

SOFTWARE REQUIREMENTS SPECIFICATION

Software Requirements Specification for a Game Streaming Application called
G Stream.



SEPTEMBER 2, 2023
BY HD SECTOR

Contents

1. Introduction	2
1.1 Purpose	2
1.2 Intended Use	2
1.3 Scope	2
2. Overall Description	2
2.1. Product Perspective	2
2.2. Product Features	2
2.3. User Classes and Characteristics	3
2.4. Operating Environment	3
3. Specific Requirements	3
3.1. Functional Requirements	3
3.1.1. User Registration	3
3.1.2. User Authentication	3
3.1.3. Profile Management	3
3.1.4. Content Discovery	4
3.1.5. Live Streaming	4
3.1.6. Chat and Interaction	4
3.1.7. Notifications	4
3.1.8. Monetization	4
3.1.9. Content Reporting and Moderation	4
3.2. Non-Functional Requirements	4
3.2.1. Performance	4
3.2.2. Security	5
3.2.3. Compatibility	5
3.2.4. Data Privacy	5
3.3 Technology Requirements	5
3.3.1. Programming Languages	5
3.3.2. Data Requirements	5
4. System Models	5
5. Test and Validation	6
6. Project Timeline and Milestones	6
7. Contact	7

1.Introduction

1.1 Purpose

The purpose of the GStream Streaming App is to provide gamers with a robust platform to stream their gameplay live and to offer viewers an engaging way to enjoy and interact with gaming content in real time. The app aims to foster a dynamic gaming community and enhance the gaming experience for both streamers and viewers.

1.2 Intended Use

The Game Streaming App enables gamers to broadcast their gameplay sessions in real time, allowing viewers to watch and interact with the streamers. It facilitates engagement through live chat, comments, and reactions, creating a social and interactive experience around gaming content.

1.3 Scope

The Game Streaming App will encompass the following features:

- User registration and authentication process to ensure secure access.
- User profiles for streamers and viewers, with customizable avatars and profile information.
- Live streaming functionality that allows streamers to broadcast their gameplay sessions.
- Interactive features like live chat, comments, and real-time reactions.
- Search and discovery tools to find streams based on game titles, genres, and streamer profiles.
- Notifications for followers and subscribers about new streams and interactions.
- Analytics dashboard for streamers to track viewership, engagement, and trends.
- Cross-platform compatibility for major operating systems (iOS, Android, Windows, macOS).

2. Overall Description

2.1. Product Perspective

The GStream app operates as a standalone live streaming platform, enabling users to discover, watch, and interact with live broadcasts from streamers. It is part of the larger GStream ecosystem, which includes a web platform and APIs for third-party integrations.

2.2. Product Features

Key features of the GStream app include:

- User registration and authentication.
- User profile management.
- Content discovery through categories, tags, and recommendations.
- Live streaming capabilities with chat functionality.
- Real-time notifications for followers and favorite streamers.
- Monetization options for streamers.
- Content reporting and moderation tools.

2.3. User Classes and Characteristics

- Viewers: Individuals interested in watching live streams.
- Streamers: Content creators who broadcast live.
- Moderators: Trusted users responsible for maintaining platform guidelines.

2.4. Operating Environment

The app is designed to run on iOS and Android smartphones (version X.X and above) and web browsers (supporting modern standards).

3. Specific Requirements

3.1. Functional Requirements

3.1.1. User Registration

- Users can sign up using email, phone number, or social media accounts.
- Validation of user-provided information during registration.
- Account verification via email or SMS.

3.1.2. User Authentication

- Secure login mechanism with password encryption.
- Support for two-factor authentication (2FA).
- Session management for logged-in users.

3.1.3. Profile Management

- Users can create and edit their profiles.
- Profile includes display name, profile picture, and brief description.

- Option to link social media profiles.

3.1.4. Content Discovery

- Browse live streams by category, tags, and popularity.
- Personalized content recommendations based on user preferences and viewing history.

3.1.5. Live Streaming

- Streamers can start and manage live broadcasts.
- Viewers can watch live streams and interact through chat.

3.1.6. Chat and Interaction

- Real-time chat for viewers to interact with streamers and other viewers.
- Moderation tools for streamers and moderators to manage chat.
- Emote and gifting system for viewer engagement.

3.1.7. Notifications

- Push notifications for follower activity (e.g., new followers, live stream alerts).
- Option to customize notification preferences.

3.1.8. Monetization

- Streamers can monetize their content through ads, donations, and subscription models.
- Integration with payment gateways (e.g., PayPal, Stripe) for transactions.
- Revenue tracking and withdrawal options for streamers.

3.1.9. Content Reporting and Moderation

- Users can report inappropriate content.
- Moderators can review and take action on reported content.
- Automated content flagging for potential violations.

3.2. Non-Functional Requirements

3.2.1. Performance

- App should load quickly and stream smoothly on various network conditions.

- Response time for user interactions should be under 2 seconds.

3.2.2. Security

- Secure storage and transmission of user data.
- Regular security audits and vulnerability assessments.

3.2.3. Compatibility

- Support for major web browsers (Chrome, Firefox, Safari) and mobile platforms (iOS, Android).
- Responsive design for various screen sizes.

3.2.4. Data Privacy

- Compliance with GDPR and other data privacy regulations.
- User data encryption at rest and in transit.
- Transparent privacy policy.

3.3 Technology Requirements

3.3.1. Programming Languages

Backend: Choose a backend programming language for server-side development. Common choices include Python (Django, Flask), Node.js (Express.js), Ruby (Ruby on Rails), or Java (Spring Boot).

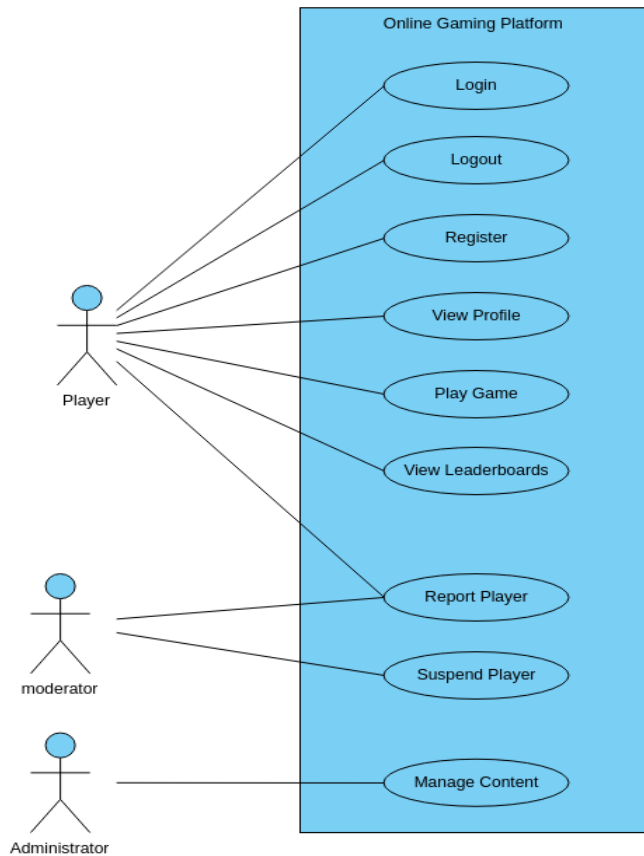
Frontend: Use HTML, CSS, and JavaScript (React, Angular, or Vue.js) for building the user interface and client-side functionality.

3.3.2. Data Requirements

- Use of a relational database (e.g., MySQL) for user profiles and interactions.
- Use of NoSQL databases (e.g., MongoDB) for real-time chat and notifications.

4. System Models

- Use Case Diagrams, Sequence Diagrams, State Diagrams, and Entity-Relationship Diagrams will be provided in separate documents as per project needs.



Player, Moderator and Administrator Relation Diagram

5. Test and Validation

During the project's testing phase the tests will be performed

- Performance Testing.
- Security Testing
- Functional Testing
- Usability and Accessibility Testing
- Platform and Device Specific Testing

6. Project Timeline and Milestones

- The project will be divided into phases: Planning, Design, Development, Testing, and Deployment.
- Milestones will be defined for each phase, with associated deliverables and deadlines.

Milestone	Reporting	Time
Planning		2 months
Designing	Submit Design	2-3 months
Development		6-8 months
Testing	Test Final work	2 months
Deployment	Hand Over	1 month

12. Payment Terms

We propose the following payment terms:

Paid on acceptance of this proposal.	10% (10%)
Paid on signing of our Application development agreement.	40% (50%)
Paid at 70% Application Demonstration.	25% (75%)
Paid at completion the Application.	25% (100%)

7. Contact

By Phone:

+8801706999666

Contact Us By Email

HDsector@gmail.com

On our website

www.hdsector.us

Agreement Signed By:

.....

Client Signature

.....

Authority Signature
Managing Director (MD)
HD SECTOR