**Business Requirements Specification (BRS): Twitter-like Short Messaging Service**

**1. Introduction**

1.1. **Purpose:** This document outlines the business requirements for the development of an online social networking service similar to Twitter, where users can post and read short messages (tweets).

1.2. **Audience:** This document is intended for the development team, stakeholders, and anyone involved in the planning and execution of this project.

1.3. **Scope:** This BRS covers the core functionalities of the service, including user posting, following, media inclusion, and timeline generation. It focuses on the essential features for an initial version of the service.

**2. Business Goals**

2.1. Enable users to share short text-based messages (tweets) with their followers. 2.2. Allow users to follow other registered users to view their tweets. 2.3. Provide a platform for users to consume a real-time stream of tweets from the users they follow (timeline). 2.4. Support the inclusion of multimedia content (photos and videos) in tweets. 2.5. Ensure the system is highly scalable to accommodate a large number of users and tweets. 2.6. Allow unregistered users to view publicly available tweets.

**3. Functional Requirements**

3.1. **Tweet Posting:** \* **FR01:** Registered users shall be able to compose and post new tweets. \* **FR02:** The system shall record the user ID of the user who posted the tweet. \* **FR03:** The system shall record the content (text) of the tweet. \* **FR04:** Users shall have the option to include their current location with a tweet. \* **FR05:** The system shall provide an API endpoint for posting new tweets, accepting user ID, tweet data, and optional location information. \* **FR06:** Upon successful posting, the API shall return a unique identifier (URL) for the created tweet.

3.2. **User Following:** \* **FR07:** Registered users shall be able to follow other registered users. \* **FR08:** The system shall store the list of users that a particular user is following. \* **FR09:** The system shall support a large number of followers for each user. \* **FR10:** The system shall support a user following a large number of other users. \* **FR11:** The system shall provide mechanisms to manage user following relationships.

3.3. **Media Inclusion:** \* **FR12:** Registered users shall be able to include photos in their tweets. \* **FR13:** Registered users shall be able to include videos in their tweets. \* **FR14:** The system shall support the storage and retrieval of media files associated with tweets. \* **FR15:** The system shall maintain a link between a tweet and its associated media.

3.4. **Timeline Generation and Display:** \* **FR16:** Registered users shall have a personal timeline that displays the most recent tweets from the users they follow. \* **FR17:** The timeline shall be updated with new tweets in near real-time. \* **FR18:** The timeline shall display tweets containing text, photos, and videos. \* **FR19:** The system shall efficiently retrieve and order tweets for a user's timeline, considering the large number of potential followers. \* **FR20:** The system shall employ caching mechanisms to optimize timeline retrieval and reduce latency. \* **FR21:** The system shall rank and prioritize tweets in the timeline based on an internal algorithm (details to be defined).

3.5. **Tweet Reading (Public Access):** \* **FR22:** Unregistered users shall be able to view publicly accessible tweets. \* **FR23:** The system shall provide a mechanism to retrieve and display individual or a set of public tweets.

**4. Non-Functional Requirements**

4.1. **Scalability:** \* **NFR01:** The system shall be designed to handle a large number of concurrent users. \* **NFR02:** The system shall be able to store and manage billions of tweets. \* **NFR03:** The system architecture shall be horizontally scalable to accommodate future growth in users and data. \* **NFR04:** The system shall utilize database sharding or partitioning to distribute data and load.

4.2. **Performance:** \* **NFR05:** The system shall provide low latency for posting new tweets. \* **NFR06:** The system shall generate and display user timelines with minimal delay. \* **NFR07:** Read operations (fetching tweets) shall be highly optimized for speed.

4.3. **Availability:** \* **NFR08:** The system shall be highly available to ensure continuous operation. \* **NFR09:** The system shall incorporate fault tolerance mechanisms to minimize downtime.

4.4. **Security:** \* **NFR10:** The system shall ensure the security and privacy of user data. \* **NFR11:** Appropriate measures shall be implemented to prevent unauthorized access and data breaches.

**5. API Requirements (Initial)**

5.1. **Post Tweet API:** \* **Endpoint:** /api/tweets \* **Method:** POST \* **Request Parameters:** userId, tweetText, location (optional), media (optional) \* **Response:** URL of the newly created tweet.

5.2. **Get User Tweets API:** \* **Endpoint:** /api/users/{userId}/tweets \* **Method:** GET \* **Response:** List of tweets posted by the specified user.

**6. Data Requirements (Initial)**

6.1. **User Data:** User ID, Name, Email, Creation Time, etc. (Details may expand). 6.2. **Tweet Data:** Tweet ID, User ID, Content (Text), Timestamp, Location (Optional). 6.3. **Friendship Data:** User ID (Follower), User ID (Following). 6.4. **Media Data:** Media ID, Type (Image/Video), Description, Location (Storage Path). 6.5. **Feed Item Data:** Feed Item ID, User ID, Content (Reference to Tweet/Media).

**7. Future Considerations (Out of Scope for Initial Release)**

7.1. Direct Messaging 7.2. Hashtags and Trending Topics 7.3. Search Functionality 7.4. Retweeting/Sharing 7. Analytics

**📘 Business Requirements Specification (BRS)**

**Twitter-like Short Messaging Service**

**1. Introduction**

**1.1 Purpose**

This document defines the business and functional requirements for developing an online social networking service that allows users to share and consume short messages (tweets), similar in functionality to Twitter.

**1.2 Intended Audience**

This BRS is intended for all stakeholders including product managers, developers, UI/UX designers, quality assurance teams, and infrastructure engineers involved in the planning, development, and deployment of the system.

**1.3 Scope**

The document outlines the essential features for the **initial release** of the platform. These include user registration, posting tweets, following other users, media attachment, timeline generation, and public tweet viewing. Advanced features such as messaging, trends, and analytics are outside the scope of this version.

**2. Business Objectives**

1. Enable users to publish short text-based messages (tweets).
2. Allow users to follow other users and view their content.
3. Provide a real-time timeline of tweets from followed users.
4. Support photo and video attachments in tweets.
5. Allow public access to selected tweets without requiring registration.
6. Design a scalable system capable of handling high user volume and content throughput.

**3. Functional Requirements**

**3.1 Tweet Creation**

* **FR1**: Registered users can create and post new tweets.
* **FR2**: Each tweet must record the author’s user ID and timestamp.
* **FR3**: Tweets may include optional location metadata.
* **FR4**: Tweets may include attached media (photos/videos).
* **FR5**: A REST API endpoint shall be provided for posting tweets.
* **FR6**: Each tweet shall be assigned a unique URL upon creation.

**3.2 User Following**

* **FR7**: Users can follow and unfollow other users.
* **FR8**: The system must maintain and retrieve a user’s following and follower lists.
* **FR9**: The following system must support high fan-out relationships.
* **FR10**: APIs shall be available to manage follow/unfollow actions.

**3.3 Media Attachments**

* **FR11**: Users can attach one or more images or videos to a tweet.
* **FR12**: The system must store and retrieve media efficiently.
* **FR13**: Media must be linked to the corresponding tweet.

**3.4 Timeline Generation**

* **FR14**: Each user shall have a personalized timeline displaying tweets from followed users.
* **FR15**: Timeline should be updated in near real-time.
* **FR16**: Timeline must include tweets with media attachments.
* **FR17**: The system must order tweets by relevance and/or recency.
* **FR18**: Caching mechanisms must be used to improve timeline load performance.

**3.5 Public Tweet Access**

* **FR19**: Unregistered users may access publicly available tweets.
* **FR20**: A public endpoint must be provided to fetch and view such tweets.

**4. Non-Functional Requirements**

**4.1 Scalability**

* **NFR1**: The system must handle millions of concurrent users.
* **NFR2**: The architecture should support billions of tweets and associated media.
* **NFR3**: Horizontal scalability (via sharding/partitioning) must be supported.

**4.2 Performance**

* **NFR4**: Tweet creation and retrieval must be low-latency.
* **NFR5**: Timelines should load with minimal delay.
* **NFR6**: All read operations should be optimized for high throughput.

**4.3 Availability**

* **NFR7**: The platform must be designed for high availability (HA).
* **NFR8**: Redundancy and fault-tolerance must be implemented at every layer.

**4.4 Security**

* **NFR9**: User data must be securely stored and transmitted.
* **NFR10**: Authentication and authorization must be enforced on all write and user-specific endpoints.
* **NFR11**: Protection mechanisms must be in place against data breaches and abuse.

**5. API Specifications (Initial Version)**

**5.1 Post Tweet API**

* **Endpoint**: POST /api/tweets
* **Request Payload**:

json

CopyEdit

{

"userId": "string",

"tweetText": "string",

"location": "string (optional)",

"media": ["media1.jpg", "media2.mp4"]

}

* **Response**:

json

CopyEdit

{

"tweetId": "string",

"url": "https://app.com/tweets/{tweetId}"

}

**5.2 Get User Tweets API**

* **Endpoint**: GET /api/users/{userId}/tweets
* **Response**: Array of tweet objects including media URLs and timestamps.

**6. Data Model (Initial)**

**6.1 Users**

* user\_id, name, email, password\_hash, created\_at, is\_verified

**6.2 Tweets**

* tweet\_id, user\_id, content, timestamp, location

**6.3 Follows**

* follower\_id, followee\_id, followed\_at

**6.4 Media**

* media\_id, tweet\_id, type, description, media\_url, uploaded\_at

**6.5 Feed Items**

* feed\_id, user\_id, tweet\_id, rank\_score, inserted\_at

**7. Future Enhancements (Out of Scope)**

* Direct Messaging between users
* Hashtags and Trending Topics
* Full-text search and filtering
* Retweeting and quote tweets
* Tweet analytics and engagement metrics