**Requirements Specification — Facebook-Style Newsfeed Service**

| **Document status** | **Draft v1.0** |
| --- | --- |
| **Prepared for** | Design & Architecture review |
| **Prepared by** | System-Design Team |
| **Date** | 25 May 2025 |

**1 Purpose**

Define the functional and non-functional requirements for a scalable **Newsfeed Service** (similar to Facebook’s), forming the basis for architecture, UML artefacts, and subsequent implementation.

**2 Scope**

The service delivers a personalised, chronologically-and-rank-ordered stream of posts (text, photo, video, link) from people, pages, and groups the user follows. It must support **billions of users**, **millions of concurrent sessions**, and near-real-time propagation of new content.

**3 Glossary**

| **Term** | **Definition** |
| --- | --- |
| **Post / FeedItem** | A unit of content shared by a user or page. |
| **Producer** | User or entity that creates a post. |
| **Consumer** | User requesting/scrolling their newsfeed. |
| **FNS** | Feed Notification Service – pushes & pulls feed data. |
| **NGS** | Newsfeed Generation Service – aggregates & ranks posts. |

**4 Stakeholders**

| **Stakeholder** | **Interest** |
| --- | --- |
| End Users | Fast, relevant, engaging feed. |
| Product Team | Feature velocity, engagement metrics. |
| Backend/Platform Team | Scalability, fault-tolerance. |
| DevOps/SRE | Operational simplicity, observability, CI/CD. |
| Data & ML Team | Access to post & engagement data for ranking models. |

**5 Functional Requirements**

**5.1 Feed Creation**

* **FR-01** Users can create posts containing text, images, videos, or links.
* **FR-02** Posts are stored with metadata: author ID, timestamp, privacy flags, media-IDs, location.
* **FR-03** Media uploads complete before the post becomes visible.

**5.2 Feed Distribution**

* **FR-04** When a post is created, it is propagated to followers’ feeds within **≤ 2 s** (p95).
* **FR-05** If a follower is online, the new post is pushed proactively (WebSocket/MQTT preferred).
* **FR-06** If a follower is offline, the post is queued and delivered on next login.

**5.3 Feed Retrieval**

* **FR-07** On opening the app, users receive the latest **N (≈ 100)** posts ranked by relevance & recency.
* **FR-08** Infinite scroll loads older posts in paginated batches.
* **FR-09** The feed honours privacy settings; users never see content they are not authorised to view.

**5.4 Ranking & Personalisation**

* **FR-10** NGS scores posts using recency, engagement, relationship strength, and ML signals.
* **FR-11** Ranking models can be A/B tested and hot-swapped without downtime.

**5.5 Engagement Updates**

* **FR-12** Likes/comments are reflected in the UI within **≤ 3 s** of action (fan-out-on-write/update).
* **FR-13** Counts remain eventually consistent across devices.

**5.6 Presence Management**

* **FR-14** System tracks online/offline status via heartbeat every 30 s.
* **FR-15** Friends see presence changes within **≤ 2 s**.

**6 Non-Functional Requirements**

| **Category** | **Requirement** |
| --- | --- |
| **Scalability** | Horizontal scaling to 100 M concurrent sessions; 10 M post writes per minute. |
| **Latency** | p95 end-to-end delivery < 2 s; feed read p95 < 300 ms. |
| **Availability** | ≥ 99.95 % monthly (includes regional failover). |
| **Consistency** | Timeline is *eventually consistent*; per-user ordering is preserved. |
| **Data Durability** | All posts durable in triplicate across AZs. |
| **Security & Privacy** | OAuth 2.0 auth, TLS 1.3 in transit, media at-rest encryption (KMS). |
| **Observability** | Structured logs, distributed tracing, SLIs (latency, error rate, freshness). |
| **Compliance** | GDPR/CCPA delete & export APIs; content retention policies. |

**7 Assumptions & Constraints**

1. Mobile clients support WebSockets; legacy clients fall back to long-polling.
2. Primary DB is a **NoSQL wide-column store (HBase/Cassandra)**; media in object storage (S3-like).
3. Cache layer (Redis/Elasticache) holds last 100 posts per user (LRU).
4. Sequence numbers are generated per conversation/user via Snowflake-style IDs.
5. System must run in at least two regions with active-active replication.

**9 Future Considerations**

* Edge caching/CDN for media thumbnails.
* ML-driven feed deduplication & story grouping.
* Real-time analytics pipeline (Kafka → Flink → ClickHouse).
* GraphQL gateway for flexible client queries.

**10 Approval**

| **Name** | **Role** | **Signature** | **Date** |
| --- | --- | --- | --- |