Use Case Specification
for
Soccer Live

Version: 1.0
Prepared by:

- Usman Awan (22F-3378)
- Muhammad Houd (22F-3376)
- Mutassim Billa (22F-8772)

Organization: FAST NUCES CFD

Date: 8th October, 2024

Contents

Revision History

Name	Date	Reason For Changes	Version

Use Case: Live Match Streaming

Nr.	Section	Content / Explanation
1	Designation	UC-001
2	Name	Live Match Streaming
3	Authors	Muhammad Houd, Usman Awan, Mutasim Billa
4	Priority	High
5	Criticality	Medium – failure may result in loss of user engagement and dissatisfaction
6	Source	Stakeholder feedback, User survey
7	Person responsible	Product Manager
8	Description	This use case allows users to stream live soccer matches in real-time from the app. Users can interact and view the match from various devices, enhancing their viewing experience.
9	Trigger event	User clicks on a live match stream link
10	Actors	User
11	Pre-conditions	User is logged in and has a stable internet connection. The match is scheduled to be live.
12	Post-conditions	The user is successfully watching the live stream, and the streaming service is actively delivering the content.
13	Result	Live match stream is displayed on the user's device, providing real-time video and audio coverage of the event.
14	Main scenario	- User navigates to the "Live Matches" section - User selects a live match link - Streaming server retrieves the match data from the CDN - The match stream starts and is displayed on the user's screen

Nr.	Section	Content / Explanation
		- User can interact with the interface (pause, rewind, chat) while watching the stream.
15	Alternative scenarios	- **Low Bandwidth Adaptation**: If the user has a low bandwidth connection, the system automatically adjusts the stream quality to prevent buffering. - **User Changes Match**: User decides to switch to a different ongoing match from the live matches list. - **User Uses Picture-in-Picture**: User can minimize the stream to a smaller window while using other features of the app.
16	Exception scenarios	- **Stream Fails to Load**: The live stream fails to load due to a server error or network issue, prompting the user with a retry option. - **User Connection Lost**: The user's internet connection drops, resulting in a temporary disconnection from the stream. - **Match Canceled**: If the match is canceled or postponed, the user receives a notification and is redirected to available matches.
17	Qualities	- **Performance**: The streaming service must deliver high-definition quality with minimal latency to ensure a smooth viewing experience. - **Scalability**: The system should efficiently handle varying numbers of concurrent viewers without degrading performance. - **Reliability**: The streaming service should maintain a high uptime percentage and quickly recover from any disruptions. - **Usability**: The user interface must be intuitive, allowing easy navigation between matches and quick access to streaming features. - **Accessibility**: The system should provide options for captions and audio descriptions to accommodate users with disabilities.

Use Case: Match Notifications

Nr.	Section	Content / Explanation
1	Designation	UC-002
2	Name	Match Notifications
3	Authors	Muhammad Houd, Usman Awan, Mutasim Billa
4	Priority	High
5	Criticality	Medium – failure to send notifications may result in user disengagement and missed events.
6	Source	Stakeholder feedback, User survey
7	Person responsible	Notification System Manager
8	Description	This use case enables users to receive real-time notifications about upcoming matches, score updates, and important events during the match. Users can customize their notification preferences to suit their interests.
9	Trigger event	Scheduled match time or significant event during a match (e.g., goal, card, halftime).
10	Actors	User
11	Pre-conditions	User must be logged into their account and have opted in to receive notifications.
12	Post-conditions	Users receive timely notifications on their selected devices according to their preferences.
13	Result	Users are informed of match start times, scores, and critical events, enhancing their engagement and experience.
14	Main scenario	- User enables match notifications in the app settings. - Notification Service retrieves match schedules from the Match Data Provider. - At the scheduled match time, the system sends a push notification to the user's device.

Nr.	Section	Content / Explanation
		- During the match, if a goal is scored, an additional notification is sent. - User receives and views notifications on their device.
15	Alternative scenarios	- **User Modifies Preferences**: User decides to change notification preferences for specific teams or match types. - **User Receives Delayed Notification**: In cases of network congestion, the user receives notifications slightly delayed but still timely enough to catch critical match events. - **User Unsubscribes**: User can opt-out of match notifications at any time through the settings menu.
16	Exception scenarios	- **Notification Delivery Failure**: The notification service encounters an error and fails to send notifications, prompting the user with a fallback message. - **User Device Not Connected**: If the user's device is not connected to the internet, notifications will be queued and sent once connectivity is restored. - **Invalid Notification Settings**: If the user has selected an invalid configuration (e.g., notifications for a non-existent team), the system alerts them to correct the settings.
17	Qualities	- **Performance**: The notification system must deliver notifications instantly to enhance user engagement. - **Reliability**: The notification service should consistently deliver messages without failure, even during peak times. - **Usability**: The user interface for setting notification preferences should be straightforward and intuitive. - **Scalability**: The system must handle a growing number of users without impacting performance. - **Configurability**: Users should have the ability to customize notification types, such as match reminders, score alerts, and player statistics.

Use Case: Real-Time Match Updates

Nr.	Section	Content / Explanation
1	Designation	UC-003
2	Name	Real-Time Match Updates
3	Authors	Muhammad Houd, Usman Awan, Mutasim Billa
4	Priority	High
5	Criticality	High – failure to provide updates may lead to user dissatisfaction and decreased engagement.
6	Source	Stakeholder feedback, User requirements document
7	Person responsible	Match Update Manager
8	Description	This use case allows users to receive real-time updates about ongoing matches, including score changes, player substitutions, and other significant events. It aims to enhance user engagement by keeping them informed about match progress without needing to watch the stream.
9	Trigger event	Significant event during the match, such as a score change, yellow/red card issuance, or halftime.
10	Actors	User
11	Pre-conditions	User must be logged into their account and have selected the match they want to follow.
12	Post-conditions	Users receive timely updates on match events and can view a detailed timeline of updates.
13	Result	Users are informed about match developments in real-time, improving their overall experience and satisfaction with the service.
14	Main scenario	- User selects a match to follow on the application. - The Update Notification Service connects to the Match Data Provider.

Nr.	Section	Content / Explanation
		- As events occur (e.g., goals, cards), the Match Data Provider sends updates to the Notification Service. - The Notification Service pushes updates to the user's device instantly. - User receives and views updates in the app or website.
15	Alternative scenarios	- **User Refreshes Updates**: User can manually refresh the updates if they suspect a delay in notifications. - **User Changes Match**: User opts to switch to another match to receive updates on a different game. - **User Adjusts Notification Preferences**: User modifies settings to receive fewer updates (e.g., only goal alerts).
16	Exception scenarios	- **Network Error**: If the user's device experiences connectivity issues, updates will be queued and sent once the connection is restored. - **Server Downtime**: If the Match Data Provider is temporarily down, the system informs users of the unavailability of updates. - **Invalid Match Selection**: If the user selects a match that has not started or is already completed, the system prompts them with an appropriate message.
17	Qualities	- **Performance**: Updates must be delivered within seconds of an event occurring to ensure users receive timely information. - **Reliability**: The system should consistently provide updates without loss or delay, even during high-traffic periods. - **Usability**: The user interface for viewing updates should be clean and easy to navigate. - **Scalability**: The system must handle a growing number of simultaneous users following various matches. - **Interactivity**: Users should be able to engage with updates, such as clicking for more details on specific events or players.

Use Case: User Profile Management

Nr.	Section	Content / Explanation
1	Designation	UC-004
2	Name	User Profile Management
3	Authors	Muhammad Houd, Usman Awan, Mutasim Billa
4	Priority	Low
5	Criticality	Medium – failure to manage user profiles may lead to poor user experience and data integrity issues.
6	Source	Stakeholder interviews, User requirements document
7	Person responsible	User Experience Manager
8	Description	This use case allows users to create, edit, and manage their profiles, including personal information, preferences, and privacy settings. It enhances user experience by providing a personalized environment within the application.
9	Trigger event	User initiates profile creation or modification through the settings menu.
10	Actors	User
11	Pre-conditions	User must be logged in to access profile management features.
12	Post-conditions	User profile is successfully created or updated, and the changes are reflected in the system.
13	Result	The user’s profile is updated or created, enhancing personalization and user engagement with the application.
14	Main scenario	- User navigates to the profile management section. - User selects to create a new profile or edit an existing one. - User fills out the required fields (e.g., name, email, profile picture). - User submits the changes.

Nr.	Section	Content / Explanation
		- The system validates the input and updates the user profile in the database. - The system confirms the successful update to the user.
15	Alternative scenarios	- **Profile View**: User can view their profile without making changes. - **Profile Deletion**: User opts to delete their profile and confirms the action. - **Password Change**: User selects the option to change their password within the profile management interface.
16	Exception scenarios	- **Invalid Input**: If the user enters invalid data (e.g., incorrect email format), the system displays an error message. - **Network Issues**: If there's a network failure during profile update, the system prompts the user to retry the action. - **Session Timeout**: If the user session expires while editing, the system prompts the user to log in again.
17	Qualities	- **Usability**: The interface should be intuitive, allowing users to easily manage their profiles. - **Performance**: Profile updates should occur within a few seconds to ensure a responsive user experience. - **Security**: User data must be securely stored, and changes should be protected against unauthorized access. - **Reliability**: The system must consistently save and reflect changes without data loss. - **Scalability**: The profile management feature should handle a growing number of users without degradation in performance.

Use Case: Language Selection

Nr.	Section	Content / Explanation
1	Designation	UC-005
2	Name	Language Selection
3	Authors	Muhammad Houd, Usman Awan, Mutasim Billa
4	Priority	Medium
5	Criticality	Medium – failure to implement language selection may lead to user dissatisfaction and limit accessibility.
6	Source	Stakeholder interviews, User requirements document
7	Person responsible	User Experience Manager
8	Description	This use case allows users to select their preferred language for the application interface. This enhances accessibility and user experience by catering to a diverse user base.
9	Trigger event	User clicks on the language selection menu or dropdown.
10	Actors	User
11	Pre-conditions	User must be logged in (if required) to access language selection features.
12	Post-conditions	The application interface reflects the selected language across all relevant components.
13	Result	The user's preferred language is saved, and the application interface is displayed in that language.
14	Main scenario	- User navigates to the language selection option in settings. - User clicks on the dropdown menu. - User selects their preferred language from the list. - The system updates the language settings and refreshes the interface

Nr.	Section	Content / Explanation
		accordingly. - The system confirms the language change to the user.
15	Alternative scenarios	- **Default Language**: If no language is selected, the application defaults to the system language. - **Language Reset**: User can reset the language selection back to the default language. - **Language Preview**: User can preview how the application will look in the selected language before confirming.
16	Exception scenarios	- **Unsupported Language**: If a user selects a language that is not supported, the system displays an error message. - **Network Issues**: If there's a failure in loading language files, the application retains the previous language setting. - **Session Timeout**: If the user's session expires during language selection, they are prompted to log in again.
17	Qualities	- **Usability**: The language selection interface should be intuitive and easy to navigate. - **Performance**: Language changes should be applied instantly without noticeable delay. - **Accessibility**: The system should support multiple languages to cater to diverse user needs. - **Scalability**: The language selection feature should easily accommodate additional languages in the future. - **Localization**: All aspects of the application, including content and UI elements, must be accurately translated and culturally appropriate.

Use Case: Low-Bandwidth Streaming Mode

Nr.	Section	Content / Explanation
1	Designation	UC-006
2	Name	Low-Bandwidth Streaming Mode
3	Authors	Muhammad Houd, Usman Awan, Mutasim Billa
4	Priority	High
5	Criticality	High – failure to implement low-bandwidth streaming may lead to user frustration and loss of users in areas with limited connectivity.
6	Source	Stakeholder interviews, User requirements document
7	Person responsible	Technical Lead
8	Description	This use case allows users to enable a low-bandwidth streaming mode, optimizing video quality and reducing buffering for users with limited internet speed. This feature is critical for ensuring a smooth viewing experience in areas with poor connectivity.
9	Trigger event	User selects the low-bandwidth streaming option from the settings menu.
10	Actors	User
11	Pre-conditions	- User must have an active account and be logged in to access streaming settings. - The application must detect the current network quality.
12	Post-conditions	The system streams content in a lower resolution or bitrate, optimizing for the user's network conditions.
13	Result	The user successfully streams video content with reduced data consumption, experiencing minimal buffering and interruptions.
14	Main scenario	- User navigates to the settings menu. - User selects the “Low-Bandwidth Mode” option. - The system confirms the selection and adjusts streaming settings accordingly.

Nr.	Section	Content / Explanation
		- User resumes watching content, experiencing improved streaming performance. - The system maintains a connection to monitor network quality for any adjustments needed.
15	Alternative scenarios	- **Auto-Adjustment**: The system automatically switches to low-bandwidth mode if it detects a significant drop in network quality. - **Manual Override**: User can switch back to regular streaming mode at any time, allowing them to control their viewing experience. - **Notification**: User receives a notification about network quality changes and suggestions to enable low-bandwidth mode.
16	Exception scenarios	- **Network Disruption**: If the network connection is lost, the system pauses the stream and notifies the user. - **Unsupported Content**: If a specific video does not support low-bandwidth streaming, the system informs the user and suggests alternative content. - **Playback Error**: If an error occurs during playback in low-bandwidth mode, the system displays an error message and allows the user to retry or switch to another mode.
17	Qualities	- **Usability**: The option to enable low-bandwidth mode should be easily accessible and clearly labeled in the user interface. - **Performance**: Streaming in low-bandwidth mode should minimize buffering and provide a smooth viewing experience. - **Adaptability**: The system should be able to adapt the streaming quality based on real-time network conditions. - **Reliability**: The low-bandwidth mode should function consistently across different devices and network types. - **User Feedback**: The system should collect feedback from users regarding their experience in low-bandwidth mode to inform future improvements.

Use Case: Favorites Management

Nr.	Section	Content / Explanation
1	Designation	UC-007
2	Name	Favorites Management
3	Authors	Muhammad Houd, Usman Awan, Mutasim Billa
4	Priority	Low
5	Criticality	Medium – failure to manage favorites can lead to user dissatisfaction and hinder user engagement.
6	Source	Stakeholder interviews, User requirements document
7	Person responsible	Product Manager
8	Description	This use case allows users to manage their favorite matches or content within the application, providing an easy way to access preferred items quickly. The feature enhances user experience by streamlining content discovery and interaction.
9	Trigger event	User selects the option to add or remove content from their favorites list.
10	Actors	User
11	Pre-conditions	- User must have an active account and be logged in. - Content must be available for favoriting (e.g., matches, videos).
12	Post-conditions	The system updates the user's favorites list accordingly, and the user can view and manage their favorites at any time.
13	Result	The user's favorites list is updated, allowing them to quickly access their preferred content.
14	Main scenario	- User browses through available content. - User selects a match or video to favorite. - The system confirms the action, adding the item to the user's favorites list.

Nr.	Section	Content / Explanation
		- User can access their favorites through a dedicated menu or section. - User can view, remove, or reorder items in the favorites list as needed.
15	Alternative scenarios	- **Bulk Action**: User selects multiple items to add or remove from their favorites at once. - **Search Favorites**: User uses a search function to find specific favorites in their list. - **Recommendations**: The system suggests content based on the user's favorites, enhancing discovery.
16	Exception scenarios	- **Item Not Found**: If the content to be favorited is no longer available, the system notifies the user and prevents the action. - **Network Error**: If a network issue occurs while updating the favorites list, the system alerts the user and attempts to retry the action. - **Duplicate Favorite**: If the user tries to favorite an already favorited item, the system prompts a message indicating the item is already in the favorites list.
17	Qualities	- **Usability**: The process of adding or removing favorites should be intuitive and easily accessible. - **Performance**: The system should quickly update the favorites list without noticeable delays. - **Scalability**: The favorites management feature should efficiently handle a growing number of items as the user's preferences evolve. - **Reliability**: The system should consistently save changes to the favorites list and maintain its integrity across sessions. - **User Feedback**: The system should collect feedback on favorites management to continuously improve user experience.

Use Case: Authentication

Nr.	Section	Content / Explanation
1	Designation	UC-008
2	Name	Authentication
3	Authors	Muhammad Houd, Usman Awan, Mutasim Billa
4	Priority	High
5	Criticality	High – improper authentication can lead to unauthorized access or security breaches.
6	Source	Security requirements document, IT compliance policies
7	Person responsible	IT Security Manager
8	Description	This use case defines the process of verifying user identity to ensure that only authorized users gain access to the system. This use case is included within the "Login" and "Sign Up" use cases.
9	Trigger event	A user submits login credentials or a new account is created during sign-up.
10	Actors	User, Administrator
11	Pre-conditions	- The user must provide login credentials. - The authentication server must be available.
12	Post-conditions	- The user is authenticated and granted access. - If authentication fails, an error message is shown.
13	Result	Successful authentication and access to restricted features, or an error notification on failure.
14	Main scenario	- The system receives the user's credentials. - The system checks the credentials against the database or identity service. - If valid, the user is authenticated. - The user is granted access to the system.

Nr.	Section	Content / Explanation
15	Alternative scenarios	- **Two-Factor Authentication**: After initial authentication, the user is prompted for a second verification code (e.g., via SMS). - **Single Sign-On (SSO)**: User is authenticated through a third-party identity provider (e.g., Google, Microsoft).
16	Exception scenarios	- **Authentication Timeout**: If the system fails to authenticate within a set time, the user is prompted to retry. - **Invalid Credentials**: If the credentials are incorrect, the user is notified and prompted to re-enter them.
17	Qualities	- **Security**: Strong encryption for user credentials during transmission. - **Scalability**: Must handle a large volume of authentication requests. - **Availability**: Must be highly reliable to ensure constant authentication service.

Use Case: User Support and Feedback

Nr.	Section	Content / Explanation
1	Designation	UC-009
2	Name	User Support and Feedback
3	Authors	Muhammad Houd, Usman Awan, Mutasim Billa
4	Priority	Medium
5	Criticality	Medium – while failure in this use case may not directly impact functionality, it can lead to user dissatisfaction and increased churn.
6	Source	Stakeholder interviews, User experience research, Support documentation
7	Person responsible	Customer Support Manager
8	Description	This use case describes how users can provide feedback or seek support regarding their experience with the application. The process enables the company to gather insights for improvement and assist users with any issues they encounter.
9	Trigger event	User clicks on the "Support" or "Feedback" option in the application menu.
10	Actors	User
11	Pre-conditions	- User must be logged into the application. - User must have internet access.
12	Post-conditions	- User receives confirmation that their feedback has been submitted or support request acknowledged. - Support team is notified of the new request.
13	Result	User feedback is collected, and support requests are logged for follow-up by the support team.
14	Main scenario	- User navigates to the "Support" or "Feedback" section. - User selects the type of request (feedback, technical support, etc.). - User fills out the form with details (issue description, contact information).

Nr.	Section	Content / Explanation
		- User submits the form. - System acknowledges receipt of the request and provides a reference number.
15	Alternative scenarios	- **Chat Support**: User opts to initiate a live chat with a support representative instead of filling out a form. - **Feedback Categories**: User selects a category for their feedback (e.g., bug report, feature request) to streamline the support process. - **Feedback Submission Error**: If the submission fails, the user is prompted to check their internet connection or provide missing information.
16	Exception scenarios	- **System Downtime**: If the support system is temporarily unavailable, the user receives an error message indicating that support is currently down. - **Invalid Input**: If the user provides invalid information (e.g., an email address format error), the system prompts the user to correct it. - **Feedback Limit Exceeded**: If a user tries to submit feedback too frequently, they are notified of a limit and suggested to wait before submitting again.
17	Qualities	- **Usability**: The support and feedback interface should be intuitive and easy to navigate for users. - **Responsiveness**: The system should respond quickly to user input and provide instant feedback on submissions. - **Security**: User data must be securely stored and handled, complying with privacy regulations. - **Scalability**: The feedback and support system should handle an increasing number of requests without degradation in performance. - **Analytics**: The system should provide tools for analyzing user feedback trends and support request types for continuous improvement.

Use Case: Watch Match Highlights

Nr.	Section	Content / Explanation
1	Designation	UC-010
2	Name	Watch Match Highlights
3	Authors	Muhammad Houd, Usman Awan, Mutasim Billa
4	Priority	High
5	Criticality	Medium – a failure in this use case can affect user engagement and satisfaction but does not prevent access to other core functionalities.
6	Source	Stakeholder interviews, User experience research, Market analysis
7	Person responsible	Content Manager
8	Description	This use case outlines how users can access and watch highlights of completed matches. The feature provides users with a quick way to catch up on important moments without watching the full match.
9	Trigger event	User selects the “Watch Highlights” option for a specific match from the match list.
10	Actors	User
11	Pre-conditions	- User must be logged into the application. - Highlights must be available for the selected match. - User must have a stable internet connection.
12	Post-conditions	- User has successfully viewed the match highlights. - User can rate or share highlights if the feature is enabled.
13	Result	The user is able to view match highlights, with the option to interact (like, share, comment). The system logs user engagement metrics for analytics.
14	Main scenario	- User navigates to the match list. - User selects a completed match. - User clicks on the "Watch Highlights" button.

Nr.	Section	Content / Explanation
		- The system loads the highlights video. - User views the highlights. - User may provide feedback or share the highlights.
15	Alternative scenarios	- **User wants to see highlights for a future match**: The system informs the user that highlights are only available for completed matches. - **User selects a specific moment**: The user can navigate to specific highlights based on key events (goals, fouls) marked within the video.
16	Exception scenarios	- **Highlights not available**: If no highlights are available for the selected match, the user receives a message indicating that highlights are currently not available. - **Video playback issues**: If there is a problem with video streaming, the user is prompted to check their internet connection or try again later.
17	Qualities	- **Usability**: The highlight viewing feature should be user-friendly and accessible within a few clicks. - **Performance**: Video loading times should be minimal, ensuring smooth playback without buffering. - **Scalability**: The system should handle high traffic, especially during popular matches. - **Interactivity**: Users should have options to interact with highlights (sharing, rating) to enhance engagement. - **Analytics**: The system should capture view metrics and user interactions for continuous improvement.

Use Case: Social Media Sharing

Nr.	Section	Content / Explanation
1	Designation	UC-011
2	Name	Social Media Sharing
3	Authors	Muhammad Houd, Usman Awan, Mutasim Billa
4	Priority	High
5	Criticality	Medium – failure in this use case can lead to decreased user engagement and marketing reach but does not affect core functionalities.
6	Source	Stakeholder interviews, User feedback, Market research
7	Person responsible	Marketing Manager
8	Description	This use case describes how users can share content from the application to various social media platforms, enhancing user engagement and increasing visibility for the application.
9	Trigger event	User selects the "Share" option on a piece of content (e.g., article, video, match highlights).
10	Actors	User
11	Pre-conditions	- User must be logged into the application. - User must have linked social media accounts. - Content to be shared must be available and not restricted.
12	Post-conditions	- The selected content is shared successfully to the user's chosen social media platform. - The user receives confirmation of the share action.
13	Result	The content is published on the user's social media profile, and analytics data is updated to reflect the engagement generated from the share.

Use Cases Specifications for Soccer Live

Nr.	Section	Content / Explanation
14	Main scenario	- User navigates to the content they wish to share. - User clicks the "Share" button. - User selects the desired social media platform. - The system prompts the user to add a comment or caption (optional). - User confirms the share. - The system posts the content on the selected platform. - User receives a confirmation message.
15	Alternative scenarios	- **User wants to edit content before sharing**: User chooses to edit the caption or select a different image before confirming the share. - **User cancels the sharing action**: The user can cancel the sharing process before confirmation.
16	Exception scenarios	- **User is not logged in**: If the user is not logged in, the system prompts the user to log in before sharing. - **Linking social media account failed**: If there's an error while linking the social media account, the system shows an error message and suggests troubleshooting steps. - **Content sharing limits reached**: If the content has reached its maximum sharing limit on a specific platform, the user is notified.
17	Qualities	- **Usability**: The sharing process should be intuitive and quick, allowing users to share content within a few clicks. - **Performance**: Sharing actions should occur swiftly with minimal loading times. - **Interoperability**: The system should seamlessly integrate with various social media platforms. - **Analytics**: The system should track the success of shared content (likes, shares, comments) for future marketing efforts. - **Security**: Ensure that user data and social media credentials are protected during the sharing process.

Use Case: Chat with Other Fans

Nr.	Section	Content / Explanation
1	Designation	UC-012
2	Name	Chat with Other Fans
3	Authors	Muhammad Houd, Usman Awan, Mutasim Billa
4	Priority	Medium
5	Criticality	Medium – failure in this use case could lead to reduced user engagement and satisfaction.
6	Source	Stakeholder interviews, User feedback, Market research
7	Person responsible	Community Manager
8	Description	This use case describes how users can engage in real-time chat with other fans of the same team or event, fostering community interaction and enhancing user experience.
9	Trigger event	User clicks on the "Chat" feature from the application interface.
10	Actors	User
11	Pre-conditions	- User must be logged into the application. - User must have an active internet connection. - Chat feature must be enabled for the event or team.
12	Post-conditions	- User is engaged in a chat session with other fans. - The chat history is saved for future reference. - User receives notifications for new messages.
13	Result	The user successfully engages in conversation with other fans, enhancing community interaction and possibly increasing the time spent in the app.
14	Main scenario	- User navigates to the event or team page. - User clicks on the "Chat" button.

Nr.	Section	Content / Explanation
		- The system connects the user to the chat server. - The user sees the chat interface with existing messages. - User types a message and sends it. - Other fans in the chat see the message and can respond. - User engages in ongoing conversations.
15	Alternative scenarios	- **User chooses to leave the chat**: User decides to exit the chat at any point. - **User wants to change chat rooms**: User switches to a different chat related to another event or team.
16	Exception scenarios	- **Chat server is down**: If the chat server is unavailable, the user receives a notification indicating the issue. - **User sends a message but it fails**: If the message fails to send due to connectivity issues, the system prompts the user to try again. - **User is blocked by another user**: If another user blocks the user, they will no longer receive messages from that user.
17	Qualities	- **Usability**: The chat interface should be user-friendly, allowing easy message sending and reading. - **Responsiveness**: Messages should be sent and received in real-time with minimal lag. - **Scalability**: The system should support a large number of simultaneous users without performance degradation. - **Security**: User data and conversations should be secure, with measures in place to prevent harassment and spam. - **Moderation**: There should be mechanisms for reporting inappropriate behavior or content in the chat.

Use Case: Content Management

Nr.	Section	Content / Explanation
1	Designation	UC-013
2	Name	Content Management
3	Authors	Muhammad Houd, Usman Awan, Mutasim Billa
4	Priority	High
5	Criticality	High – failure in this use case could lead to outdated or incorrect content being displayed to users.
6	Source	Stakeholder interviews, User requirements document
7	Person responsible	Content Manager
8	Description	This use case outlines how administrators can manage content within the system, including creating, updating, and deleting content items.
9	Trigger event	Administrator logs into the content management system and selects the content management option.
10	Actors	Administrator
11	Pre-conditions	- Administrator must have appropriate access rights. - Content management system must be operational. - Valid content must be available for addition or editing.
12	Post-conditions	- Content changes are saved in the system. - Users can view the updated content. - System logs the actions taken by the administrator.
13	Result	Successful management of content, ensuring that the content displayed to users is accurate and up to date.
14	Main scenario	- Administrator logs into the content management system. - Administrator selects the "Manage Content" option.

Nr.	Section	Content / Explanation
		- Administrator views existing content. - Administrator chooses to add, edit, or delete content. - If adding or editing, administrator enters content details and submits changes. - System validates the input and saves the changes. - System confirms the successful action to the administrator.
15	Alternative scenarios	- **Bulk content upload**: Administrator selects the option to upload multiple content items at once instead of managing them individually. - **Content preview**: Before saving changes, the administrator may choose to preview how the content will appear to users.
16	Exception scenarios	- **Unauthorized access**: If an administrator attempts to access content management without proper permissions, the system displays an error message. - **Input validation failure**: If the administrator enters invalid content data (e.g., too long, missing required fields), the system prompts the administrator to correct the input.
17	Qualities	- **Usability**: The content management interface should be intuitive and easy to navigate. - **Security**: Access to content management features must be restricted to authorized users. - **Performance**: Changes should be processed and reflected in the system without significant delay. - **Scalability**: The system should handle a growing volume of content without performance degradation. - **Auditability**: The system should maintain logs of all content management actions for accountability and traceability.

Use Case: Manage User Accounts

Nr.	Section	Content / Explanation
1	Designation	UC-014
2	Name	Manage User Accounts
3	Authors	Muhammad Houd, Usman Awan, Mutasim Billa
4	Priority	High
5	Criticality	Critical – failure to manage user accounts effectively can lead to unauthorized access or data breaches.
6	Source	Stakeholder interviews, User requirements document
7	Person responsible	User Account Manager
8	Description	This use case outlines how administrators can manage user accounts, including creating, updating, deactivating, and deleting user profiles.
9	Trigger event	Administrator logs into the user management system and selects the user management option.
10	Actors	Administrator
11	Pre-conditions	- Administrator must have appropriate access rights. - User management system must be operational. - Valid user data must be available for processing.
12	Post-conditions	- User accounts are successfully created, updated, deactivated, or deleted. - System logs account management actions. - Users receive notifications for relevant actions (e.g., account creation, deactivation).
13	Result	Successful management of user accounts, ensuring that only authorized users have access and that their information is up-to-date.

Nr.	Section	Content / Explanation
14	Main scenario	- Administrator logs into the user management system. - Administrator selects the "Manage User Accounts" option. - Administrator views existing user accounts. - Administrator chooses to create, update, deactivate, or delete a user account. - If creating or updating, the administrator enters the required user details and submits the changes. - System validates the input and saves the changes. - System confirms the successful action to the administrator.
15	Alternative scenarios	- **Bulk user import**: Administrator selects the option to import multiple user accounts from a CSV or other file format. - **Account role assignment**: Administrator selects a user account to assign or modify roles and permissions.
16	Exception scenarios	- **Unauthorized access**: If an administrator attempts to access user management without proper permissions, the system displays an error message. - **Input validation failure**: If the administrator enters invalid user data (e.g., incorrect email format, missing required fields), the system prompts the administrator to correct the input.
17	Qualities	- **Usability**: The user management interface should be intuitive and easy to navigate. - **Security**: User account management features must be restricted to authorized users only. - **Performance**: Changes to user accounts should be processed and reflected in the system without significant delay. - **Scalability**: The system should handle an increasing number of user accounts without performance degradation. - **Auditability**: The system should maintain logs of all user account management actions for accountability and traceability.

Use Case: Login

Nr.	Section	Content / Explanation
1	Designation	UC-0015
2	Name	Login
3	Authors	Muhammad Houd, Usman Awan, Mutasim Billa
4	Priority	High
5	Criticality	High – failure in login leads to user frustration and loss of access to the system.
6	Source	Stakeholder interviews, user feedback, security requirements document
7	Person responsible	IT Security Manager
8	Description	This use case outlines the process for users to log in to their accounts by providing valid credentials. This use case includes the "Authentication" use case to verify the user's identity.
9	Trigger event	A user opens the login page and attempts to access their account by entering credentials.
10	Actors	User, Administrator
11	Pre-conditions	- The user has a valid account. - The login page is accessible. - The user must provide credentials.
12	Post-conditions	- The user is logged in and redirected to the dashboard. - If login fails, the user is notified.
13	Result	Successful login and access to the system, or an error message indicating failure.
14	Main scenario	- The user navigates to the login page. - The user enters credentials (username and password). - The system includes the "Authentication" use case to verify credentials. - If valid, the user is logged in and redirected to their dashboard.

Use Cases Specifications for Soccer Live

Nr.	Section	Content / Explanation
15	Alternative scenarios	- **Password Reset**: The user clicks "Forgot Password" and resets the password to regain access. - **Social Login**: The user logs in using third-party credentials (e.g., Google, Facebook).
16	Exception scenarios	- **Account Locked**: After multiple failed login attempts, the user is locked out. - **Server Unavailability**: If the login server is down, the user receives an error message indicating service disruption.
17	Qualities	- **Usability**: Login process should be intuitive and efficient. - **Security**: Strong authentication and secure handling of credentials. - **Performance**: The system should handle high traffic for concurrent logins efficiently.