



IntapTV Software Requirements Specification

Version 1.5

April 2020

<https://intap.tv>

© 2020 Intap, Inc.

Table of Contents

- 1. PREFACE 5**
 - 1.1. Introduction..... 5**
 - 1.2. Purpose and Goal of Document 5**
 - 1.3. Audience of Document..... 5**
 - 1.4. Glossary 6**
 - 1.5. References 8**
- 2. OVERALL DESCRIPTION..... 9**
 - 2.1. Purpose of System 9**
 - 2.2. User Classes and Characteristics 9**
 - 2.3. Description of Solution 10**
 - 2.4. Constraints, Assumptions and Dependencies..... 14**
 - 2.5. Scope..... 15**
- 3. USER REQUIREMENTS 16**
 - 3.1. UC-1 – Install and register mobile app..... 16**
 - 3.2. UC-2 – Make settings to registered devices and viewers 18**
 - 3.3. UC-3 – View unrecalled ads 20**
 - 3.4. UC-4 – Manage list of favorite ads 23**
 - 3.5. UC-5 – Import ad metadata 25**
- 4. FUNCTIONAL REQUIREMENTS 27**
 - 4.1. User and Device Management..... 27**
 - FR-1.1 – Display devices and viewers registered in user account 27
 - FR-1.2 – Connect device to user account 27
 - FR-1.3 – Enable/disable device in user account 28
 - FR-1.4 – Assign and remove viewers to/from devices 28
 - FR-1.5 – Register mobile app 28
 - FR-1.6 – Log in from mobile app..... 28
 - FR-1.7 – Log out from mobile app..... 28
 - FR-1.8 – Update user account settings in mobile app..... 29
 - FR-1.9 – Update mobile app settings..... 29

FR-1.10 – Collect Viewer’s ad preferences and send them to cloud database	29
FR-1.11 – Save Viewer’s ad preferences in cloud database.....	29
4.2. Management of Ad Metadata	29
FR-2.1 – Validate and parse a file with ad metadata.....	29
FR-2.2 – Receive ad metadata via API and send responses.....	29
FR-2.3 – Validate and parse ad metadata received via API.....	30
FR-2.4 – Store ad metadata in cloud database.....	30
4.3. Impressions Management.....	30
FR-3.1 – Collect metadata about viewing ad by viewer	30
FR-3.2 – Send metadata about ad impressions to cloud database.....	30
FR-3.3 – Save metadata about ad impressions in cloud database	30
4.4. Working with Unrecalled Ads	31
FR-4.1 – Make request of unrecalled ads	31
FR-4.2 – Return metadata for unrecalled ads	31
FR-4.3 – Save Viewer's request of unrecalled ads	36
FR-4.4 – Display unrecalled ads	36
FR-4.5 – Save results for unrecalled ads in Viewer's history	36
FR-4.6 – Display Viewer's history of unrecalled ads	36
FR-4.7 – Mark/unmark chosen unrecalled ads as Viewer's favorite.....	37
FR-4.8 – Display a list of Viewer's favorite ads	37
FR-4.9 – Open a link of a chosen ad on the mobile app	37
FR-4.10 – Save information about following ad link	37
FR-4.11 – Make a phone call for a chosen ad on the mobile app	37
FR-4.12 – Save information about phone calls from ads	38
FR-4.13 – Share a chosen ad on a social network on the mobile app	38
FR-4.14 – Save information about sharing ads on social networks.....	38
FR-4.15 – Display recent unrecalled ads simultaneously with content.....	39
FR-4.16 – Display recent unrecalled ads without request.....	39
FR-4.17 – Display details for chosen unrecalled ad	39
FR-4.18 – Clear history of unrecalled ads	40

4.5. Reporting	40
FR-5.1 – Display report on actual showing of ads to viewers.....	40
FR-5.2 – Display report on requesting unrecalled ads by viewers	40
FR-5.3 – Display report on showing unrecalled ads to viewers.....	40
FR-5.4 – Display report on clicks by links in ads.....	41
FR-5.5 – Display report on phone calls from ads	41
FR-5.6 – Display report on sharing ads on social networks	41
FR-5.7 – Display funnel report of viewer interaction with ad	42
5. NON-FUNCTIONAL REQUIREMENTS	43
5.1. External Interface Requirements	43
5.1.1. User Interfaces.....	43
5.1.2. Hardware Interfaces	44
5.1.3. Software Interfaces	44
5.1.4. Communications Interfaces.....	46
5.2. Other Non-functional Requirements	47
5.2.1. Performance Requirements	47
5.2.2. Security Requirements	47
5.2.3. Software Quality Attributes.....	48
6. DATA MODELS	50
7. BUSINESS RULES	55

List of Figures

Fig. 1. Data flows between the parties	11
Fig. 2. The solution architecture	12
Fig. 3. Example of unrecalled ads for several devices	32
Fig. 4. Example of unrecalled ads for one device and a smartphone.....	33
Fig. 5. Retrieving ads within the ad search timeframe.....	34
Fig. 6. Conceptual data model	50
Fig. 7. Logical data model	54

1. Preface

1.1. Introduction

Intap, Inc. invented the approach to display “unrecalled” information such as advertisements, video clips, news plots, fragments of shows and films, etc. This is accomplished by associating metadata to a video clip or advertisement and retrieving this metadata according to timestamps of an actual demonstration of the content to viewers.

The technology also helps streaming service providers to build in-house tools for audience measurement and targeting. The data they will collect will be extremely valuable because it can be used for improves TV attribution and yields powerful insights for AI-driven ad personalization which allows advertisers to serve individual ads to each viewer based on viewer audience insights.

1.2. Purpose and Goal of Document

The SRS describes the functionality of the invention as a part of the Service provider's System (the “System”).

The **purpose** of this document is to serve as a resource for the development of the invented functionality of the System.

The **goal** of this document is to describe how the System should work and what requirements it must meet.

1.3. Audience of Document

The intended audience of this document is (1) the Service providers' stakeholders, CEOs, who make decisions on the development of the new functionality, and (2) the development team of the System. This document should be used during all phases of the functionality design and while planning its integration to the existing system.

1.4. Glossary

Advertiser — a business wishing to promote their products or services on a smart TV.

Advertisement (ad) — TV advertisement, also video clips displayed to the Viewer based on previously viewed video content, Viewer's preferences, and other parameters.

Ads listing — ad metadata that are created by advertisers for all aired ads, which include the following information:

- Ad IDs;
- Lengths of the ads (hours, minutes, seconds, milliseconds, etc.);
- Landing page URLs and titles of the listings, phone number, or any other ad-related information, which will be displayed automatically as a companion banner or shown to the Viewer when he/she taps the process activation button.

Device — User's digital video/audio data reproducing device used for watching content from the Service provider. The device can be a smartphone, a tablet, a desktop computer, a notebook, or a smart TV. For service providers that have an option of several viewers' profiles in one user account, such profiles are meant as a device across the document.

Device ID — a distinctive number associated with a device, used for associating devices with a user account and associating viewed ads to devices on which they were shown. Device IDs are unique for all devices registered by the Service provider.

Impression, viewable impression — the fact that an ad was viewable when transmitted (in part, entirely or based on other conditional parameters).

IntapTV — the part of the System described in this document, the invention by Intap, Inc.

Mobile application — here, Service provider's mobile application used for reproducing video content on a Viewer's mobile device and viewing "unrecalled" ads. Also, the mobile application may be used by the Viewer as a remote control while watching content from the Service provider on a smart TV. The mobile application is a part of the System.

Service provider — an owner of a video streaming service that provides video content to users and related viewers; also Service provider is one of the System's user classes.

System — the system under development intended for ad insertion, ad transmission, data collection, and demonstration of "unrecalled" ads. Usually, the Service provider is the System's owner and developer.

Unrecalled advertisement, unrecalled ad — in the context of this document, a piece of visual digital information in the form of digital signals, which a viewer sees while viewing the main content from the Service provider, but for some reason (e.g., a TV ad ended, and the viewer could not write down or memorize its contact details) could not catch or missed a moment to memorize the information. The purpose of the System is to facilitate the search and retrieval of the missed information. Ads that are currently transmitted and watched by the Viewer are also considered unrecalled ads. They are retrieved and listed for the Viewer, at his/her request, together with other recently displayed ads.

User — an individual who concludes a contract with the Service provider for watching video content on a subscription basis or without it. The User is a customer of the Service provider and the main end-user of the System.

User account — a record in the System which is associated with the User and allows the User to connect to the Service provider network for watching content.

Viewer — an individual that can be a User or a person who can watch video content from the Service provider together with or without the User and receive relevant or personalized ads.

Viewer profile — settings related to an individual viewer or several viewers together within a user account. Profiles, if offered by a Service provider, allow precise personalization of the viewer experience, filtering, and tracking content views by different viewers. Viewer profiles are meant as **devices** across the document as they have the same functioning logic.

1.5. References

1. U.S. Patent 10,003,931: “Method for Retrieval of Unrecalled Information from Digital Advertisement.”
2. ATSC: “ATSC Standard: ATSC 3.0 Interactive Content”, Doc. A/344:2017. Advanced Television Systems Committee. 18 December 2018. <https://www.atsc.org/wp-content/uploads/2017/12/A344-2017-Interactive-Content-2.pdf>
3. Guidelines for Identifier for Advertising (IFA) on OTT platforms. IAB Tech Lab. <https://iabtechlab.com/standards/guidelines-identifier-advertising-over-the-top-platforms/>
4. Digital Video Ad Serving Template (VAST). Version 4.2. IAB Tech Lab. Released: April 2019. https://iabtechlab.com/wp-content/uploads/2019/04/VAST_4.2_final_23April2019.pdf
5. Ad-ID | Ad-ID is the industry standard for identifying advertising assets across all media platforms. <http://ad-id.org>.

2. Overall Description

2.1. Purpose of System

The entire System is intended for transmitting video streaming content to viewers with server-side or client-side dynamic insertion of relevant or personalized ads. The **described functionality** is related to retrieving metadata about ads (currently or previously displayed) at the request of viewers, providing details on these ads, and collecting statistics on these processes.

The purpose of this functionality is to:

- show relevant or personalized ads to **viewers**;
- help **viewers** get details on currently and previously viewed ads with options to reach an advertised company quickly, and/or share this information on social networks to discuss them with other people;
- increase viewers' click-through rate (CTR) and revenue of the **service provider**;
- collect statistics of viewers' reactions to ads and provide them in real-time to **advertisers** for making decisions on future ad placement.

2.2. User Classes and Characteristics

It is assumed that the System will have the following user classes: customers, viewers, service provider's users, and advertisers.

- **Customers** — consumers of the service provider's video streaming services. Each customer concludes a contract with a service provider. Customers are the **primary user class**. They regularly work with the System while watching video content and ads, usually several times per week. Some customers do it more often, some (especially those who buy these services for other family members) do it less often.
Main requested functions: user account settings, viewing video content, viewing unrecalled ads. Usually, customers are more technically savvy than viewers. Customers need to access and modify data for their user account and only within it.
- **Viewers** — people who watch video content from the service provider using a user account registered by a service provider's customer. Usually, viewers (who are not customers at the same time) are family members or friends of a customer. It is the **primary user class**. They can use the System regularly (several times per week or even several times per day) or occasionally (e.g., once a week).

Main requested functions: viewing video content, viewing unrecalled ads. Often, viewers are not technically savvy. The Viewer can see details related to their user account and make personal settings (preferences).

- **Service providers** — groups of users related to service providers. This user class is presented by different user roles, from administrators with full permissions to observers with just read-only permissions depending on the organization of a certain service provider. Service providers use the System on a daily basis, as it is their job.

Main requested functions: making System settings, monitoring video transmission processes, analyzing statistics of viewers' interaction with the System (including ads statistics), user support. Usually, most of the service providers' staff is technically savvy. They can access data related to all their customers according to their responsibilities, roles, and permissions.

- **Advertisers** — users related to the companies that wish to show their ads and ads-related metadata via the service provider's video streaming service and a mobile app. Advertisers use the System regularly (daily, or at least several times a week), as it is their job.

Main requested functions: managing ads listings, analyzing statistics of viewing ads, and following links from these ads for targeting viewers with more relevant or personalized ads. Often advertisers' staff is technically savvy. They can see statistics on their ads and cannot access any data about service providers' customers.

2.3. Description of Solution

The described System is going to be built as an extension to an existing system(s) owned by a provider of digital television or OTT/IPTV/VOD and other services.

The System is intended to have the following features:

- Importing and managing ad metadata;
- Retrieving ad metadata on viewers' requests for currently displayed and previously displayed ads;
- Providing details about unrecalled ads to viewers and opening related links (also as companion ads displayed simultaneously with the main content);
- Dialing ads-related phone numbers on viewers' requests;
- Sharing ads-related details on social networks on viewers' requests;
- Collecting statistics on requesting unrecalled ads by viewers, opening links, dialing phone numbers and sharing on social networks;
- Displaying collected statistics.

The data flows between involved parties are represented in Fig. 1.

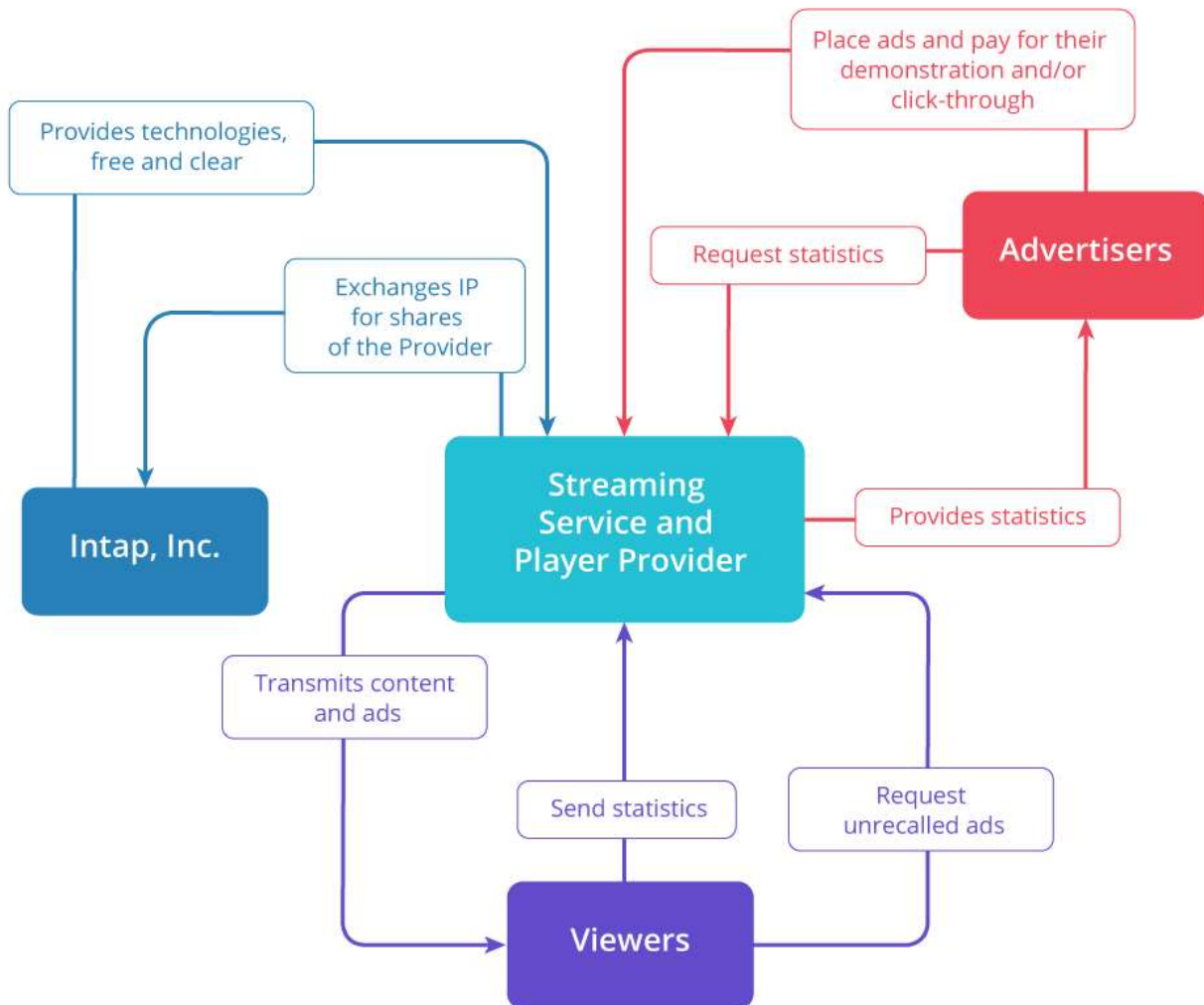


Fig. 1. Data flows between the parties

The reference logical architecture allowing a server-side ad inserter (SSAI) to perform dynamic ad insertion, where the ad decisions are provided from an ad decision service (ADS).

The System and the ad insertion process: Ad metadata (more precisely, really a link to ad-id) is transmitted synchronously when inserting and displaying ads. This information is transmitted through the API and stored in a cloud DB/viewer file with a timestamp insert. For this timestamp, unrecalled ads are fetched.

Besides this logical architecture, other technologies/standards, suitable for implementation of the method of the described System, can be used:

- VAST, VMAP and VPAID;
- MPEG-DASH and HLS;
- ATSC 3.0 and DVB;
- All SCTE.

The solution architecture is displayed in Fig. 2. The main System components to be changed/implemented are depicted in blue. The User-faced components are purple. Internal existing components that are not supposed to be changed are depicted in red. Data flows and their directions are represented by arrows.

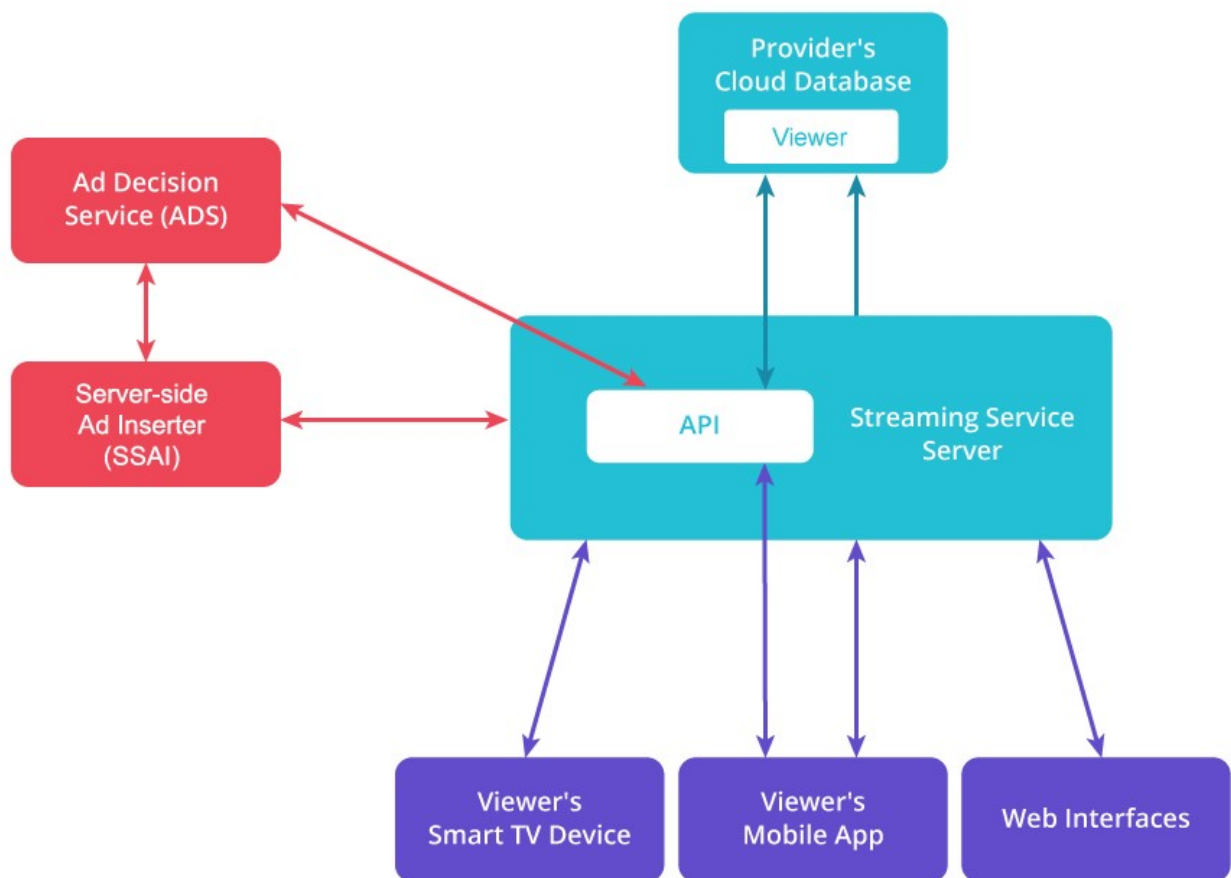


Fig. 2. The solution architecture

- **Provider's Cloud Database** — a database used for storing metadata for ads listings, historical data of actual ads demonstration to each viewer, requesting unrecalled ads, opening links from them, and performing other actions. It is not necessarily a cloud database, which is given as an example.

- **Streaming Service Server** — a server or several servers that control all operations related to video streaming (including ads), management of ad metadata and showing unrecalled ads, collection of historical data, and generating reports/dashboards. The server provides an API for accessing the cloud database. Also, the server provides interfaces for interaction with user devices and web interfaces for different user classes.
- **Ad Decision Service (ADS)** — a System component that determines what advertising content needs to be inserted inside the non-advertising (e.g., entertainment) content. For determining advertising content, this service uses viewers' preferences and viewers' statistics that are requested from the System API.
- **Server-Side Ad Inserter (SSAI)** — a System component that inserts ad predetermined by the ADS in the video content stream.
- **Viewer's mobile app** — a mobile application developed by the Service Provider and installed on a Viewer's mobile device (a smartphone or a tablet). The application can be used by viewers for watching video/audio content, requesting details about unrecalled ads and viewing them. Usually, it also can be used as a software remote control while watching a smart TV.
- **Viewer's smart TV device** — a Viewer's device for reproducing video/audio content without the Service Provider's mobile app installed on it, (i.e., it is not a smartphone or a tablet, but a smart TV, a digital media player or a video game console). While using this kind of device, viewers also use the Service Provider's mobile app on their mobile devices to receive unrecalled ads (with or without a request). Alternatively, it can be a desktop computer or a notebook, but these device types are not typically used by viewers in combination with their mobile devices while viewing content.
- **Web interfaces** — user interfaces available in a web browser to different user classes for their purposes. Each user class has a web interface: customers access their user accounts, advertisers manage ad metadata and view ads-related statistics, service providers manage System settings and view statistics.

2.4. Constraints, Assumptions and Dependencies

Constraints:

- Service provider has an existing system and infrastructure.
- Service provider has its development team(s) and/or uses some outsource services.
- Business process operates with customers' personal data, and this data must be kept secure.
- Business process implies collecting ads statistics, which could be confidential information (trade secret), and thus must be kept secure.

Assumptions:

- Viewers will be interested in details about unrecalled ads, will request them, and will see displayed information (including automatically displayed).
- System is available online for all user classes wherever the Internet is available.
- Viewers download and install the service provider's mobile application, which besides others, includes the feature of viewing unrecalled ads.
- According to a contract and a privacy policy, the Service provider will be able to collect data about users and viewers (what ads they watch, when they request details about unrecalled ads, what links they follow from the received listings of unrecalled ads, etc.).

Dependencies:

- Existing and commonly used standards of digital television (mainly ATSC 3.0, but also DVB and others).
- OTT/IPTV/VOD and others.

2.5. Scope

The scope of the described solution is limited to the functionality of working with ad metadata, showing unrecalled ads, and collecting and displaying related statistics. Thus, development and setting up the following components are included **in the scope**:

- Service provider's Cloud Database;
- Service provider's Server with related functionality:
 - API for providing access to the cloud database;
 - Importing and managing ad metadata through a web user interface;
 - Retrieving and returning unrecalled ads on viewer's requests;
 - Displaying statistical reports and dashboards through a web user interface;
- Mobile app with related functionality:
 - Viewer and device (app) registration;
 - Making settings (including Viewer's ad preferences);
 - Providing details about unrecalled ads and opening related links;
 - Dialing ads-related phone numbers;
 - Sharing ads-related details on social networks.

Out-of-scope for the described solution are:

- Transmitting video/audio content to viewers' reproducing devices;
- Ads insertion into video content (SSAI);
- Algorithms of making decisions for displaying relevant or personalized ads to viewers (ADS);
- Collecting payments from advertisers on the Pay-per-Tap® basis of viewers' statistics.

3. User Requirements

3.1. UC-1 – Install and register mobile app

Actors

Viewer

Description

Viewer installs the mobile app of the Service provider, registers the app in the System and changes settings to be able to view content and related ads.

Trigger

Viewer installs the mobile app from an application store and opens it.

Preconditions

1. Viewer has a contract with the Service provider.
2. Viewer has credentials for the user account.

Postconditions

1. Viewer's mobile app is registered in the user account.
2. Viewer's preference profile is saved.

Basic flow of events

1.0. Registering viewer and mobile app

1. **Viewer** installs the mobile app of the Service provider from an application store and opens it.
2. System requests the Viewer to provide user account credentials ([FR-1.6](#)).
3. **Viewer** provides user account credentials.
4. System authenticates the Viewer in the user account ([FR-1.6](#)) and registers the mobile app in the user account ([FR-1.5](#)).
5. System requests the Viewer to fill out a preferences profile ([FR-1.10](#)).
6. **Viewer** fills out the profile.
7. System sends to the cloud database and saves information about Viewer's preferences ([FR-1.11](#)) and opens the main page of the mobile app, letting the Viewer see content from the Service provider.

Alternative flows

1.1. Logging in on a registered device

If in step 4 of the basic flow the System detects that this device was already authenticated with the provided credentials and the Viewer has a filled preferences profile, then:

1. System opens the main page of the mobile app allowing the Viewer to view content from the Service provider.
2. Use case ends.

1.2. Breaking and resuming registration process

If in step 6 of the basic flow the Viewer interrupted the registration process, then next time when the Viewer opens the application:

1. The use case resumes from step 5.

Exceptions

1.0.E.1. Wrong credentials

If in step 4 of the basic flow the System detects that the provided credentials are wrong, then

1. System displays a related message asking to check and provide correct credentials.
2. Return to step 3 of the basic flow.

Included use cases

None.

Business Rules

None.

Special Requirements

1. Credentials for the user account are provided in the contract or can be updated by another viewer with suitable permissions.

3.2. UC-2 – Make settings to registered devices and viewers

Actors

User, Viewer

Description

User makes settings to registered devices and associated viewers in the user account, System applies them to all devices and viewers.

Trigger

User initiates changing settings to registered devices and viewers.

Preconditions

1. User is authenticated in the user account.
2. At least one device is registered in the user account.
3. At least one viewer is registered in the user account.

Postconditions

1. Updated settings to registered devices and viewers are saved in the user account.

Basic flow of events

2.0. Making setting changes to registered viewers

1. **User** initiates changing settings to registered devices and viewers.
2. System displays the current situation with devices and viewers registered in the user account ([FR-1.1](#)).
3. **User** decides to make changes to registered viewers.
4. System allows User assigning a viewer to one of the registered devices (not assigned yet), reassigning a viewer from one registered device to another, or disconnecting a viewer from all registered devices ([FR-1.4](#)).
5. **User** makes desired setting changes.
6. System saves the settings.

Alternative flows

2.1. Making setting changes to registered devices

If in step 3 of the basic flow, the User decides to make changes to registered devices, then:

1. System allows the User to disable a chosen device in the user account ([FR-1.3](#)).
2. The use case resumes from step 5.

Exceptions

2.0.E.1. Reassigning viewer from currently viewed device

If in step 4 of the basic flow, the System detects that a viewer is being reassigned from a device on which he/she is currently viewing content, then:

1. System notifies the User about this and requests confirmation.
2. **User** confirms reassigning the viewer from the devices.
3. System displays a notification on the device currently being watched by the Viewer.
4. The use case resumes from step 6.

2.1.E.1. Disabling device while viewing content from it

If in step 1 of the alternative flow 2.1, the System detects that content is currently viewed on the device that the User wishes to disable, then:

1. System notifies the User about this and requests confirmation.
2. **User** confirms disabling the device.
3. System displays a notification on the device currently watched by the Viewer.
4. The use case resumes from step 6.

Included use cases

None.

Business Rules

None.

Special Requirements

1. User cannot reassign himself/herself as a viewer from a device from which the Viewer is modifying the settings.
2. It must not be possible to remove all viewers from a device.
3. It must not be possible to delete/disable a device from which the User is modifying the settings.

3.3. UC-3 – View unrecalled ads

Actors

Viewer, Social network

Description

Viewer requests unrecalled ads. System displays a list of ads currently or recently displayed on TV and Viewer chooses one to see details on.

Trigger

Viewer wants to view unrecalled ads and requests them or sees an ad and wants to get more details about it.

Preconditions

1. Viewer is authenticated by the System.
2. Ads data exists in the database.

Postconditions

1. At least one unrecalled ad is displayed to the Viewer.
2. The landing page for a chosen ad is shown to the Viewer.
3. Data about Viewer's request, returned ads, and following links are saved.

Basic flow of events

3.0. Requesting recent unrecalled ads

1. **Viewer** wants to view unrecalled ads and requests them.
2. System makes a call to the cloud database ([FR-4.1](#)) and saves information about the Viewer's request ([FR-4.3](#)).
3. System returns metadata about unrecalled ads for the Viewer ([FR-4.2](#)), saves this metadata in the history for the Viewer ([FR-4.5](#)) and displays a list of these ads to the Viewer ([FR-4.4](#)).
4. **Viewer** reviews the list of unrecalled ads and chooses one to see its details.
5. System displays details of the chosen ad ([FR-4.17](#)), saves information about choosing this ad ([FR-4.10](#)), and allows the Viewer to use available functions for the ad.
6. **Viewer** decides to open a website link for the ad to get more details.
7. System opens the link to a landing page for the chosen ad ([FR-4.9](#)) and saves data about opening this link by the Viewer ([FR-4.10](#)).

Alternative flows

3.1. Calling by phone for the chosen ad

If in step 6 of the basic flow, the Viewer chooses to call by phone for an ad, then:

1. System dials a phone number from the chosen ad ([FR-4.11](#)) and saves data about the call made by the Viewer ([FR-4.12](#)).
2. The use case ends.

3.2. Sharing chosen ad on social networks

If in step 6 of the basic flow, the Viewer chooses to share an ad on one of the supported social networks, then:

1. **Viewer** chooses one of the supported social networks ([FR-4.13](#)).
2. System initiates sharing the chosen ad in a chosen social network, displays a post preview ([FR-4.13](#)), and saves data about sharing the ad on social network ([FR-4.14](#)).
3. **Viewer** views the preview and confirms publishing the post.
4. **Social network** publishes the confirmed post.
5. The use case ends.

3.3. Viewing unrecalled ads from history

If in step 1 of the basic flow the Viewer wants to see unrecalled ads from the history of previous requests, then:

1. System displays a list of previous unrecalled ads ([FR-4.6](#)).
2. The use case resumes from step 4 of the basic flow.

3.4. Viewing unrecalled ads from a list of favorites

If in step 1 of the basic flow the Viewer wants to see a list of previously viewed unrecalled ads marked as favorites, then:

1. System displays a list of favorites ([FR-4.8](#)).
2. The use case resumes from step 4 of the basic flow.

3.5. Viewing automatically shown recent unrecalled ad

If while the Viewer is watching content in the Service provider's mobile app the System shows an ad, then:

1. System automatically shows the recent unrecalled ad simultaneously with the main content ([FR-4.15](#)).

2. **Viewer** decides to get more details about the ad.
3. The use case resumes from step 5 of the basic flow.

3.6. Viewing automatically shown recent unrecalled ad while watching smart TV

If while showing an ad to the Viewer on a smart TV the System detects that the Viewer opened the mobile app (and is not watching content on it), then:

1. System makes a call to the cloud database ([FR-4.1](#)) and saves information about the Viewer's request ([FR-4.3](#)).
2. System returns metadata about the recent unrecalled ad for the Viewer ([FR-4.2](#)), saves this metadata in the history for the Viewer ([FR-4.5](#)), and shows the recent unrecalled ad in the Viewer's mobile app ([FR-4.16](#)).
3. **Viewer** decides to get more details about the ad.
4. The use case resumes from step 5 of the basic flow.

Exceptions

3.0.E.1. Viewer has no recent impressions

If in step 3 of the basic flow the System detects that the Viewer has not viewed any ads recently ([ad search timeframe](#) defined in [FR-4.2](#)), then:

1. System displays a message explaining that ads will be shown after the Viewer watches ads.
2. System terminates the use case.

Included use cases

None.

Business Rules

[BR-4](#)

Special Requirements

1. While showing ads on a smart TV, a notice should be displayed that ad details can be viewed in the Service provider's mobile app.

Notes

1. Different types of unrecalled ads (a list of requested unrecalled ads, automatically shown simultaneously with other content or shown while watching a smart TV) can be displayed in different or same areas of the mobile app depending on its UI design.

2. A viewer that is watching TV together with other viewers on a device assigned to multiple viewers and is using the mobile app as a remote control for this TV (or requesting unrecalled ads) may have priority for showing relevant or personalized ads.

3.4. UC-4 – Manage list of favorite ads

Actors

Viewer

Description

Viewer views a list of ads (unrecalled, favorites, or history), marks/unmarks them as favorites to update the list of favorite ads so the Viewer can find them in the future.

Trigger

Viewer opens a list of unrecalled ads, the history of unrecalled ads, or the list of favorite ads.

Preconditions

1. Viewer is authenticated by the System.

Postconditions

1. Changes to favorite marks of ads are saved.
2. A list of favorite ads is displayed to the Viewer.

Basic flow of events

4.0. Marking an ad as a favorite in the list of unrecalled ads

1. **Viewer** wants to view unrecalled ads and requests them.
2. Use case [UC-3](#) is invoked.
3. **Viewer** chooses an ad from the list and wants to mark it as a favorite.
4. System marks the chosen ad as a favorite ([FR-4.7](#)).
5. **Viewer** opens the list of favorite ads.
6. System displays a list of ads marked as favorites ([FR-4.8](#)).
7. **Viewer** sees the ad previously marked as favorite.

Alternative flows

4.1. Marking an ad as a favorite in the history

If in step 1 of the basic flow the Viewer wants to view the history of ads, then:

1. System displays a list of previously displayed unrecalled ads ([FR-4.6](#)).
2. The use case resumes from step 3.

4.2. Removing ads from the list of favorites

If in step 1 of the basic flow, the Viewer wants to view the list of favorite ads, then:

1. System displays a list of ads marked as favorites ([FR-4.8](#)).
2. **Viewer** chooses an ad to remove from favorites.
3. System removes the chosen ad from the list of favorites ([FR-4.7](#)) and updates the list.

Exceptions

4.1.E.1. History is empty

If in step 1 of the alternative flow 4.1, the System detects that for the Viewer the history of unrecalled ads is empty, then:

1. System displays a message about this and explains how ads appear in the history.
2. System terminates the use case.

4.2.E.1. Favorites are empty

If in step 1 of the alternative flow 4.2, the System detects that for the Viewer the list of favorite ads is empty, then:

1. System displays a message about this and about how the Viewer can mark ads as a favorite to display on this list.
2. System terminates the use case.

Included use cases

[UC-3 – View unrecalled ads](#)

Business Rules

[BR-4](#)

Special Requirements

None.

3.5. UC-5 – Import ad metadata

Actors

Advertiser, Advertiser's system

Description

Advertiser prepares and imports ad metadata into the System which stores it and then uses it to display ads to viewers.

Trigger

Advertiser prepares ad metadata in a predefined format and initiates its import to the System.

Preconditions

1. Advertiser knows the structure of the accepted data format (or has a file template).

Postconditions

1. Ad metadata is imported/updated in the System database.

Basic flow of events

5.0. Importing ad metadata from file

1. **Advertiser** prepares ad metadata in a predefined format and initiates its import to the System.
2. System requests to provide a file with ad metadata to import.
3. **Advertiser** provides a file.
4. System validates the provided file and parses it ([FR-2.1](#)).
5. System stores AD metadata ([FR-2.4](#)) and notifies the Advertiser about successful import.

Alternative flows

5.1. Importing AD metadata via API

If in step 1 of the basic flow, the Advertiser initiates import of ad metadata from its system to the System API, then:

1. **Advertiser's system** sends ad metadata to the System.
2. System accepts an API call, receives ([FR-2.2](#)), validates, and parses ad metadata ([FR-2.3](#)).

3. System stores ad metadata ([FR-2.4](#)) and sends a response to the Advertiser's system ([FR-2.2](#)).

Exceptions

5.0.E.1. Wrong data format in a file

If in step 4 of the basic flow, the System detects that the provided file has a wrong format, then:

1. System notifies the Advertiser, indicating an error type and eventual position in the file ([FR-2.3](#)).
2. System terminates the use case.

5.1.E.1. Wrong data format via API

If in step 2 of the basic flow, the System detects that the received ad metadata is in a wrong format, then:

1. System sends a corresponding response to the Advertiser's system ([FR-2.2](#)).
2. System terminates the use case.

Included use cases

None.

Business Rules

[BR-1](#)

Special Requirements

1. After importing, the Advertiser can update ad metadata or delete it, in the event that it is not relevant anymore. In this scenario, all statistical data is retained.

4. Functional Requirements

4.1. User and Device Management

FR-1.1 – Display devices and viewers registered in user account

System shall display lists of [devices](#) and [viewers](#) registered in the [user account](#). These lists must show relations of viewers to certain devices or show some viewers as registered, but not assigned to any device.

Example

Device: John's Smartphone

Viewer: John

Device: Mary's Tablet

Viewer: Mary

Device: Smart TV

Viewer: John

Viewer: Mary

Viewer: Alice

No device

Viewer: Bob

FR-1.2 – Connect device to user account

System shall be able to register a new [device](#) in a [user account](#) while logging into the account from this device ([FR-1.6](#)). After that, the device must appear in the list of devices ([FR-1.1](#)).

A device can be connected only to **one user account** to which it is logged in. After logging in to another user account, the device becomes assigned to this account and is automatically disabled in the previous one ([FR-1.3](#)).

FR-1.3 – Enable/disable device in user account

System shall be able to disable a [device](#) previously registered in a [user account](#). After that, the device must be displayed as disabled in the list of devices ([FR-1.1](#)).

System shall be able to enable a previously disabled device in a user account if this device has been logged in to this user account.

Both enabling and disabling devices must be available to a user in user account settings.

FR-1.4 – Assign and remove viewers to/from devices

System shall be able to assign [viewers](#) to [devices](#) and remove [viewers](#) from [devices](#). A viewer can be assigned to several devices, or not assigned to any device. Thus, while removing a viewer from a device, this viewer is reassigned to another device or is not assigned to any device.

FR-1.5 – Register mobile app

System shall register each instance of the mobile app while logging in to a [user account](#) from this app.

With that, a new [device](#) (on which the mobile app is installed) must be created within the user account. A new [viewer](#) assigned to this device must be created too.

By default, a [device name](#) is filled with a current device name from a device operating system. Later, it can be modified by the Viewer in the mobile app settings ([FR-1.9](#)).

[Device id](#) with which the device is registered in the user account must be unique and generated based on the device hardware ([HI-1](#)). It is recommended to follow the Guidelines for IFA [3] to standardize the identifiers.

FR-1.6 – Log in from mobile app

System shall allow logging into a [user account](#) from the mobile app.

Permissions depending on a role are described in [SE-3](#).

FR-1.7 – Log out from mobile app

System shall allow logging out from a [user account](#) in the mobile app. After logging out, the Viewer cannot watch any content, and the System does not track activity concerning a particular [viewer](#).

FR-1.8 – Update user account settings in mobile app

System shall allow updating settings of a [user account](#) from the mobile app:

- [Email](#)
- [Password](#)
- [Name](#)

FR-1.9 – Update mobile app settings

System shall allow updating settings of a certain instance of the mobile app:

- [Device name](#)
- Consent for data collection ([SE-4](#))
- Ad preferences ([FR-1.10](#))

FR-1.10 – Collect Viewer’s ad preferences and send them to cloud database

System shall be able to request and collect the Viewer’s [ad preferences](#) in the form of a questionnaire, and after the Viewer’s submission, send them to the cloud database ([FR-1.11](#)). This should work on all devices where the Viewer can log in to a user account ([FR-1.6](#)).

FR-1.11 – Save Viewer’s ad preferences in cloud database

System shall be able to save in the cloud database [ad preferences](#) expressed by the Viewer ([FR-1.10](#)).

4.2. Management of Ad Metadata

FR-2.1 – Validate and parse a file with ad metadata

System shall be able to validate and parse an uploaded file with [ad metadata](#).

Results of validation and parsing must be displayed to a user. Conditions to validate are listed in [FR-2.3](#). Additionally, parsing errors should be made available to the user, where each error displays details according to an expected file format. [SQA-3](#) defines the file format(s).

FR-2.2 – Receive ad metadata via API and send responses

System shall be able to receive [ad metadata](#) via its own API.

In response to the calls to this API, the System must send results of data processing ([FR-2.3](#)).

FR-2.3 – Validate and parse ad metadata received via API

System shall be able to validate and parse [ad metadata](#) received via the API ([FR-2.2](#)). The following conditions are possible:

- Data is correct and saved;
- Some data is missing;
- Some data has incorrect data type/format;
- Data has conflicts with previous data.

FR-2.4 – Store ad metadata in cloud database

System shall be able to store in the cloud database [ad metadata](#) received and parsed from a data source ([FR-2.1](#), [FR-2.3](#)).

4.3. Impressions Management

FR-3.1 – Collect metadata about viewing ad by viewer

System shall be able to collect [metadata](#) about ad impressions on a viewer's device to send it after that to the cloud database ([FR-3.2](#)). This is related to broadcasted ads, relevant or personalized dynamically inserted ads.

The following metrics can be collected:

- Ad is viewed fully or in part;
- How many seconds from the beginning of the ad were viewed, if the ad was viewed in part;
- Which action the viewer made (skipped, paused, canceled, etc.), if the ad was viewed in part.

FR-3.2 – Send metadata about ad impressions to cloud database

System shall be able to send [metadata](#) about ad impressions from a viewer's device to the System's cloud database. Sending must be triggered for each ad that has been started and ended.

FR-3.3 – Save metadata about ad impressions in cloud database

System shall be able to store in the System's cloud database [metadata](#) about ad impressions from a certain viewer on his/her device ([FR-3.1](#), [FR-3.2](#)). If the metadata

about a certain impression already exists in the database, then it must be updated with the newly received one.

4.4. Working with Unrecalled Ads

FR-4.1 – Make request of unrecalled ads

System shall make a request from the mobile app to the cloud database to get unrecalled ads when the Viewer wants to see them. The following parameters must be passed in the request:

- [viewer](#)
- [device](#) from which the request is being sent.

FR-4.2 – Return metadata for unrecalled ads

System shall return [ad metadata](#) for the ads recently viewed by a [Viewer](#) in response to a request from the Viewer's mobile app ([FR-4.1](#)).

When a Viewer assigned to several devices is viewing ads on a personal [device](#) (to which only this Viewer is assigned), and the request was sent from this device, then only unrecalled ads shown on this particular device must be included in the response. Otherwise, the System must equally consider all devices owned by the Viewer while retrieving unrecalled ads.

Thus, this should be a two-step process:

1. System must retrieve unrecalled ads shown on the [device](#) from which the request was received.
2. If no unrecalled ads were retrieved on step 1, then the System retrieves them for all devices assigned to the [Viewer](#).

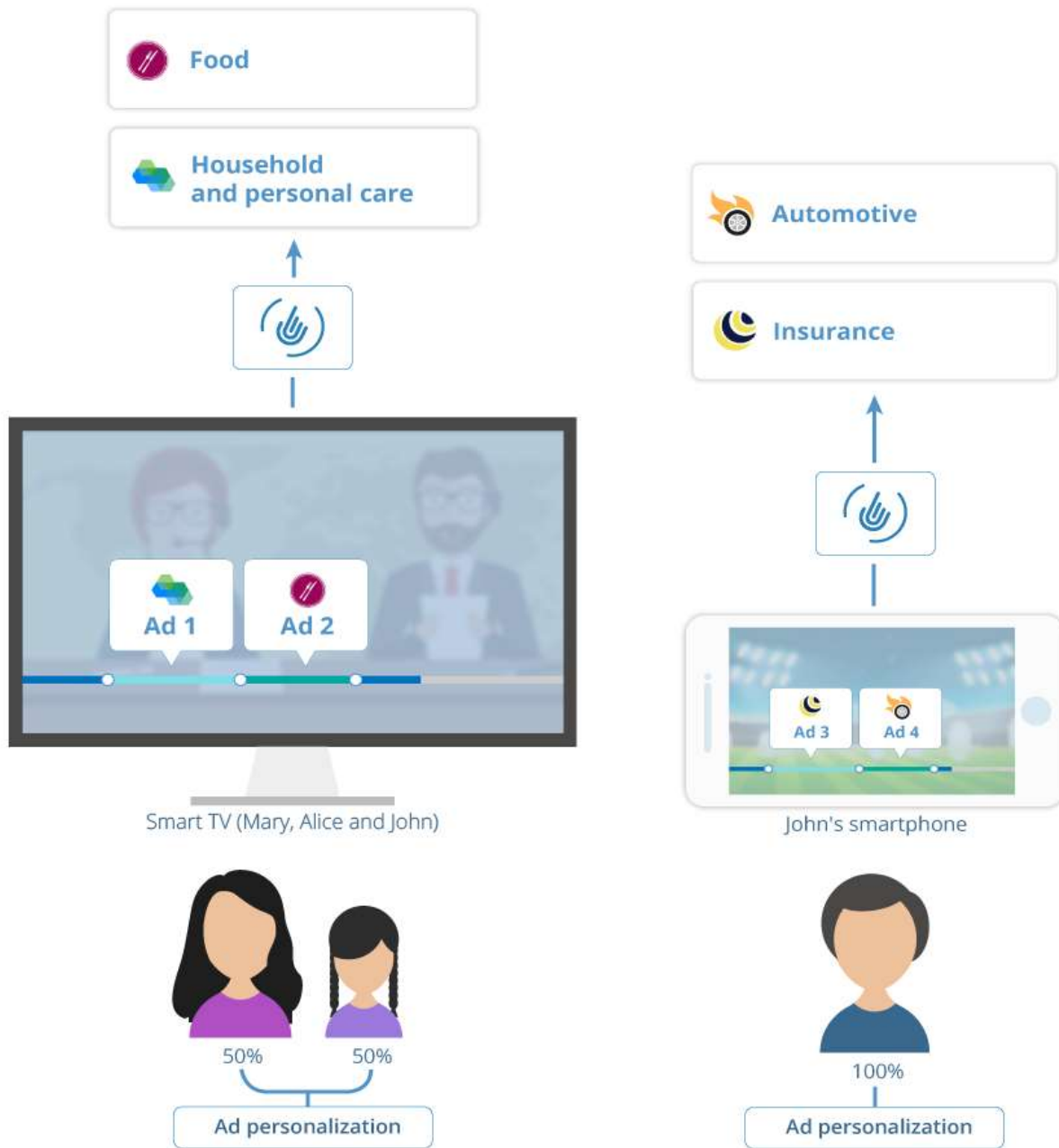


Fig. 3. Example of unrecalled ads for several devices

Example

It can be shown using the example from [FR-1.1](#). Assume that John viewed the ads Ad 3 and Ad 4 on his smartphone, while other family members, Mary and Alice, viewed the ads Ad 1 and Ad 2 on the smart TV (Fig. 3).

1. In this situation, after requesting unrecalled ads on his smartphone, John will get Ad 3 and Ad 4, but not the ads displayed on the TV.
2. If John did not see Ad 3 and Ad 4 on his smartphone recently (but watched video content simultaneously with other family members who viewed Ad1 and Ad2), he would get Ad 1 and Ad 2 in the list of unrecalled ads (Fig. 4).

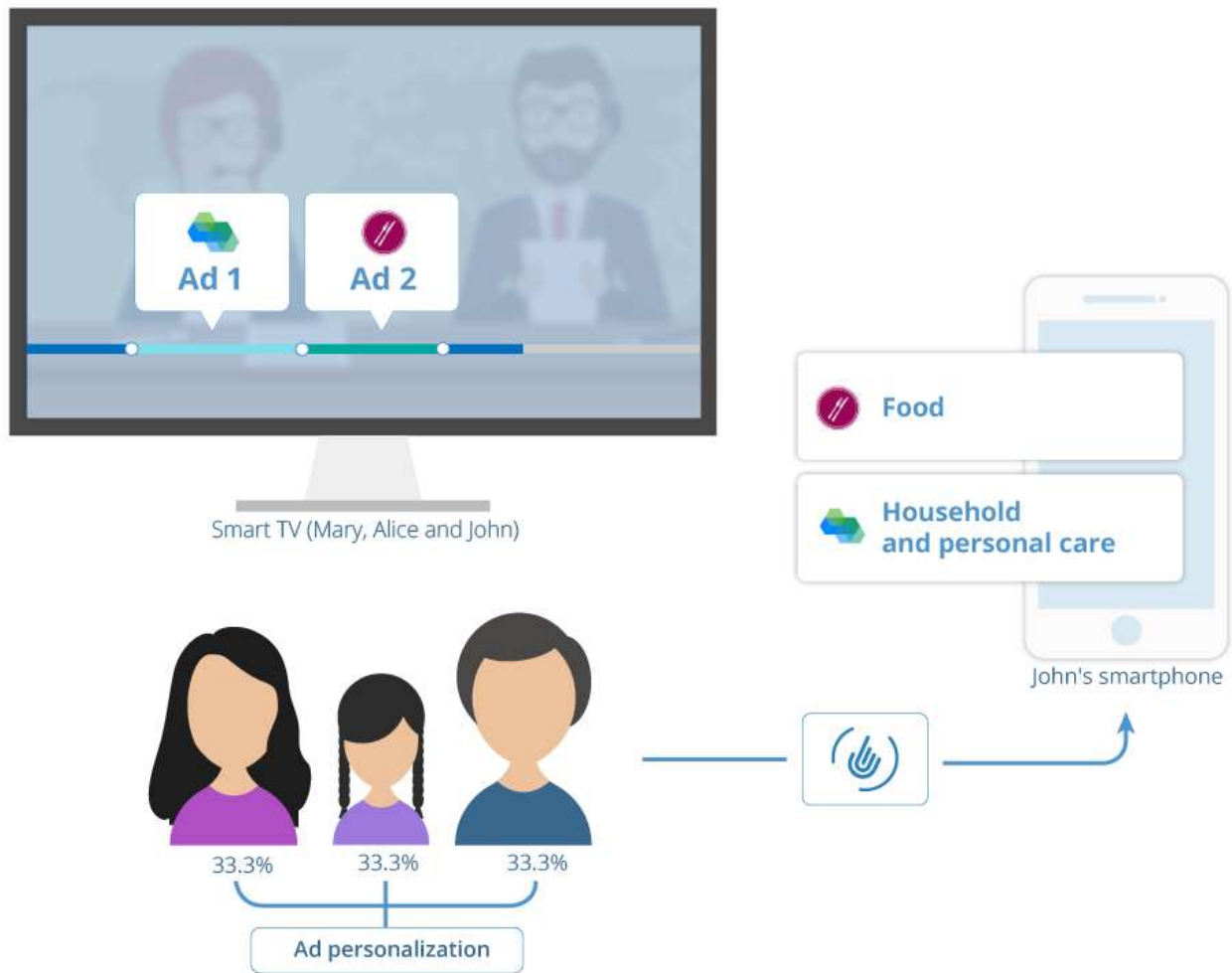


Fig. 4. Example of unrecalled ads for one device and a smartphone

The System's response should be determined by one of the methods that will be described next. The method is chosen by the Service provider for processing all requests or chosen conditionally based on some predefined parameters, such as type of content, time of day, day of the week, etc.

I. Time-bound method

This method retrieves unrecalled ads viewed within a predefined [ad search timeframe](#), counted back from the time when the request has been received. All viewed ads that started and/or ended within this timeframe are included in the list of unrecalled ads.

Therefore:

- An ad being viewed at the moment of receiving the request (if any) is included in the response.
- All ads that started during the ad search timeframe are included in the response.
- An ad that started before the beginning of the ad search timeframe, and ended within this timeframe, is included in the response.

Example

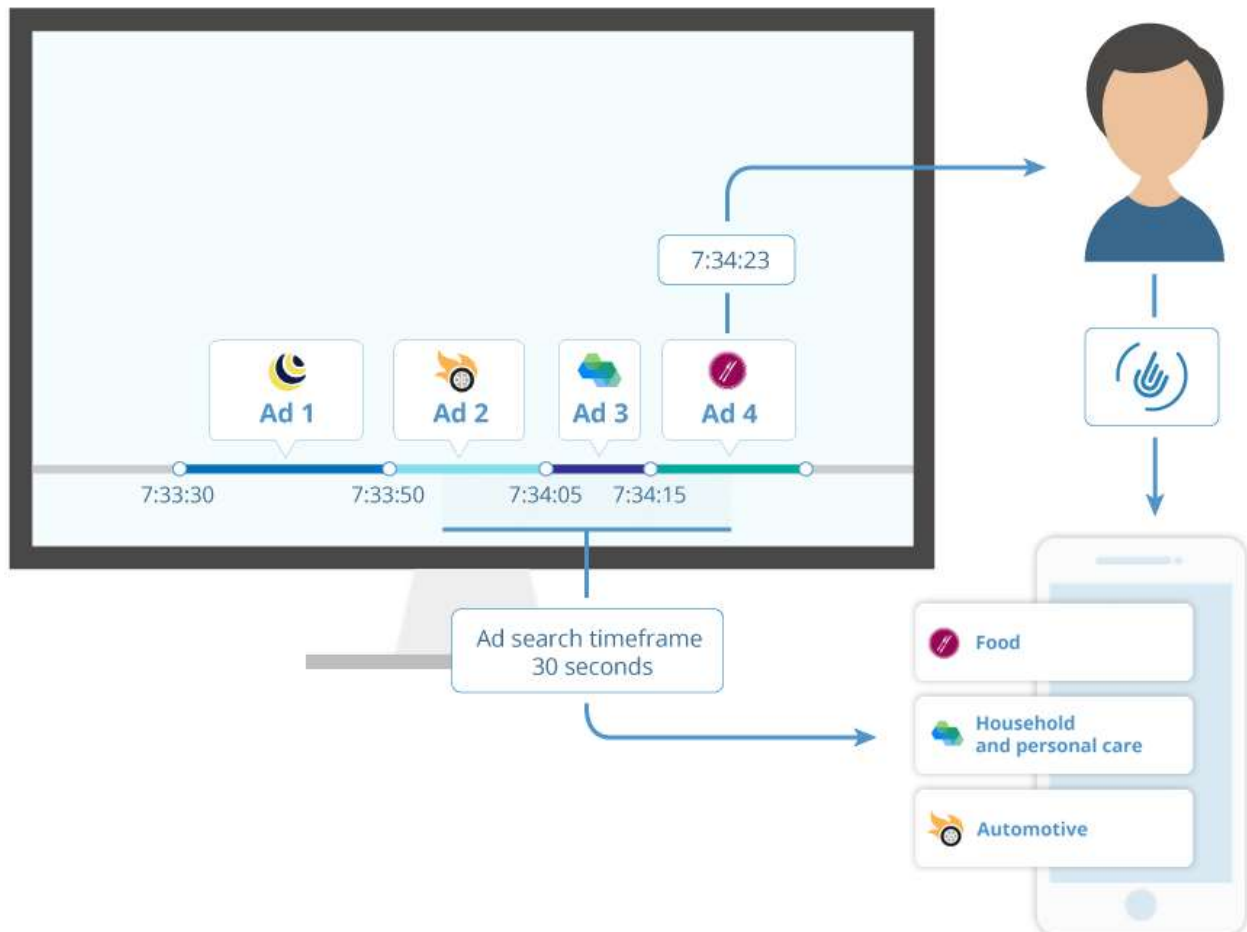


Fig. 5. Retrieving ads within the ad search timeframe

Assume that in the recent time a viewer has seen the ads listed in the table:

Advertisement	Start timestamp	Duration
Ad 1	06:01:2019:07:33:30	20 seconds
Ad 2	06:01:2019:07:33:50	15 seconds
Ad 3	06:01:2019:07:34:05	10 seconds
Ad 4	06:01:2019:07:34:15	15 seconds

Ad search timeframe is 30 seconds.

The request for unrecalled ads is received at 06:01:2019:07:34:23.

Timestamps are represented as MM:DD:YYYY:HH:MM:SS, where MM designates a month, DD designates a day, YYYY designates a year, HH designates an hour in 24-hour format, MM designates a minute, and SS designates a second.

As a result, the response will include Ad 2, Ad 3, and Ad 4 because Ad 1 is out of the ad search timeframe.

II. Volume-bound method

This method retrieves a certain number of unrecalled ads relative to the time when the request was received per the predefined volume constraint — [number of unrecalled ads](#). The response will contain a number of unrecalled ads equal to the predefined volume constraint or the number of ads viewed, whichever is fewer.

E.g., during a recent time (an hour or so), a viewer has seen four ads (Ad 1, Ad 2, Ad 3, and Ad 4, displayed in this order). If the number of unrecalled ads is specified as 3, then the response will return Ad 2, Ad 3, and Ad 4 as the three most recent ads.

III. Combined method

This method combines the two methods described above. Unrecalled ads are retrieved based on ads viewed within a predefined [ad search timeframe](#), counted back from the time when the request was received, but no more than a predefined [number of unrecalled ads](#).

E.g., if the number of unrecalled ads is set to 2, then in the example of the first method description, the response would contain only Ad 3 and Ad 4.

Ad metadata in the response must be ordered in reverse chronological order by the [start time](#) of corresponding viewed ads.

If no ads meet the criteria, the System must return a response with no [ad metadata](#).

FR-4.3 – Save Viewer's request of unrecalled ads

System shall save each Viewer's request for unrecalled ads when receiving it from the Viewer's mobile app ([FR-4.1](#)). The request must be saved with its receipt [timestamp](#) and also must be associated with a response if it is not empty ([FR-4.2](#)). This data can be used in the future for analysis of the Viewer's interests.

FR-4.4 – Display unrecalled ads

System shall display [unrecalled ads](#) in the Viewer's mobile app after getting a response ([FR-4.1](#), [FR-4.2](#)).

A list of unrecalled ads may include one or several ads. If an empty response is received, it should be shown in a message stating that unrecalled ads will be displayed after viewing ads inserted into video content from the service provider.

For each displayed unrecalled ad, it must be possible to view its details and use operations available for this ad ([FR-4.17](#)).

FR-4.5 – Save results for unrecalled ads in Viewer's history

System shall be able to save in the Viewer's history results of [unrecalled ads](#) returned in response to the Viewer's requests ([FR-4.2](#)). Metadata of unrecalled ads are saved in the Service provider's cloud database and can be retrieved when the Viewer requests his/her history ([FR-4.6](#)).

FR-4.6 – Display Viewer's history of unrecalled ads

System shall display to the Viewer the history of [unrecalled ads](#) previously requested and received by the Viewer ([FR-4.5](#)). Past unrecalled ads must be listed in reverse chronological order (recently requested and received ads at the top). Each individually received list of unrecalled ads must be annotated with a date and time of receipt (timestamp of [unrecalled ads](#)). Information about the same ad received at different times is displayed as separate instances under a different date/time.

From the history of unrecalled ads, it must be possible for the Viewer to do the same operations that were possible in the list of recently received unrecalled ads ([FR-4.4](#)).

If the history is empty because the Viewer has not got any unrecalled ads or cleared the history ([FR-4.18](#)), then a message must be displayed. It must explain that ads will appear after requesting and receiving unrecalled ads.

FR-4.7 – Mark/unmark chosen unrecalled ads as Viewer's favorite

System shall allow marking a chosen [unrecalled ad](#) as a Viewer's favorite. Such an ad can be chosen either from a recently received list of unrecalled ads ([FR-4.4](#)) or from the Viewer's history ([FR-4.6](#)).

Also, System shall allow unmarking a chosen [unrecalled ad](#) as a Viewer's favorite. Such an ad, previously marked as a favorite, can be chosen from a list of the Viewer's favorite ads ([FR-4.8](#)), a recently received list of unrecalled ads ([FR-4.4](#)), or from the Viewer's history ([FR-4.6](#)). After unmarking an ad in the list of the Viewer's favorite ads, it must disappear from this list.

If an ad earlier marked as favorite appears again in the list of unrecalled ads, or appears several times in the Viewer's history, all these instances must be displayed as favorite.

FR-4.8 – Display a list of Viewer's favorite ads

System shall be able to display a list of unrecalled ads marked by the Viewer as a favorite ([FR-4.7](#)). If the Viewer does not have any favorite ads, the System must display an empty list with a description of how to add ads into this list.

FR-4.9 – Open a link of a chosen ad on the mobile app

System shall allow opening a link to a [landing page](#) of an unrecalled ad chosen by the Viewer in the mobile app. The link must be opened in a new tab of a mobile web browser.

FR-4.10 – Save information about following ad link

System shall be able to save information when the Viewer opens a link to a landing page related to a chosen unrecalled ad ([FR-4.9](#)). The following data must be sent from the mobile app to the service provider's server and saved in the database:

- [viewer](#) (id)
- [device](#) (on which the link has been opened)
- [ad id](#)
- [timestamp](#)

FR-4.11 – Make a phone call for a chosen ad on the mobile app

System shall allow dialing a [phone number](#) related to an unrecalled ad chosen by the Viewer in the mobile app. Dialing must be performed using the default dialer app on a Viewer's mobile device (including VoIP apps). If no app is set as default, then the mobile app must request which app should be used for dialing.

FR-4.12 – Save information about phone calls from ads

System shall be able to save information when the Viewer makes a phone call to the phone number related to a chosen unrecalled ad ([FR-4.11](#)). The following data must be sent from the mobile app to the service provider's server and saved in the database:

- [viewer](#) (id)
- [device](#) (from which the Viewer has made the call)
- [ad id](#)
- [timestamp](#)

FR-4.13 – Share a chosen ad on a social network on the mobile app

System shall allow sharing an unrecalled ad chosen by the Viewer in the mobile app on one of the supported social networks (according to a Viewer's choice):

- [Facebook](#)
- [Instagram](#)
- [Twitter](#)

When the Viewer initiates sharing on a chosen social network, a standard sharing dialog of the chosen social network must be displayed, requesting credentials (if needed), additional details, and a confirmation before sharing.

Available information from the ad metadata must be passed to a social network for sharing:

- [ad title](#)
- [ad description](#)
- [ad image](#)

FR-4.14 – Save information about sharing ads on social networks

System shall be able to save information about sharing a chosen unrecalled ad on a chosen social network ([FR-4.13](#)). The following data must be sent from the mobile app to the Service provider's server and saved in the database:

- chosen social network
- [viewer](#) (id)
- [device](#) (from which the ad has been shared)
- [ad id](#)
- [timestamp](#) of a moment when sharing the ad has been initiated

Data must be saved for all ads for which sharing was initiated (not necessarily that the ad was shared successfully).

FR-4.15 – Display recent unrecalled ads simultaneously with content

System shall be able to display unrecalled ads simultaneously with other non-advertising content. Such ads must be determined according to [FR-4.2](#). This also concerns displaying information about the currently displayed video ad when an ad's content is currently displayed. Such type of unrecalled ads can be compared to companion ads according to [4] (section 3.13, Companion as Concurrent Display Ad).

An unrecalled ad displayed simultaneously with other content can be shown in any of the following forms:

- a textual link formed of an [ad title](#) (and with displaying an [ad description](#));
- a graphical banner formed of an [ad image](#);
- eventual combined options.

Upon following the link or the banner, a related [landing page](#) must be opened ([FR-4.9](#)). It must also be possible to dial a related [phone number](#) (if it is specified for the ad, [FR-4.11](#)) and share it on a social network ([FR-4.13](#)).

FR-4.16 – Display recent unrecalled ads without request

System shall be able to display unrecalled ads that the Viewer has watched and/or is currently watching on a smart TV associated with the Viewer as a device, without a Viewer's request on opening the mobile application. The unrecalled ads must be displayed in the list ([FR-4.4](#)) according to [FR-4.2](#), and it must be possible to view ads details ([FR-4.17](#)) in the same way as when they are displayed on Viewer's request.

FR-4.17 – Display details for chosen unrecalled ad

System must display details for a chosen unrecalled ad in its detailed view:

- [ad title](#)
- [ad description](#)
- [ad image](#)

Also, one or several of the following operations must be available to the Viewer depending on ad-related information:

- Open a related landing page ([FR-4.9](#))
- Dial a related phone number ([FR-4.11](#))
- Share the ad on a social network ([FR-4.13](#))

In the event that neither a landing page nor a phone number is for an ad, it must always be possible to share it on social networks.

FR-4.18 – Clear history of unrecalled ads

System shall be able, on Viewer's request, to clear the Viewer's history of unrecalled ads ([FR-4.6](#)). After clearing the history, received unrecalled ads continue to be saved in it ([FR-4.5](#)).

4.5. Reporting

This section contains requirements for sample reports, which can be generated on the basis of data collected by the System. Final implementation (including report formats, exact filtering parameters, and eventual breakdowns) depends on the needs of a certain Service provider and its advertisers.

FR-5.1 – Display report on actual showing of ads to viewers

System shall be able to display a report on how often a chosen video ad was actually shown to viewers. The report must represent the number of views over time. The following parameters must be supported for report filtration:

- Interval — start date and end date
- Ad ([ad id](#))
- Viewers' [preferences](#)
- Viewers' region
- Device type (smartphone, tablet, desktop, smart TV)
- Device OS
- Impression type (partial/full)

FR-5.2 – Display report on requesting unrecalled ads by viewers

System shall be able to display a report on how many viewers requested unrecalled ads during viewing a certain video ad or [ad search timeframe](#) after it ended. The report must represent the number of requests over time. The following parameters must be supported for report filtration:

- Interval — start date and end date
- Ad ([ad id](#))
- Viewers' [preferences](#)
- Viewers' region

FR-5.3 – Display report on showing unrecalled ads to viewers

System shall be able to display a report on how a certain ad was shown to viewers as unrecalled. The report must represent the number of views of a chosen ad over time.

Also, the percentage of the total number of video ad views must be indicated. The following parameters must be supported for report filtration:

- Interval — start date and end date
- Ad ([ad id](#))
- Viewers' [preferences](#)
- Viewers' region

FR-5.4 – Display report on clicks by links in ads

System shall be able to display a report on how viewers clicked links from unrecalled ads. The report must represent the number of clicks by a link of a chosen unrecalled ad over time. The percentage of the total number of unrecalled ad views must be indicated also. The following parameters must be supported for report filtration:

- Interval — start date and end date
- Ad ([ad id](#))
- Viewers' [preferences](#)
- Viewers' region

FR-5.5 – Display report on phone calls from ads

System shall be able to display a report on how viewers called by phone numbers from unrecalled ads. The report must represent the number of phone calls from a chosen unrecalled ad over time. The percentage of the total number of unrecalled ad views must be indicated. The following parameters must be supported for report filtration:

- Interval — start date and end date
- Ad ([ad id](#))
- Viewers' [preferences](#)
- Viewers' region

FR-5.6 – Display report on sharing ads on social networks

System shall be able to display a report on how viewers shared unrecalled ads on social networks. The report must represent the number of successful shares of a chosen unrecalled ad in the course of time. The percentage of the total number of unrecalled ad views must be indicated. The following parameters must be supported for report filtration:

- Interval — start date and end date
- Ad ([ad id](#))
- Viewers' [preferences](#)
- Viewers' region

- Social network

FR-5.7 – Display funnel report of viewer interaction with ad

System shall display a funnel report for viewer interaction with a chosen ad by the following stages:

- Viewed a video ad
- Viewed an unrecalled ad
- Interacted with an unrecalled ad (opened ad details, opened a landing page, made a phone call, shared on a social network)

For each stage the System must display a total number of viewers, and their percentage relative to the first stage, which is 100%.

The following parameters must be supported for report filtration:

- Interval — start date and end date
- Ad ([ad id](#))
- Viewers' [preferences](#)
- Viewers' region
- Device type (smartphone, tablet, desktop, smart TV) on which the original ad was viewed

5. Non-functional Requirements

5.1. External Interface Requirements

5.1.1. User Interfaces

User interfaces of the System present in the following forms:

- Viewers' devices without the mobile app;
- Viewers' mobile apps;
- Web interface for customers;
- Web interface for advertisers;
- Web interface for administrators.

This section describes general requirements for all user interfaces.

UI-1 – Reactive UI

Each System UI shall react to every user action that is supported by the System in no more than 0.2 seconds. This reaction must be visible.

UI-2 – Touch support

For screens of **mobile devices**, the following constraints must be applied to the System user interfaces:

1. All UI elements must be large enough to be easily touchable. Touch target (touchable area) must be 9 mm in size, and never less than 7 mm, according to device pixel density.
2. UI elements must be spaced so that adjacent targets have at least 2 mm of space between them.

For **web user interfaces** that can be used on touch devices (like the web interface for customers), the same requirements must be applied.

UI-3 – Responsive UI

System UI on different platforms must be responsive to screen/window size and orientation (landscape/portrait).

UI-4 – Internationalization and localization

The System must support [internationalization](#), which includes language localization and the use of national data formats.

By default, the System UI must be in American English (en-US) so that all numbers, currency, date, time, phone numbers, and postal addresses (if needed) meet the national formats used in the USA.

UI-5 – Mandatory fields indication

System must mark all mandatory fields in case of multiple field filling.

UI-6 – Displaying requirements to field values

System must display requirements for specifying values, according to the [Data Dictionary](#) if some of them were submitted incorrectly.

5.1.2. Hardware Interfaces

The described functionality must work on the same devices as the entire System works.

HI-1 – Determining unique device id

System shall use platform-specific hardware interfaces to determine unique [device id](#). This determination may be made on the basis of device [MAC address](#), [IMEI](#) (for mobile devices), and other device properties that allow the System uniquely to identify a device.

5.1.3. Software Interfaces

SI-1 – Support of operating systems and browsers

From the user's perspective, the System's web interfaces shall work at the following platforms and operating systems in the listed browsers:

- Microsoft Windows:
 - Google Chrome (recent version);
 - Mozilla Firefox (recent version);
 - Microsoft Internet Explorer 10 and 11;
 - Microsoft Edge (recent version).
- Linux:
 - Mozilla Firefox (recent version);
 - Google Chrome (recent version);
- Apple macOS:
 - Safari (recent version);
 - Google Chrome (recent version);
 - Mozilla Firefox (recent version);
- Apple iOS:
 - Safari (recent version);

- Google Chrome (recent version);
- Microsoft Edge (recent version);
- Google Android:
 - Google Chrome (recent version);
 - Mozilla Firefox (recent version);
 - Microsoft Edge (recent version).

Supported Viewers' devices (including **the mobile app** on Viewers' mobile devices) must be the same as what the Service provider supports for transmitting video content.

SI-2 – Integrations for sharing on social networks

System shall integrate on mobile devices with APIs of social networks for sharing ads ([FR-4.13](#)). This also includes interaction with installed native apps of these services so that, if a native app of a chosen service is installed, its dialog is used for sharing.

SI-3 – Integration with default dialer

System shall integrate with a default dialer on Viewers' mobile devices to make phone calls ([FR-4.11](#)).

SI-4 – RESTful API for internal and external use

System shall provide an API for the following purposes:

- user management and authentication;
- viewer and device management;
- management of ad metadata;
- recording of data on ads watched by viewers;
- saving information about viewers' actions;
- retrieving unrecalled ads (including viewers' favorites and history);
- reports generation.

The API must be RESTful and accept/return data in JSON.

System shall provide secure access to API calls by permissions according to [user roles](#). Currently, all endpoints and operations must be available to users with the user role “Admin”. Any such user can be able to generate, and after that, re-generate a personal random [API key](#) in the user account settings.

Responses to API calls must be standard HTTP status codes, according to a situation. At least the following codes must be used:

- 200 — call performed successfully;
- 201 — call performed successfully and a requested object has been created;

- 400 — required fields are missing or invalid (with error detailization);
- 401 — authentication failure;
- 403 — access denied (now not used because of equal permissions);
- 404 — requested data not found;
- 405 — method not allowed (returned when an API method unsupported for an endpoint is requested);
- 500 — the System cannot process the request.

All API calls must be logged for a predefined interval (30 days by default) without transmitted data, but with at least the following values:

- [user account](#);
- [timestamp](#) when an API call was received;
- API endpoint and parameters (if any);
- returned HTTP status code.

SI-5 – Application integrated to third-party services

System shall provide architecture capability to create and operate applications for third-party platforms, such as digital media players or social services. Such applications will be System-related, will interact with the System API ([SI-4](#)), and will provide to viewers the same (or similar) functionality as the System’s mobile apps (including displaying video content and unrecalled ads).

5.1.4. Communications Interfaces

System shall be available to users through the Internet by a web address in a web browser or by direct connection of the System application. HTTPS must be used for this communication and for the RESTful API ([SI-4](#)).

5.2. Other Non-functional Requirements

5.2.1. Performance Requirements

System¹ shall be able to work with stability under the following **workload**:

- Up to 100.000.000 registered users;
- Up to 500.000.000 registered viewers;
- Up to 100.000.000 viewers simultaneously watching video content and video ads in peak hours;
- Up to 500.000.000 viewed video ads per hour in peak hours;
- Up to 150.000.000 requests for unrecalled ads per hour during peak hours.

System shall meet **response times** for the following scenarios:

- Requesting unrecalled ads (from initiation by a viewer to displaying a list of unrecalled ads) — no more than 3 seconds in any scenario (pressing the activation button or opening the mobile app);
- Generating a report (from initiation by a user to displaying report data) — 30 seconds.

System shall continue meeting the stated response times as workload scales (increases) over time.

5.2.2. Security Requirements

SE-1 - Encrypted data storage

Sensitive data (such as user [passwords](#), data related to means of payment (if any), etc.) must be stored in an encrypted form.

SE-2 - Encrypted data transfer

Data transferred between different parts of the System (components) must be encrypted. All connections used for accessing the System must be encrypted (see [Communications Interfaces](#)).

SE-3 - Authorized access

System shall permit users to access operations and data objects which are available only to registered users after user authentication procedure and according to their permissions. Permissions are defined by [user roles](#) and eventual relationships of data objects to [user accounts](#), as defined in the table below.

¹ Implemented according to one or several standards listed in the section 2.3 (Description of Solution).

Object type \ User role	Admin	Customer	Advertiser
user account	CRUD ¹	CRUD ²	CRUD ²
device	CRUD	CRUD ²	—
viewer	CRUD	CRUD ²	—
preferences	CRUD	CRUD ²	—
ad metadata	CRUD	R	CRUD
ad impression metadata	CRUD	R ²	R
viewer's action	CRUD	C	R
unrecalled ads	RD ³	RD	R

Notes:

¹ CRUD stands for permissions to Create, Read, Update, and/or Delete.

² Users with roles except “Admin” can access and modify only their own user accounts and other data objects related to such accounts.

³ Unrecalled ads are saved in the history together with their requests thus do not require separate “create” permissions.

SE-4 - Consent for data collection

System shall disable most of the user’s functions in case a viewer did not give consent for collecting data. The following functions must be made disabled:

- Collecting metadata about viewing ads by the viewer ([FR-3.1](#));
- Working with unrecalled ads ([FR-4.1](#) and all other functional requirements listed in section [4.4](#)).

When the Viewer gives consent from the mobile app settings ([FR-1.9](#)), the System must enable the listed functions.

5.2.3. Software Quality Attributes

SQA-1 - Availability

System shall be available to users no less than 99.9% of the time (including planned downtime) which corresponds to unavailability around 9 hours per year or about

45 minutes per month. System is considered available if and only if the System meets [performance requirements](#).

SQA-2 - Correctness

SQA-2.1 - Correctness of data storage and modification

System shall store data and modify already stored data if and only if an appropriate operation has been performed over them.

SQA-2.2 - Correctness of unrecalled ads extraction

System shall retrieve unrecalled ads correctly according to defined algorithms ([FR-4.2](#)). Their functionality must be fully verified with automated tests.

SQA-3 - Interoperability

For interchange with external systems and data input, the System must only use standard file formats (such as CSV, JSON, XML, etc.) and communications protocols (such as HTTPS, FTP, etc.).

SQA-4 - Robustness

SQA-4.1 - Robustness, in case of a broken connection

If a connection between a User and the System or between two components of the System is broken prior to completion of an operation in which these parts were involved, the System shall leave all related data as it was before the operation started.

SQA-4.2 - Robustness, in case of failure

System shall protect stored data from corruption and inconsistency in case of failure on the System side by performing backups at least daily. These backups should include both entire disk backups (snapshots) and particular database backups. The period of storing these backups must be no less than 30 days.

SQA-5 - Scalability

System architecture must be able to handle growing amounts of users, viewers, and ads (see [Performance requirements](#)).

6. Data Models

The diagram below shows the **conceptual data model** with interrelations among the main data objects:

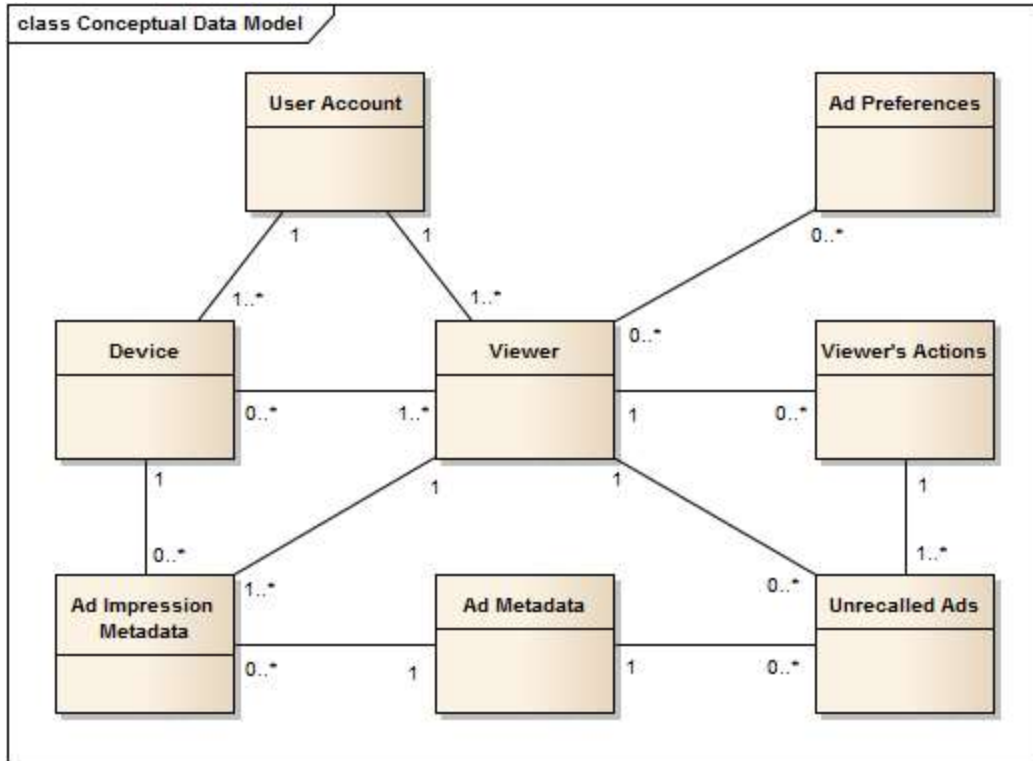


Fig. 6. Conceptual data model

Data dictionary below details the conceptual data model. It describes the contents of each data object and their requirements. The data dictionary is written using the DeMarco notation that defines the following main rules:

- Defined data item is in the left column, and its definition is in the right column, which contains constituents of the defined item (one per line).
- Elementary data items are explained using asterisk signs: * elementary item *.
- Comments in complex data items are given in the same way.
- Optional items are given in round brackets: (optional item).
- Multiplicity is shown in curly brackets: "m:n{item}", where m and n are the minimum and maximum number of items (if n is not defined, then the maximum number of items is limited only with technical capabilities of the System).
- Enumerations are given in square brackets with a vertical bar between: ["first" | "second" | "third"].

All data items are listed alphabetically.

action type	=	[“Watching ad” “Unrecalled ads request” “Opening mobile app after watching the ad” “Opening link” “Making call” “Social sharing”] * Type of a Viewer’s action related to an ad that is recorded by the System. *
ad description	=	* A string that describes an ad (goods or services) and is displayed in the details of an ad or together with an ad title while displaying video content. Minimum length is 150 characters. *
ad duration	=	* Decimal number that represents ad duration in seconds. *
ad id	=	* An alphanumeric string that uniquely identifies an ad. As a common option, the advertising industry standard unique identifier [5] that has its maximum length of 12 characters can be used for this purpose. *
ad image	=	* A string with an absolute URL of a bitmap image in one of the supported formats (PNG, JPG, GIF) that visually represents a related ad. Minimum and maximum image resolutions are defined by the Service provider. The maximum size is 150 KB. *
ad impression metadata	= + + +	ad id start time viewer device
ad metadata	= + + + + + +	ad id ad duration ad title ad description ad image (landing page) (phone number)
ad search timeframe	=	* An integer parameter used for retrieving unrecalled ads based on viewed ads. It represents a time interval in seconds during which viewed ads are retrieved relatively to the request time. The default value is 30 seconds. *
ad title	=	* A string that is displayed to the Viewer as a title of an unrecalled ad and gives the main meaning of the ad. The maximum supported length is 100 characters. *

api key	=	* An alphanumeric string that uniquely defines a System user and serves as a password for accessing the System API.
channel number	=	* A non-negative integer number of a service provider's channel. *
device	= +	device id device name
device id	=	* A string that uniquely identifies a Viewer's device among all devices registered at the Service provider. This identifier is device-dependent, i.e., it matches only a particular device and can be generated anew based on the same parameters. *
device name	=	* A string that represents a device in the user interface. Maximum 20 characters. *
email address	=	* A non-empty text string that uniquely identifies an email address according to the standard RFC 5322 ; maximum length is 254 characters (see also Email address on Wikipedia). *
landing page	=	* A string that represents a URL of a landing web page assigned to an ad. Starts with "http://" or "https://" and meets a URL format . *
name	=	* A non-empty alphabetic string that stands for a user name or a profile/viewer name. The length is up to 30 characters. *
number of unrecalled ads	=	* A positive integer number that defines how many unrecalled ads will be returned in response to a viewer's request. *
password	=	* A text string used for user identification along with a username (email address). The minimum allowed length is eight characters; the maximum length is 256 characters. The password must contain at least one uppercase, one lowercase, one digit, and one special character (ASCII Punctuation & Symbols – #33–47 and #58–64). Passwords are stored in the database in encrypted form. *
phone number	=	* A string that represents a full phone number with a plus sign ("+"), country code, and local code. The maximum length is 20 characters. Only "+" and digits are allowed. *
preferences	=	* A set of data about a viewer and viewer's ad preferences such as viewer's age (or age group), viewer's sex, and viewer's interests from a predefined list. *

start time	=	timestamp * Timestamp of a moment when an ad is shown to a viewer started. *
timestamp	=	* Date and time (in seconds) of a related event (e.g., creating or modifying a piece of data). Used without a time zone supposing that it is a server time zone. *
viewer	= +	name preferences
viewer's action	= + +	timestamp action type ad id
unrecalled ads	= + +	0:n{ ad metadata } timestamp viewer
user account	= + + + +	email address * also user's identifier/login * password user role name (api key)
user role	=	["Customer" "Advertiser" "Admin"]

The **logical data model**, based on the data dictionary, is represented in a separate diagram to show definite data structures and their relationships. It will form a base of a physical data model and a database of the System.

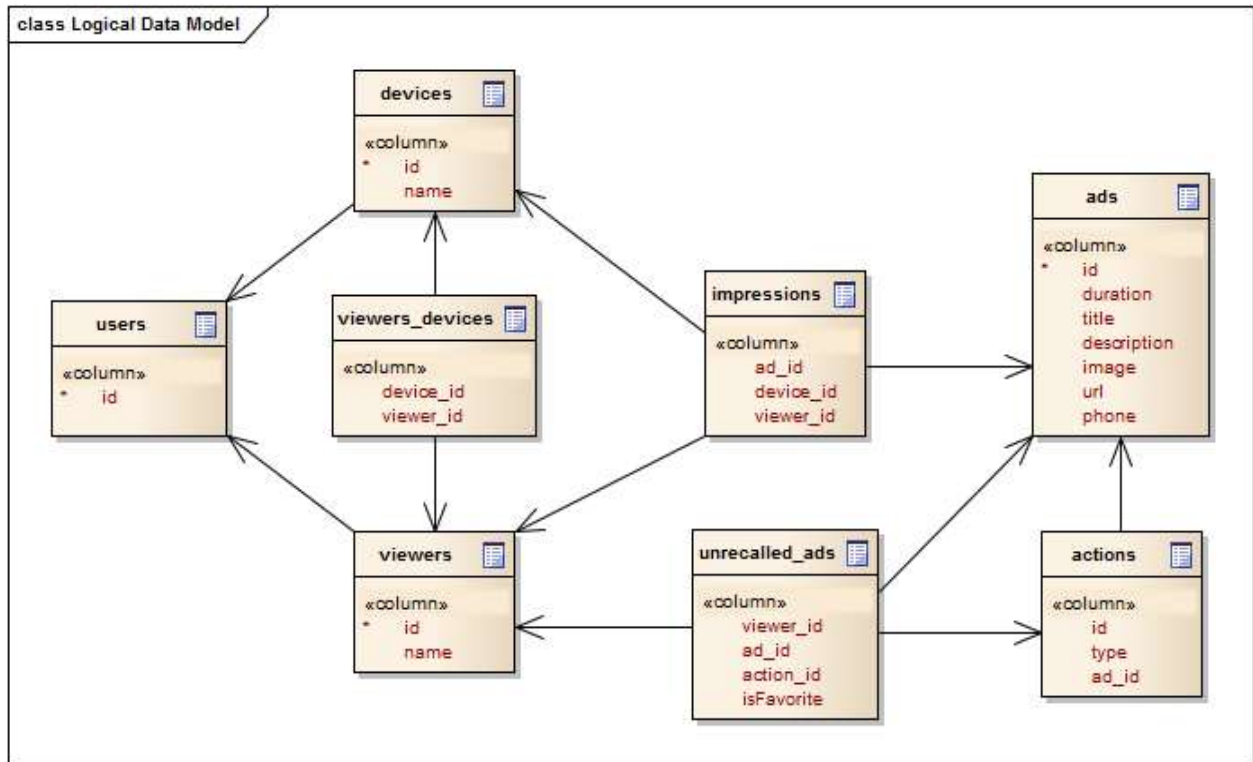


Fig. 7. Logical data model

7. Business Rules

ID	Rule Definition	Type of Rule ²	Static or Dynamic ³
BR-1	The Streaming service and player providers should have the metadata input required for the ad listings provided by the advertisers.	Fact	Static
BR-2	The Streaming service and player providers should receive money from advertisers via Pay-per-Tap® (according to the data collected with the help of the IntapTV technology).	Fact	Static
BR-3	The Streaming service and player providers should determine a reasonable cost per tap, such as that which is displayed in the following example. The average click-through rate for paid search ads is about 2%. IntapTV technology does not have any tap-through rates on paid search ads, meaning all taps are paid taps. Based on that, TV advertisers can spend X times less money on each tap. For example, if an insurance company were to run a TV ad which created an increased search volume, but then spent \$50 per click on paid search ads while on IntapTV, they would spend X times less per tap.	Inference	Dynamic
BR-4	The streaming service and player providers retain ownership over their interaction data. They can use these primary data to target audiences with more relevant or personalized advertisements.	Fact	Static

² Fact, Constraint, Action enabler, Computation or Inference

³ Indicates if a rule changes in the course of time