Description:**

Generates a personalized feed of recommended activities using OuterCircel's proprietary recommendation algorithm. Operates on user-provided data including:

- Location (GPS or preferred radius)
- Interests and hobbies
- Survey results: personality, comfort zones, social preferences
- Availability windows
- Past activity participation history
- Behavioural signals (ratings, cancellations, social compatibility, friend's activities (attending/ hosting))

The feed displays spontaneous and scheduled activities with one-tap join capabilities.

**Dependencies:**

**F1** (user registration), **F7** (user profile data),  
**F6** (ratings feedback loop).

**Requirements to Fulfill:**

**F3** (participation), **F5** (group communication)

### F3

**Activity Participation Description:**

Allows users to join spontaneous or scheduled activities. Displays full activity details including host info, date/time, and group size. Supports attendance confirmation, cancellation, waiting lists, and in-group communication. Meetup location will be only be available on confirmed attendance

**Dependencies:**

**F2** (Discovery), **F1** (Authentication)

**Requirements to Fulfill:**

**F5** (Messaging), **F6** (Feedback)

### F4

**Activity Hosting (Host Role) Description:**

Enables verified users to act as Hosts and create activities. Supports:

- Spontaneous activities (minutes/hours)
- Scheduled and recurring activities
- Activity metadata: interest type, location, description, schedule

Free-tier hosts can create fixed group-size activities of 4 participants. Premium hosts may create and own:

- flexible group sizes,
- public or private activities,
- recurring sessions.

**Dependencies:**

**F1** (Authentication), **F7** (Profile/Verification), **F10** (Premium Features)

**Requirements to Fulfill:**

**F5** (Group Communication), **F3** (Participation)



## F5

### Group Management and Communication **Description:**

Provides in-app group messaging for participants. Hosts can manage participant lists, share updates, and issue reminders. The system automatically notifies users of time/location changes.

### Dependencies:

**F3** (Participation), **F4** (Hosting)

### Requirements to Fulfill:

**F9** (Notifications)

## F6

### Post-Activity Feedback and Ratings **Description:**

Collects user ratings and feedback after an activity. Feeds evaluation data into trust, safety, and recommendation models. Improves future matching accuracy and datafeed performance.

### Dependencies:

**F3** (Participation)

### Requirements to Fulfill:

**F2** (Datafeed Learning Loop), **F8** (Safety Moderation)

## F7

### User Profile and Preference Management **Description:**

Allows users to update personal information, interests, hobbies, and visibility settings. Handles survey submission for datafeed accuracy. Stores availability windows and distance preferences. Premium users access advanced preference options.

### Dependencies:

**F1** (Authentication)

### Requirements to Fulfill:

**F2** (Datafeed), **F10** (Membership Controls)

## F8

### Administrative Moderation and Safety Management **Description:**

Administrators review, approve, or remove activities. Handle reports, safety escalations, fraud detection, and abusive behaviour. Monitor trust metrics and enforce community guidelines.

### **Dependencies:**

**F1** (User Identifiers), **F6** (Ratings), **F4** (Activity Creation)

### **Requirements to Fulfill:**

**F9** (Alerts), **F3** (Participation Logs)

## F9

### Notification and Alert System **Description:**

Sends notifications for:

- recommended activities,
- activity reminders,
- participant joins/cancellations,
- host updates,
- safety alerts.

Supports push notifications and in-app alerts.

### **Dependencies:**

**F3** (Participation), **F5** (Group Messaging)

### **Requirements to Fulfill:**

None

## F10

### Membership and Subscription Management **Description:**

Handles premium membership features. Supports subscription lifecycle: upgrade, downgrade, renewal, cancellation. Unlocks:

- flexible group sizes,

- public/private activity creation,
- advanced preferences.

**Dependencies:**

**F1** (Authentication)

**Requirements to Fulfill:**

**F4** (Hosting), **F7** (Preferences)

## 2.4 Operating Environment

OuterCirc1™ operates as a browser-based social discovery platform supported by cloud-hosted backend services. The system is designed to run reliably within modern web environments across desktop and mobile browsers. This section outlines the required operating conditions for deployment and user usage.

### 2.4.1 Web Browser Environment

The primary interface for OuterCirc1™ is a responsive web application. The platform supports the following browsers:

- **Desktop Browsers**
  - Google Chrome (latest two versions)
  - Mozilla Firefox (latest two versions)
  - Microsoft Edge (latest two versions)
  - Safari (latest two versions on macOS)
- **Mobile Browsers (Web Access)**
  - Safari on iOS
  - Chrome on Android
  - Chrome on iOS

The application must be fully responsive and maintain usability across various screen sizes and form factors.

### 2.4.2 Operating Systems

As the platform runs in the browser, the following operating systems are supported:

- **Desktop Operating Systems**

- Windows 10 or later
- macOS 11 (Big Sur) or later
- Linux distributions with a modern browser
- **Mobile Operating Systems (for browser access only)**
  - iOS 14 or later
  - Android 9 or later

No installation of native applications is required in the current version.

### 2.4.3 Backend Environment

OuterCirc<sup>TM</sup> relies on a secure and scalable cloud-hosted backend. The backend environment includes:

- REST-based microservices
- Managed SQL or NoSQL database (e.g., PostgreSQL, Firestore, DynamoDB)
- Real-time communication via WebSockets or server-sent events
- Machine learning infrastructure for personalized recommendations
- CDN-backed static resource delivery for global performance

This environment shall support horizontal scaling and automated failover for high availability.

### 2.4.4 Network Requirements

To support real-time activity discovery and group coordination, the user environment must provide:

- Stable network access via:
  - Wi-Fi
  - 4G LTE / 5G (for mobile browser access)
  - Wired broadband
- Recommended minimum bandwidth:
  - 2 Mbps download
  - 1 Mbps upload
- Tolerance to moderate latency and intermittent mobile connectivity

GPS access is recommended for proximity-based discovery features.

### 2.4.5 Device Capabilities

Since the platform executes in a web browser, it may require access to:

- **Geolocation API** – for location-based recommendations
- **Notifications API** – optional browser alerts for activity reminders
- **Local/Session Storage** – for caching session data and preferences
- **Camera Access (Optional)** – for future identity verification flows

Devices with limited browser API permissions may experience partial functionality.

### 2.4.6 Security Environment

The platform must operate within a secure execution context, including:

- HTTPS/TLS 1.2+ enforced for all communications
- Secure cookie handling and CSRF protection
- Strict Content Security Policies (CSP)
- Sanitization of all user-generated content
- Token-based authentication (JWT or equivalent)
- IP rate limiting and threat detection at the API gateway

Administrative interfaces must enforce multi-factor authentication and secure session management.

### 2.4.7 Administrative Operating Environment

Administrators access moderation tools via a secure web interface. Requirements include:

- Modern browser (Chrome, Safari, Firefox, Edge)
- Minimum resolution of  $1280 \times 720$
- Stable internet connectivity
- Authentication using MFA

Administrative access should be limited to trusted, non-public devices.

### 2.4.8 Future Mobile Application Environment (Planned)

While not part of the current operating environment, system architecture shall remain compatible with future native mobile applications. This includes:

- Mobile-ready API design
- Token-based authentication suitable for mobile clients
- Integration support for APNs and FCM push notifications
- Reusable UI component architecture for potential React Native or Flutter implementations

This future-proofing ensures minimal overhead when extending the platform to mobile apps.

## 2.5 Design

The detailed design of the OuterCircle™ platform is considered out of scope for this Software Requirements Specification (SRS). The purpose of this document is to define the functional and nonfunctional requirements of the system, independent of the technical design or implementation architecture.

All design activities—including UI/UX design, system architecture diagrams, database schema design, API structural design, and component-level specifications—shall be provided separately by the development team or by the designated system architect. These artifacts will be delivered as part of the project's design documentation and are not included within this SRS.

## 3 System Features

### 3.1 Functional Requirements

#### F1

Functional Requirements: User Registration and Authentication

**FR1.1** The system shall allow users to register using an email address and password.

**FR1.2** The system shall allow users to authenticate through a secure login form.

**FR1.3** Passwords shall be encrypted and stored using secure hashing algorithms.

**FR1.4** The system shall validate email addresses during sign-up.

**FR1.5** Users shall be able to reset their password via a secure email-based flow.

**FR1.6** The system shall support optional identity verification for users who wish to become Hosts.

**Dependencies:** None. **Requirements to Fulfill:** **F7**, **F10**

#### F2

Functional Requirements: Personalized Datafeed and Activity Discovery

**FR2.1** The system shall collect user data (location, interests, hobbies) to generate recommendations.

**FR2.2** The system shall provide a user onboarding survey with at least 10 key preference questions.

**FR2.3** The system shall generate a personalized activity feed for each user.

**FR2.4** The system shall refresh recommendations dynamically based on real-time availability.

**FR2.5** Users shall be able to filter activities manually (time, location, interest type, friends activities, categories, groups your part of).

**FR2.6** The system shall allow users to join any activity directly from the feed.

**FR2.7** The recommendation engine shall update based on participation history and ratings.

**Dependencies:** **F1**, **F7**, **F6** **Requirements to Fulfill:** **F3**, **F5**

#### F3

Functional Requirements: Activity Participation

**FR3.1** The system shall allow users to join spontaneous and scheduled activities.

**FR3.2** The system shall show full activity details before joining (time, place) Meetup spot is shown upon successful participation.

**FR3.3** The system shall allow users to cancel participation with a single action.

**FR3.4** The system shall notify users upon successful join or cancellation.

**FR3.5** If participating a private activity, an approval will be required from the host

**FR3.6** The system shall allow users to join a waiting list if an activity is full.

**FR3.7** The system shall restrict joining private activities unless invited.

**Dependencies:** **F1**, **F2** **Requirements to Fulfill:** **F5**, **F6**

#### F4

Functional Requirements: Activity Hosting

**FR4.1** Verified users shall be able to create spontaneous activities.

**FR4.2** Verified users shall be able to create scheduled and recurring activities.

**FR4.3** Free-tier users shall be restricted to a fixed group size of 4 participants.

**FR4.4** Premium users shall be able to set a custom group size.

**FR4.5** Premium users shall be able to create as well as "own" or create a public or private group/curated list - with activities that can be hosted within that group - cap group size of 15

**FR4.6** The system shall allow hosts to edit activity details until the activity starts.

**FR4.7** The system shall allow hosts to remove participants if needed.

**FR4.8** Free-tier users shall be restricted to only 2 hosting per month.

**Dependencies:** **F1**, **F7**, **F10** **Requirements to Fulfill:** **F3**, **F5**

#### F5

Functional Requirements: Group Management and Communication

**FR5.1** The system shall provide group messaging for each activity.

**FR5.2** Users shall receive updates for changes in time, location, or host messages.

**FR5.3** Hosts shall be able to send pinned announcements.

**FR5.4** Users shall be able to report inappropriate behaviour within the group.

**FR5.5** The system shall support real-time message delivery.

**FR5.6** The system shall generate an automated pre and post activity messages.

**FR5.7** The system shall generate an automated survey post the activity.

**Dependencies:** **F3**, **F4** **Requirements to Fulfill:** **F9**

#### F6

Functional Requirements: Post-Activity Feedback

**FR6.1** The system shall prompt users for feedback after an activity ends.

**FR6.2** The system shall allow users to rate activities on a 1–5 scale.



**FR6.3** The system shall allow users to provide optional written comments.  
**FR6.4** Feedback shall influence recommendation and trust algorithms.  
**FR6.5** Ratings for repeated misconduct shall trigger administrative review.  
**FR6.6** System shall allow rating another user who is a part of the activity.  
**FR6.7** System shall gain consent to access this review data for analysis.  
**FR6.8** System shall create an average rating from the past 10 activities attended per user.  
**FR6.9** System shall display user rating to premium members only.  
**Dependencies:** **F3** **Requirements to Fulfill:** **F2**, **F8**

## F7

Functional Requirements: User Profile and Preferences

**FR7.1** The system shall allow users to update personal information.  
**FR7.2** Users shall be able to manage interests, hobbies, preference tags, add/update gender and upload a profile picture.  
**FR7.3** Users shall be able to edit onboarding survey answers at any time.  
**FR7.4** Users shall be able to set availability windows.  
**FR7.5** Users shall be able to set distance radius for recommendations.  
**FR7.6** Premium users shall access advanced preference options.  
**Dependencies:** **F1** **Requirements to Fulfill:** **F2**, **F10**

## F8

Functional Requirements: Moderation and Safety

**FR8.1** Admins shall be able to view all user-generated activities.  
**FR8.2** Admins shall be able to remove activities violating guidelines.  
**FR8.3** Admins shall be able to review user reports.  
**FR8.4** Admins shall be notified of suspicious or repeated negative behaviour.  
**FR8.5** Admins shall be able to temporarily suspend or permanently ban users.  
**Dependencies:** **F1**, **F6**, **F4** **Requirements to Fulfill:** **F9**

## F9

Functional Requirements: Notification and Alert System

**FR9.1** The system shall notify users when recommended activities match their preferences.  
**FR9.2** The system shall notify users about upcoming activities.  
**FR9.3** The system shall notify hosts when a user joins or cancels.  
**FR9.4** The system shall send alerts for time/location changes.  
**FR9.5** The system shall support browser notifications (where permitted).

Dependencies: **F3**, **F5** Requirements to Fulfill: None

## **F10**

Functional Requirements: Membership and Subscription Management

**FR10.1** The system shall allow users to upgrade to premium membership.

**FR10.2** Premium members shall unlock larger group sizes.

**FR10.3** Premium members shall be able to create private groups.

**FR10.4** Premium members shall be able to create public groups.

**FR10.5** Users shall be able to cancel or downgrade subscriptions.

**FR10.6** The system shall restrict premium functions for non-premium accounts.

**FR10.7** The system shall display Google Ads for the standard membership tier.

Dependencies: **F1** Requirements to Fulfill: **F4**, **F7**

## 4 Other Nonfunctional Requirements

### 4.1 Performance Requirements

OuterCircI™ FFP shall meet the following performance expectations to ensure a functional and responsive initial release:

- The main dashboard (datafeed view) shall load within **3–4 seconds** under typical network conditions.
- Activity search and filtering operations shall return results within **2 seconds**.
- Joining or canceling an activity shall be processed within **2–3 seconds**.
- Group messaging latency shall not exceed **500 ms** under normal usage.
- The FFP shall support a minimum of **500 concurrent users**.
- Recommendation refreshes shall be performed asynchronously and may occur up to **every 10–30 seconds**.
- Real-time activity updates shall propagate to participants within **3 seconds**.

These targets reflect realistic expectations for an early-stage product deployed on scalable cloud services.

### 4.2 Security Requirements

The FFP shall implement essential security controls to ensure user safety and data protection, without full enterprise hardening.

#### Authentication & Authorization

- All communication between client and server shall occur over HTTPS (TLS 1.2+).
- User credentials must be hashed and never stored in plaintext.
- Basic JWT- or session-based authentication shall be implemented.

## Data Protection

- Sensitive user profile data shall be encrypted at rest where supported by the cloud provider.
- Access control shall be enforced at the API layer.
- User-generated content shall be sanitized to protect against basic XSS.

## Operational Security

- The system shall log critical actions (login, activity creation).
- FFP administrative actions must be accessible only to authorized admin accounts.

## 4.3 Software Quality Attributes

### Usability

- The interface shall be intuitive and responsive on modern desktop and mobile browsers.
- FFP workflows (join activity, create activity) shall be completable within **3 steps or fewer**.

### Reliability

- The FFP shall aim for **95% uptime** during the pilot phase.
- Basic error handling (fallback screens, retry mechanisms) shall be implemented.

### Scalability

- The FFP backend shall support simple horizontal scaling via cloud auto-scaling groups.
- Heavy optimization is not required for the FFP but architecture shall not block future scaling.

### Maintainability

- The codebase shall follow a modular structure to simplify iteration in post-FFP versions.
- API documentation may be minimal but must cover all exposed endpoints.

## **Extensibility**

- The FFP shall support future extension to mobile apps without requiring architectural redesign.
- New activity types shall be introducible with minimal backend changes.

## **Privacy**

- Users shall have control over visibility of their interests and profile data.
- Location tracking shall require explicit user consent.

## **4.4 Business Rules**

- Users must be at least 18 years old.
- Users must accept Terms of Service and Community Guidelines before first use.
- Only verified users may act as Hosts.
- Free-tier users may only create activities capped at 4 participants.
- Free-tier users will only create activities twice per month.
- Free-tier users will receive Google Ads
- Premium users may create flexible group sizes and public/private activities.
- Activities violating safety or legal guidelines shall be removed by administrators.
- Users with repeated safety flags may be restricted.

## 5 Other Requirements

### General Additional Requirements

- The platform shall comply with essential GDPR requirements for the FFP (consent, data handling transparency, user deletion).
- System logs shall be retained for a limited pilot duration (30–60 days).
- The FFP web app shall target basic accessibility compliance aligned with WCAG 2.0 (non-strict).
- Legal disclaimers and safety instructions must be shown at onboarding.