



WP7 – Multi-Device Media Representation and Interaction

D7.9.4: Multi-Device Media Representation and Interaction Public Report (Final Version)

Deliverable Lead: TPVI

Contributing Partners: TPVI, ASC, TALK, TIE, NTUA

Delivery Date: 2016-10

Dissemination Level: Public

Final

This deliverable, the fourth and last of the D7.9.X series, is the description of the fourth prototype implementations of the Task T7.4 SAM multi-device dashboard. The deliverable covers the descriptions of the software deliverables in WP7. This document is a living document that is enhanced with each delivery of the different iterations of the WP7 prototypes.



Document Status	
Deliverable Lead	TP Vision
Internal Reviewer 1	UA: David Tomas
Internal Reviewer 2	BDS: Barry Smith
Type	Deliverable
Work Package	WP7 – Multi-Device Media Representation and Interaction
ID	D7.9.4: Multi-Device Media Representation and Interaction Public Report
Due Date	10.2016
Delivery Date	10.2016
Status	Final

Document History	
Versions	V1.0: including first and second reviews feedback

Disclaimer

The views represented in this document only reflect the views of the authors and not the views of the European Union. The European Union is not liable for any use that may be made of the information contained in this document.

Furthermore, the information is provided “as is” and no guarantee or warranty is given that the information is fit for any particular purpose. The user of the information uses it at its sole risk and liability.

Project Partners



TIE Nederland B.V., The Netherlands



Ascora GmbH, Germany



Talkamatic AB, Sweden



TP Vision Belgium NV, Belgium



Institute of Communication and Computer Systems, National Technical University of Athens, Greece



The University of Reading, UK



Universidad de Alicante, Spain



Deutsche Welle, Germany



Bibliographic Data Services Limited, UK

Executive Summary

This public document describes the status and results of the different WP7 prototypes without exposing confidential information such as prototypes access information and source code. This confidential information can be found in the related Programme Participants Prototypes (D7.1.3, D7.2.3, D7.3.3 and D7.4.3).

This document is iteratively updated coinciding with the delivery of the different related software prototypes of WP7:

- SAM Multi-Device Content and Media Representation (T7.1)
- SAM 2nd Screen Media Interaction (T7.2)
- SAM 1st Screen Media Interaction (T7.3)
- SAM Multi-Device Dashboard (T7.4)

Each part is being developed iteratively with 3 prototypes, except for T7.4, which has 4 prototypes, and produces a software deliverable and an update of this public documentation (in the series D7.9.1 - .2 - .3 - .4).

As such, this fourth version of this deliverable (D7.9.4) describes the developments and results of the fourth iteration of the prototypes in WP7. Although in the DOW, T7.4 (SAM multi-device dashboard) was planned to start in M20, it was decided, and also indicated in D7.9.1, to start it in M14 because it has common parts to be used by both the 1st and 2nd Screen.

At this point in time the deliverable contains the following software prototype information:

Task	Component	Software Prototype Deliverable	Due	Section
T7.2	SAM 2 nd Screen Media Interaction	D7.2.3	M37	3
T7.4	SAM Multi-Device Dashboard	D7.4.3	M37	4

Figure 1: Overview of Tasks and Software Deliverables

This document complements D7.9.3, which contains the software prototype information for the components T7.1 – SAM Multi-Device Content and Media Representation and T7.3 – SAM 1st Screen Media Interaction.

For each of the prototypes, the following information is presented:

- **Scope and Relationship:** Describes the scope of the prototypes implementation, its purpose and the main relationships with other modules being implemented in SAM
- **Requirements and Preparations:** Introduces the information needed to deal with the prototype, in terms of technical and non-technical requirements, software to be installed, etc.
- **Installation:** Describes the steps needed to install the software, and how to build it from source code
- **Execution and Usage:** Presents the different screens and actions implemented in the prototype itself, how to access it and how to test the different implemented options
- **Limitations:** Depicts the current prototypes limitations
- **Research Background:** Presents relevant references to research publications
- **Target Performance:** Lists measurable targets for functions and usability
- **Summary:** Describes the conclusions of the implementation of the fourth prototype

Table of Contents

1	Introduction	6
1.1	SAM Project Overview	6
1.2	Deliverable Purpose, Scope and Context	6
1.3	Document Status and Target Audience	7
1.4	Abbreviations and Glossary	7
1.5	Document Structure	7
1.6	External Annexes and Supporting Documents	8
2	WP7 Introduction	9
3	SAM 2 nd Screen Media Interaction	11
3.1	Scope and Relationship	11
3.2	Requirements and Preparations	13
3.2.1	For Users	13
3.2.2	For Developers	14
3.3	Installation (Deployment)	14
3.3.1	For Users	14
3.3.2	For Developers	15
3.4	Execution and Usage	16
3.4.1	For Users	16
3.4.2	For Developers	20
3.5	Limitations	26
3.6	Research Background	26
3.7	Target Performance	27
3.7.1	Component KPIs	27
3.7.2	User Experience Measurements Tasks	27
3.8	Summary	28
4	SAM Multi-Device Dashboard	29
4.1	Scope and Relationship	29
4.2	Requirements and Preparations	33
4.2.1	For Users	33
4.2.2	For Developers	34
4.3	Installation (Deployment)	34
4.3.1	For Users	34
4.3.2	For Developers	34
4.4	Execution and Usage	34
4.4.1	For Users	34
4.4.2	For Developers	34
4.5	Limitations	38
4.5.1	Planned tasks – Future improvements	38
4.6	Research Background	38
4.7	Target Performance	39
4.7.1	Component KPIs	39
4.7.2	User Experience Measurements Tasks	40
4.8	Summary	40
5	Document Summary	42
	References	43
	Annex A: User Research	44

1 Introduction

SAM – Dynamic Social and Media Content Syndication for 2nd Screen – is a project funded by the Seventh Framework Programme of the European Commission under Grant Agreement No. 611312. It provides a content delivery platform for syndicated data to be consumed in a contextualised social way through 2nd Screen devices.

1.1 SAM Project Overview

The current generation of Internet-connected devices has changed the way users interact with media. Previously, users were restricted to being passive and unidirectional consumers; now, they are proactive and interactive media users. They can comment on and rate a television show or film and search for related information regarding cast and crew, facts and trivia or even filming locations. They do this with both friends and wider social communities through the so-called “2nd Screen”.

Another related phenomenon is “Content Syndication”, which is a field of marketing where digital content is created once and delivered to consumers through various different marketing channels (devices, markets and stakeholders) simultaneously, enabling efficient content control, delivery and feedback.

However, the 2nd Screen phenomenon has grown in a disorderly manner. Tools supplied by the media provider companies (e.g. as mobile or tablet apps) limit the potential outreach and, as a result, users are not enjoying relevant contextual syndicated information. European enterprises wishing to provide services have limited methods of receiving feedback, restricting the business intelligence that can be extracted and applied in order to profit from and enrich this growing market.

SAM is reshaping the current disorganised 2nd Screen ecosystem by developing an advanced social media delivery platform based on 2nd Screen and Content Syndication within a social media context. This is achieved by providing open and standardised means of characterising, discovering and syndicating media assets interactively. Users will be able to consume and prosume digital assets from different syndicated sources and synchronised devices (e.g. connected televisions), creating more fulfilling experiences around the original media assets.

The SAM vision that is now becoming reality sees the former, out-dated system of users searching for the information they desire replaced with a new approach where information reaches users on their 2nd Screen using content syndication. This is enriched through the creation of dynamic social communities related to the user and digital asset context (e.g. profiles, preferences and devices connected). These are continuously evolving social spaces where people share interests, socialise and build virtual communities. SAM will enable syndication of comments, ratings, facts, recommendations and new information that will enrich and energise the virtual community as well as enhance personalised knowledge and satisfaction.

1.2 Deliverable Purpose, Scope and Context

The purpose of this deliverable is to accompany the software prototypes of WP7 tasks T7.1 SAM multi-device content and media representation, T7.2 SAM 2nd Screen media interaction, T7.3 SAM 1st Screen media interaction and T7.4 SAM multi-device dashboard. Each task will contribute different components to the SAM architecture that are developed

iteratively in 3 phases as per milestones 3/4/5 at M19/25/37 for tasks T7.1 and T7.3 and in 4 phases as per milestones 3/4/5/6 at M19/25/37/37 for T7.2 and T7.4 and will produce a software deliverable and an update of this public documentation (D7.9.x) with the timings indicated in the following table:

Deliverable	Date
D7.9.1	M19
D7.9.2	M25
D7.9.3	M37
D7.9.4	M37

Figure 2: Deliverable Schedule

As the main focus of the tasks is the development of the software itself, this accompanying document focuses on providing a short summary of the main functionalities and on serving as user guide for the current status of the development.

This document builds upon document D7.9.3, which focuses on T7.1 and T7.3. This D7.9.4 is covering T7.2 and T7.4.

1.3 Document Status and Target Audience

This document is the fourth iteration of the D7.9.x series and is listed in the DOW as public. It is primarily aimed at the project partners as a user guide but it also presents status information of the prototypes of the software components of WP7 to the interested public.

1.4 Abbreviations and Glossary

A definition of common terms and roles related to the realisation of SAM, as well as a list of abbreviations, are available at <http://wiki.socialisingaroundmedia.com/index.php/Glossary>

1.5 Document Structure

This deliverable is broken down into the following sections:

- **Section 1 (Introduction):** Provides an overview of the entire document and the related pilot implementation, describing the main objectives, constraints and status
- **Section 2 (WP7 Introduction):** Provides an overview of WP7 goals and the WP7 prototypes
- **Section 3 (SAM 2nd Screen Media Interaction):** Describes the latest software deliverable developed in T7.2
- **Section 4 (SAM Multi-Device Dashboard):** Describes the latest software deliverable developed in T7.4
- **Section 5 (Document Summary):** Briefly summarises the work presented at the deliverable, as well as the overall WP7 status

In Sections 3 and 4, for each component in the SAM Architecture, the following subsections are provided:

- **Scope and Relationship:** Describes the scope of the prototype implementation, its purpose and the main relationships with other modules implemented in SAM.

- **Requirements and Preparations:** Introduces the information needed to deal with the prototype in terms of technical and non-technical requirements, software to be installed, etc.
- **Installation:** Describes the steps needed to install the software, and how to build it from source code.
- **Execution and Usage:** Presents the different screens and actions implemented at the prototype itself, how to access it, and how to test the different implemented options.
- **Limitations and Further developments:** Depicts the current prototype limitations and the expected improvements.
- **Research Background:** Presents relevant references to research papers and publications
- **Target Performance:** Lists measurable targets for functions and usability
- **Summary:** Describes the conclusions of the implementation of the fourth prototype.

1.6 External Annexes and Supporting Documents

- D7.1.3: Multi-Device Content & Media Representation
- D7.2.3: 2nd Screen Media Interaction
- D7.3.3: 1st Screen Media Interaction
- D7.4.3: Multi-Device Dashboard
- D7.9.3: Multi-Device Media Representation and Interaction Public Report (Third Version)

2 WP7 Introduction

WP7 is concerned with the multi-device representation and interaction with asset content taking into account the wide spectrum of devices with different specifications in the market.

Specific objectives of this WP include:

- To implement a framework for multi-device based media representation (T7.1)
- To provide implementations of advanced user interaction using voice recognition, Inter-Widget-Communication (IWC) and 1st Screen component detection (T7.2)
- To produce a multi-device dashboard supporting media representation and advanced user interaction (T7.3, T7.4)

The results of WP7 are a set of related applications where the consumer/user can experience all the other SAM RTD WPs. Each component is developed iteratively s per milestones 3/4/5 (for T7.1 and 7.3) at M19/25/37 and 3/4/5/6 (for 7.2. and 7.4) at M19/25/37/37 and will produce a software deliverable and an update of the public documentation (D7.9.x – this document series).

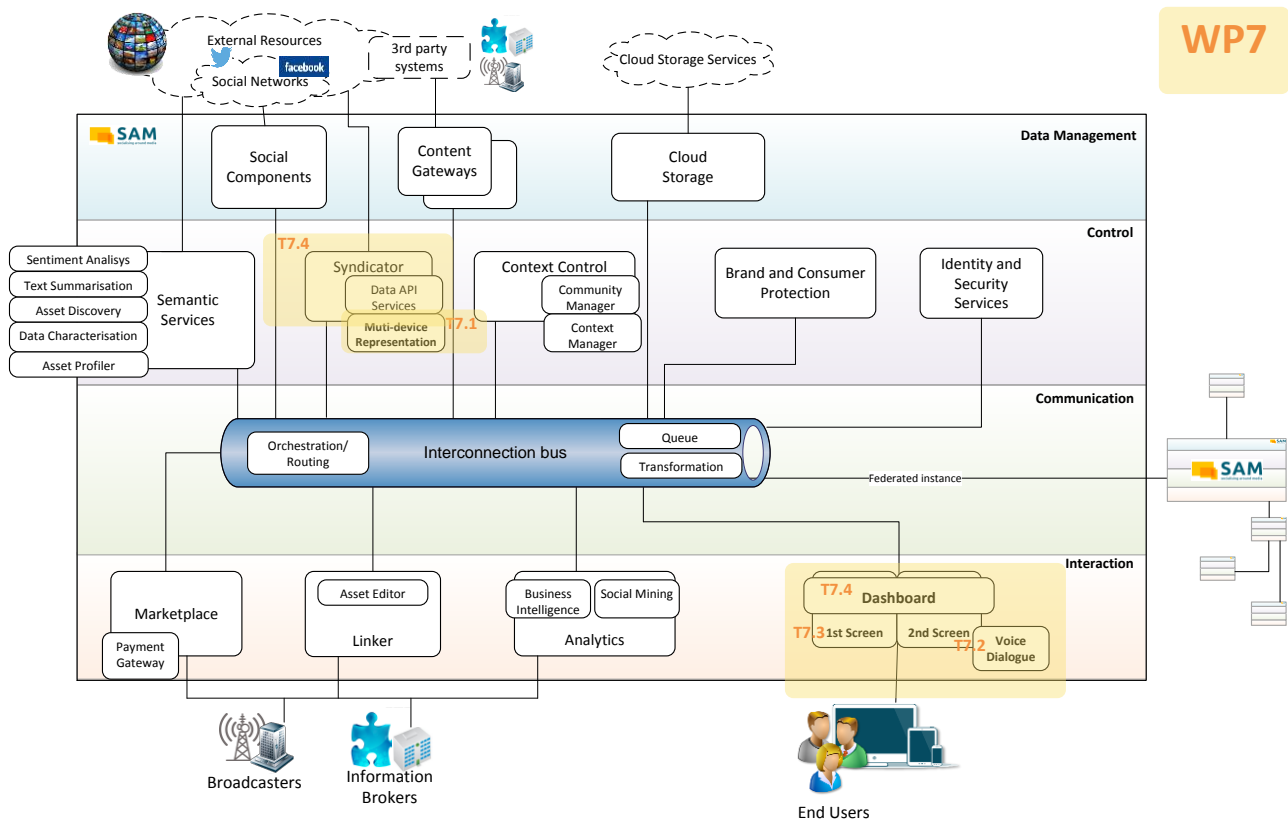


Figure 3: WP7 Components Contribution to SAM Architecture

The work in WP7 (as well as in the other development WPs) is managed by using the agile SCRUM methodology. For that purpose, a dedicated WP7 SCRUM board has been created in the SAM Jira task management system, with a representative of the WP Lead (TPVI) as the Scrum Master.

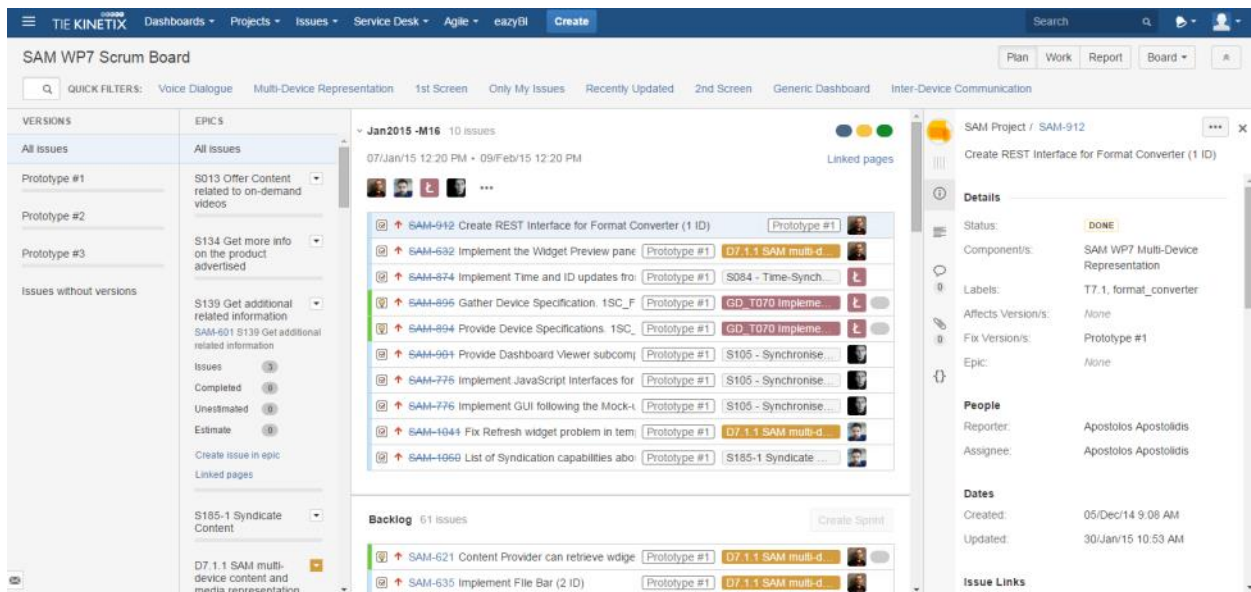


Figure 4: WP7 Scrum Board

Sprints were planned and executed monthly, and every story or task is linked to one or more specific requirements as expressed in D2.3 (User Stories and Requirements).

A Planning meeting was scheduled at the beginning of each sprint in order to plan the next monthly sprint and discuss the priorities or reschedule the unfinished work from the previous one. A Retrospective or Review meeting has also been scheduled at the end of each sprint in order to discuss the work done during the sprint and find ways of improving (if necessary) the way of working.

During the development for the fourth Prototype of the different components developed in this WP, additional technical meetings were held with all technology partners, special attention was paid to effective collaboration by having shared information and development sessions under one roof.

3 SAM 2nd Screen Media Interaction

This section describes the software deliverable D7.2.3, which is the third prototype release of the SAM 2nd Screen component.

3.1 Scope and Relationship

The 2nd Screen component enables the presumption of content corresponding to the current video, which the end user is viewing. Figure 5 shows the different subcomponents of the 2nd Screen component, the logical connections that have been established between them and the relations with other components and actors in the SAM Platform.

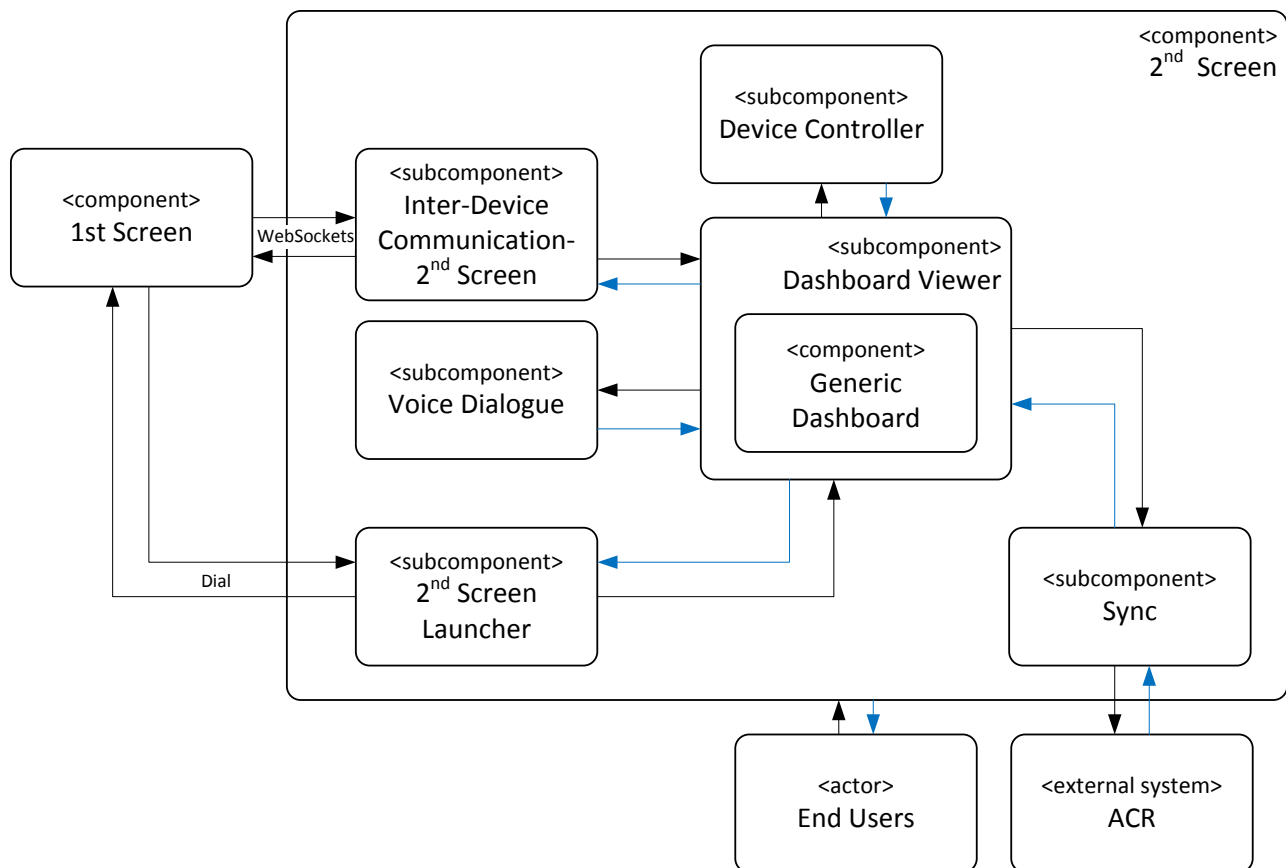


Figure 5: 2nd Screen Subcomponents and their Relationships

For further descriptions of the functional and technical foundations of these subcomponents, please revisit documents D3.2.1 Section 4.13.4 (Architecture), D3.2.2 Section 4.14.3 (Functional Specification) or D3.3.1 Section 3.14.3 (Technical Specification).

The first prototype of T7.2 provided the basic integration of the Generic Dashboard component using the Dashboard Viewer subcomponent. The Dashboard Viewer subcomponent provided JavaScript interfaces which provide device specific information by using the Device Controller subcomponent. A basic remote control function had also been implemented which allows the 2nd Screen Android application user to start, stop and pause the video element displayed on the 1st Screen.

Additionally, a first integration of the Voice Dialogue subcomponent and the Talkamatic Dialogue Manager (TDM) backend had been carried out to support voice interaction using

the 2nd Screen Android application. In the first prototype, the Voice Dialogue only received user input and gave mock feedback, not interacting with the widgets on the 2nd Screen.

A summary of the tasks carried out for the different subcomponents of the first version of the prototype is shown in the following table:

Subcomponent	Task
Voice Dialogue	<ul style="list-style-type: none"> Basic integration into the Android application and setup of the TDM back end
Dashboard Viewer	<ul style="list-style-type: none"> Display and remote control support of the Generic Dashboard component, Provide JavaScript interfaces for communication
Device Controller	<ul style="list-style-type: none"> Provision of device specifications and GPS data to the Generic Dashboard component using the Dashboard Viewer subcomponent

Figure 6: Tasks Completed for the First Prototype of T7.2

The second prototype of T7.2 added the integration of the Identity and Security services component, which enabled end users to log into the 2nd Screen application. This process is based on OpenID. Additionally, a local connection between 1st Screen and 2nd Screen devices was implemented. The search for local 1st Screen devices was realised by Android's "Network Service Discovery" (NSD) feature. Also, the user interface for a remote control was updated to match the Android design recommendations and to provide a common look and feel.

On the side of the Voice Dialogue (VD) support, the Twitter use case was implemented. This enabled the end user to tweet by using voice commands. This feature was based on the dictation functionality, which also was integrated in the second prototype.

VD was converted to support a new integration model developed by Talkamatic. This model allows the GUI, in the 2nd Screen case, to decide which actions and data it will share with VD. Previously the 2nd Screen app executed actions upon request from VD, while now the 2nd screen decides whether and how certain events dispatched from VD will be processed. This allows for a more flexible implementation of the VD component by the parent app. Currently, VD supports launching popups, Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) events and simple actions e.g. posting on Twitter.

A summary of the tasks completed for the different subcomponents of the second version of the prototype are shown in the following table:

Subcomponent	Task
2 nd Screen Launcher	<ul style="list-style-type: none"> Implement NSD support to discover local 1st Screen devices
Dashboard Viewer	<ul style="list-style-type: none"> Provide integration of Identity and Security Services component to provide login functionality based on OpenID Replace existing remote control implementation with a new approach based on Android design recommendations
Voice Dialogue	<ul style="list-style-type: none"> Implementation of the new integration model Implementation of dictation Support for the Twitter use case

Figure 7: Tasks Completed for the Second Prototype of T7.2

The third prototype of T7.2 focused mainly on bug fixing and performance optimisation. Besides that the registration form has been implemented, which allows new users to create a SAM account.

The third prototype of Voice Dialogue contains multiple and important changes. Functionalities such as posting on social media, reading users' posts, subscribing to a user's activities and learning information about people and films were implemented. Additionally, Voice Dialogue acquired support for asynchronous actions so that it can query information from the Semantic Services. Finally, a verbal help functionality was added so that the user can get hints about what he or she can say.

A summary of the tasks completed for the different subcomponents of the third version of the prototype are shown in the following table:

Subcomponent	Task
Overall	<ul style="list-style-type: none"> • Implementation of registration form • Bug fixing of already implemented features based on trial feedback
Voice Dialogue	<ul style="list-style-type: none"> • Implement posting on Facebook and SAM communities • Implement reading user's posts on SAM communities • Implement subscription to a specific user's activity on a SAM community • Implement getting information about people and films • Implement support for asynchronous actions • Improve stability of platform

Figure 8: Tasks Completed for the Third Prototype of T7.2

3.2 Requirements and Preparations

This section provides information on technical and non-technical requirements for users as well as for developers.

3.2.1 For Users

The user needs to provide an Android device (smartphone or tablet) running Android 4.4 or newer¹. Since the download of the 2nd Screen Android application is not available on Google Play² yet, the following steps have to be undertaken to retrieve the installation file and install the application on an Android device:

1. Download the installation file of the application from the Jenkins Continuous Integration server³ provided by the SAM consortium
2. Start your Android device
3. Open the Settings
4. Go to the Security section
5. Activate the option "Unknown Sources"
6. Move the installation file to the device
7. Start the installation process

Further information about the retrieval of the 2nd Screen application can be found in the D7.2.2 deliverable.

¹ <http://www.android.com/versions/lollipop-5-0/>

² <https://play.google.com/store>

³ <https://jenkins-ci.org/>

3.2.2 For Developers

For developers it is strongly recommended to use the Android Studio⁴ IDE for the implementation. Besides of the usage of a physical device, developers can also use an Android Virtual Device (AVD), which is provided by Android SDK.

Regarding the Voice Dialogue component, it is essentially a .aar file, that is, an Android library file. It can be imported to any Android project using the standard procedures for importing an .aar file. The library communicates with the Voice Backend over the Websockets protocol.

3.3 Installation (Deployment)

This section provides guidelines on how to install the app (see Section 3) and to retrieve the source code of the 2nd Screen application (see Section 3).

3.3.1 For Users

First the user has to download the 2nd Screen application, which can be found on the Jenkins server (URL can be found in deliverable D7.2.3). After downloading the 2nd Screen application and moving it to the device's internal or external storage, the user has to execute the *.apk file. As the first step, the user will be requested for the acceptance of the access rights (see Figure 9). After the installation, the 2nd Screen application can be started using the icon in the app overview. The Voice Dialogue component starts automatically each time the main application is started.

⁴ <http://developer.android.com/sdk/index.html>

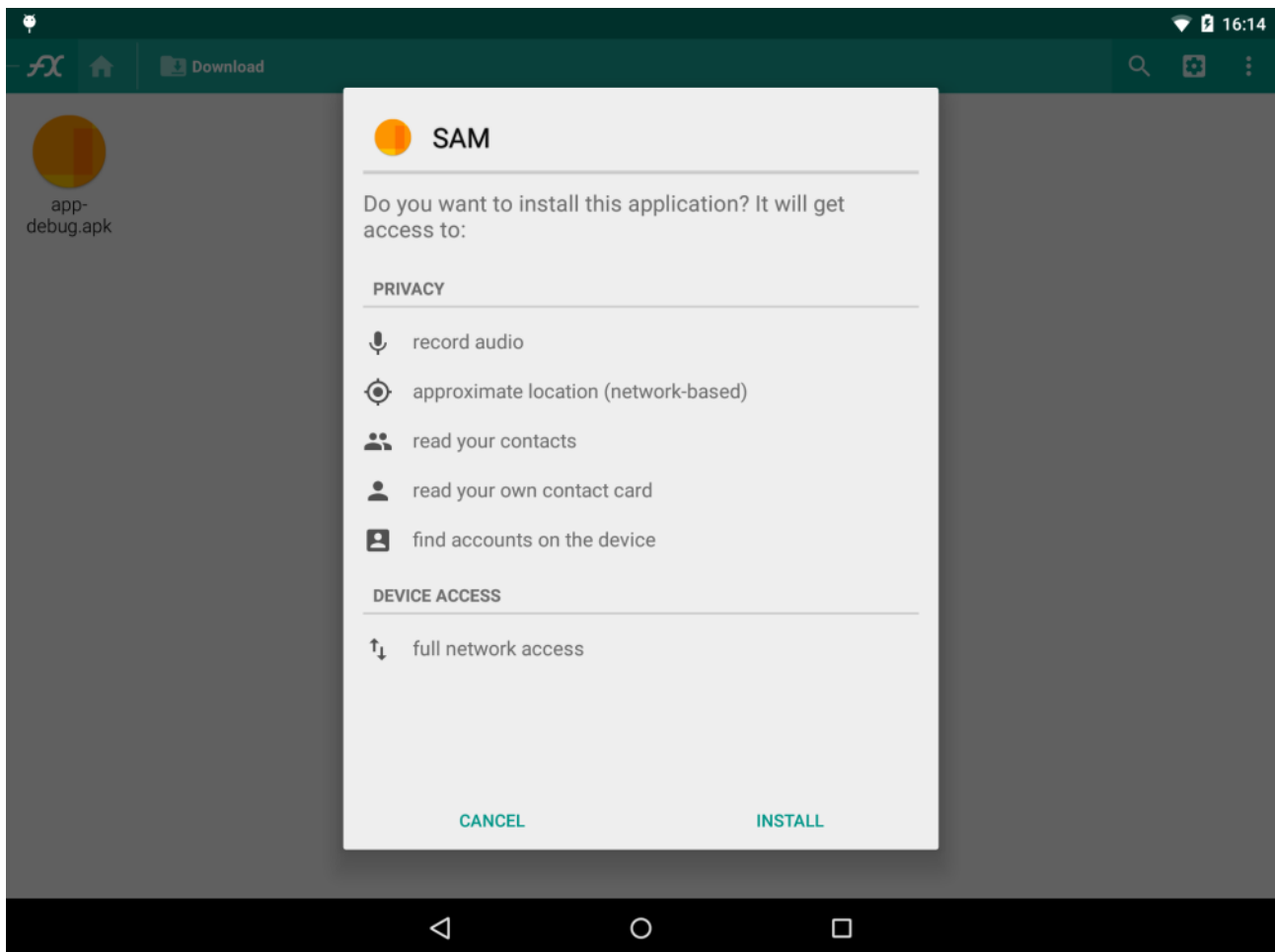


Figure 9: Installation of SAM 2nd Screen Application

3.3.2 For Developers

First, the developer has to download the 2nd Screen repository, which is available at the GitLab (see deliverable D7.2.3 for URL). The source code of the 2nd Screen application is an Android Studio project and it can be opened with Android Studio after downloading is finished (due to the fast development pace of Android Studio, adaptations can be necessary to start the project). Figure 10 shows the 2nd Screen project loaded in Android Studio.

The logic used by the Voice Backend can be retrieved through the Voice Dialogue repository, which is available at the GitLab (see deliverable D7.2.3 for URL). The source code of the 2nd Screen is set of resource documents used by an instance of Talkamatic Dialogue Manager (TDM). It is a collection of Python and XML files. No specific editor is required for editing the code.

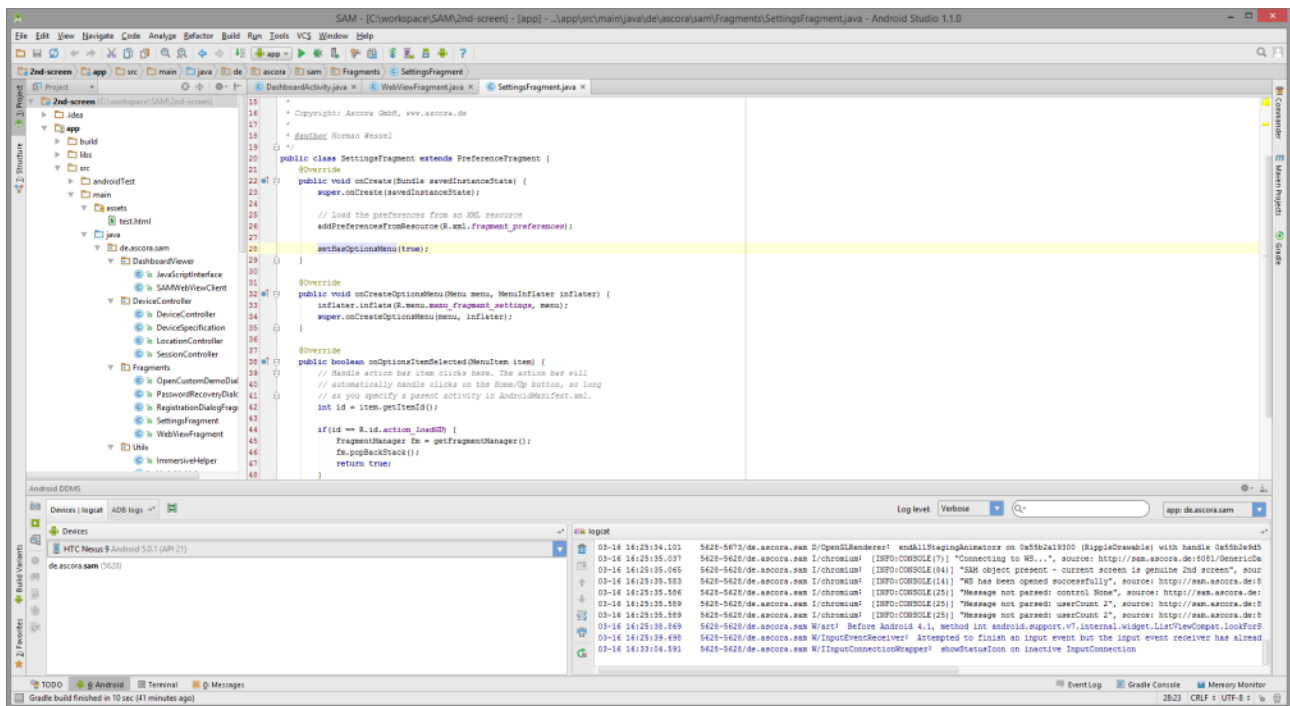


Figure 10: Android Studio IDE

3.4 Execution and Usage

This section describes how to use the prototype both as user or developer.

3.4.1 For Users

The following subsections present the different views of the 2nd Screen application.

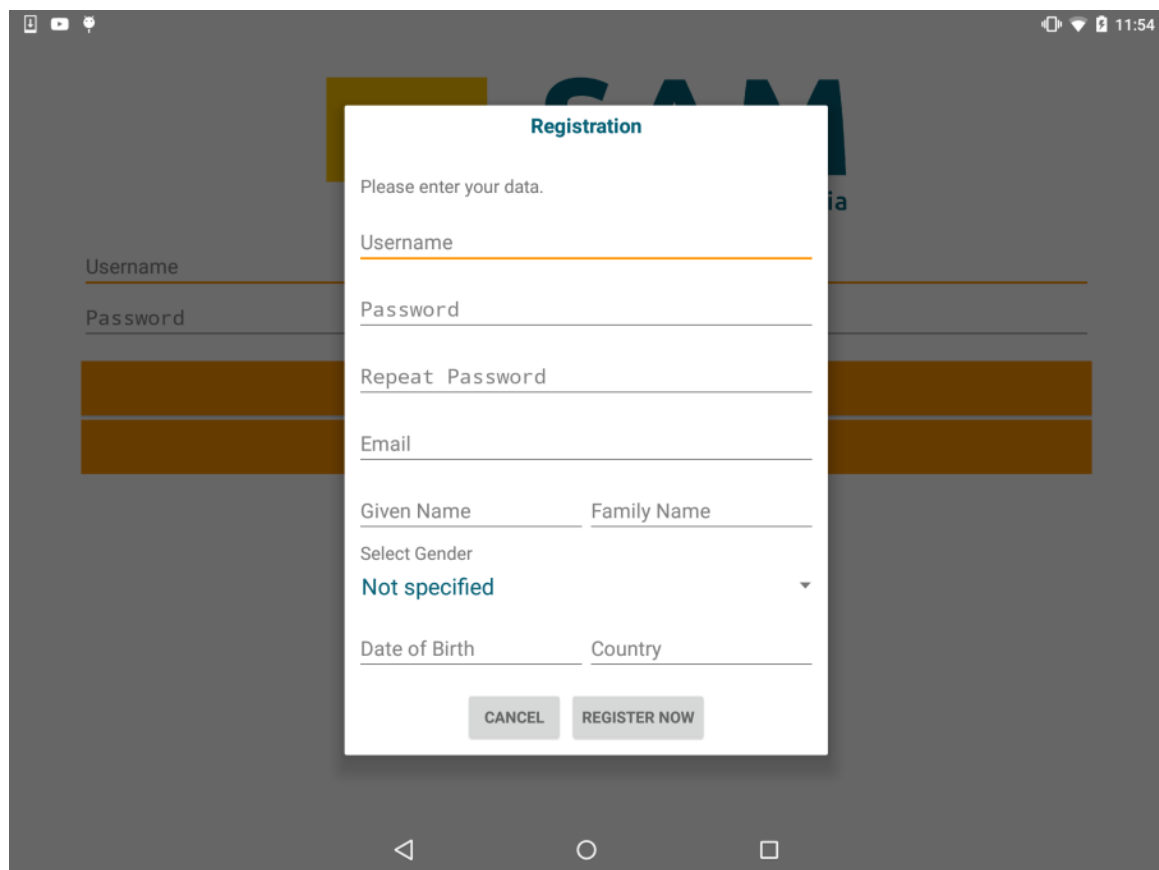
3.4.1.1 Login Mask

To use the application, the user has to have a user account. If the user is not logged in, the 2nd Screen application shows a login screen (see Figure 11). Besides the implemented login process, there is one more button which allows the user to register (see Figure 27). The user fills out the form and can finish the registration by clicking on “Register Now”. The implementation of the password recovery feature has not been integrated as it was deemed not necessary or critical for the demonstration purposes.



The image shows the SAM login screen. At the top, there is an orange status bar with icons for signal, Wi-Fi, and battery, and the time 12:25. Below this is the SAM logo, which consists of three overlapping yellow squares and the text "SAM" in large blue letters, with "socialising around media" in smaller blue text below it. The login form has two input fields: "Username" and "Password". Below these fields are two orange buttons: "SIGN IN" and "REGISTER".

Figure 11: 2nd Screen Application – Login Screen



The image shows the SAM registration screen. It features a modal dialog box titled "Registration" with the instruction "Please enter your data." The form includes the following fields: "Username", "Password", "Repeat Password", "Email", "Given Name", "Family Name", "Select Gender" (with a dropdown menu showing "Not specified"), "Date of Birth", and "Country". At the bottom of the modal are two buttons: "CANCEL" and "REGISTER NOW". The background of the screen is a blurred version of the login screen.

Figure 12: 2nd Screen Application – Registration Screen

3.4.1.2 Dashboard

After the user successfully signed in and connected to a 1st Screen device, the Generic Dashboard is displayed (see Figure 13) and ready to deliver related content. In the action bar at the top of the Generic Dashboard the remote control functions are present (see Figure 14). The remote control contains three buttons: “Play”, “Pause”, and “Stop” (Stop button is hidden in the options menu due to automatic sizing).

Additionally, settings are available in the preferences screen by selecting the options menu (⋮) in the upper right corner.



SAM is ready to deliver related content and communication streams for any SAM Video you choose.
Just select one of the SAM Videos on your TV's SAM app.

Figure 13: 2nd Screen Application – Generic Dashboard Screen



Figure 14: 2nd Screen Application – Remote Control

3.4.1.3 1st Screen Discovery

Since this third iteration, the 2nd Screen application supports the discovery of existing 1st Screen devices in the same local network (see Figure 15). This feature uses the Network Service Discovery technology provided by the Android SDK.

To search the local network for 1st Screen devices, the user has to tap on the button shown in Figure 15. This will trigger a progress bar while the search is on-going. After a short time a popup with existing 1st Screen devices will be shown (Figure 16). By clicking on a listed 1st Screen, a local connection will be established which is used for synchronisation and control means.



Figure 15: 2nd Screen Application – 1st Screen Discovery Button

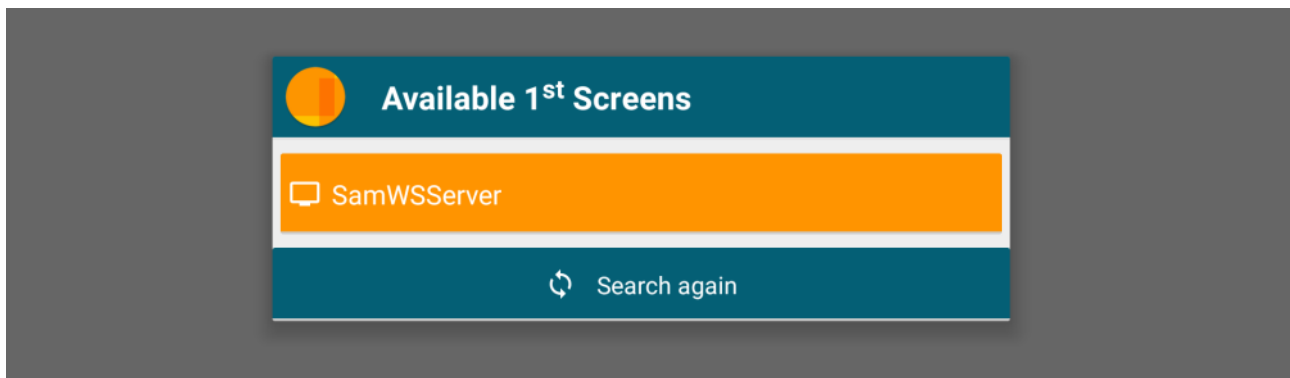


Figure 16: 2nd Screen Application – 1st Screen Discovery Popup

3.4.1.4 Voice Dialogue

Right after the application launches, a text notification is shown whenever the voice component is ready. Then, whenever the users desire to provide input they have to press the microphone button (Figure 17).



Figure 17: Voice Dialogue

There is a wide variety of requests that the user can direct to the system. For example, the user can post a tweet on Twitter (Figure 18):

SAM> What would you like to do?

User> Share my film on twitter

Sam> What comment should I post?

User> I think I really like Casino Royale

SAM> Media shared with comment I think I really like Casino Royale!

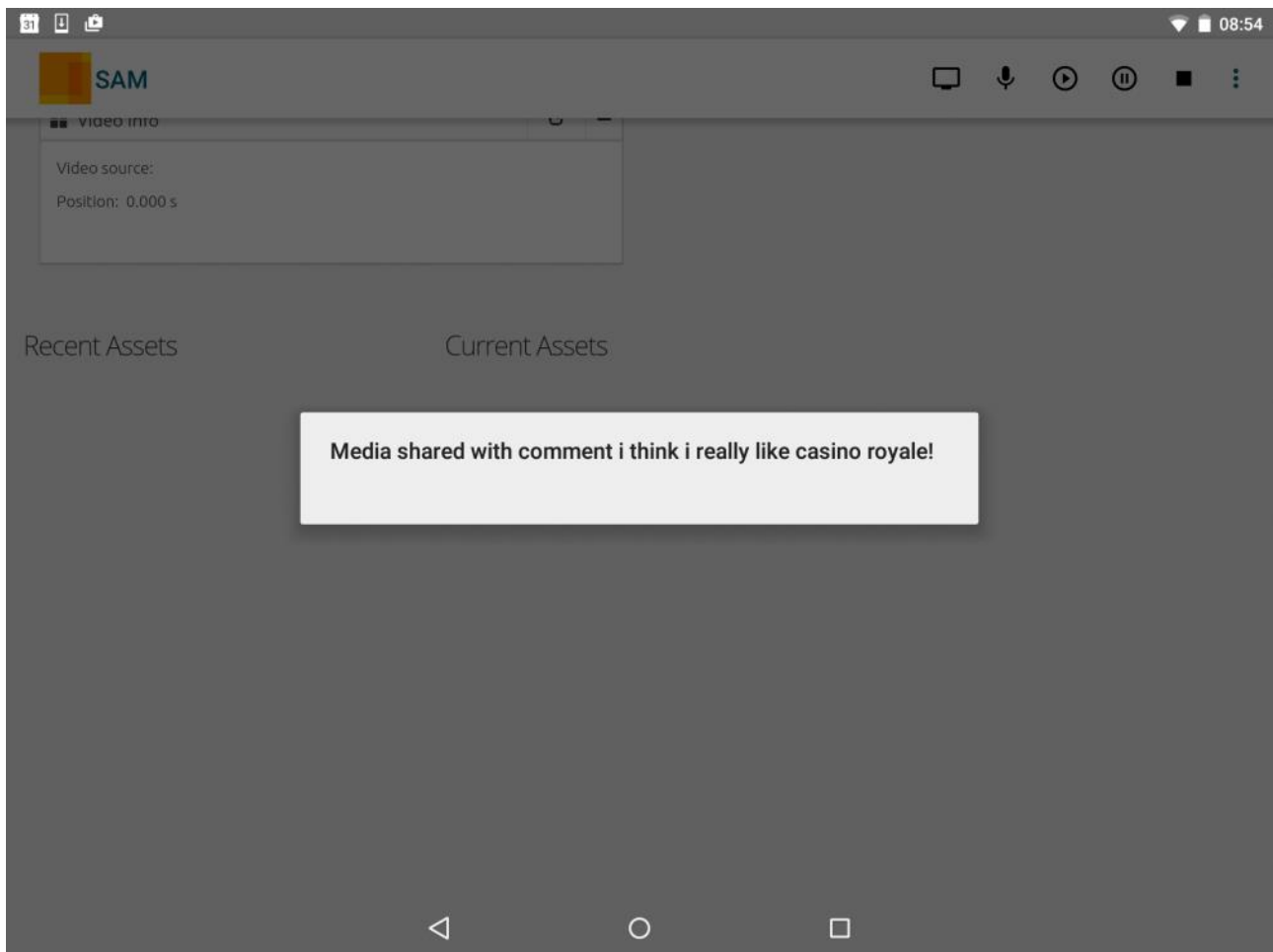


Figure 18: Media Sharing Confirmation with Dictation Results

3.4.2 For Developers

To test the 2nd Screen application without a 1st Screen Smart TV, it is possible to enable connecting to Generic Dashboard instances running on a server. By activating the options “Show Debug Menu Items in the Settings Menu” additional 1st Screen devices appear in the discovery mode.

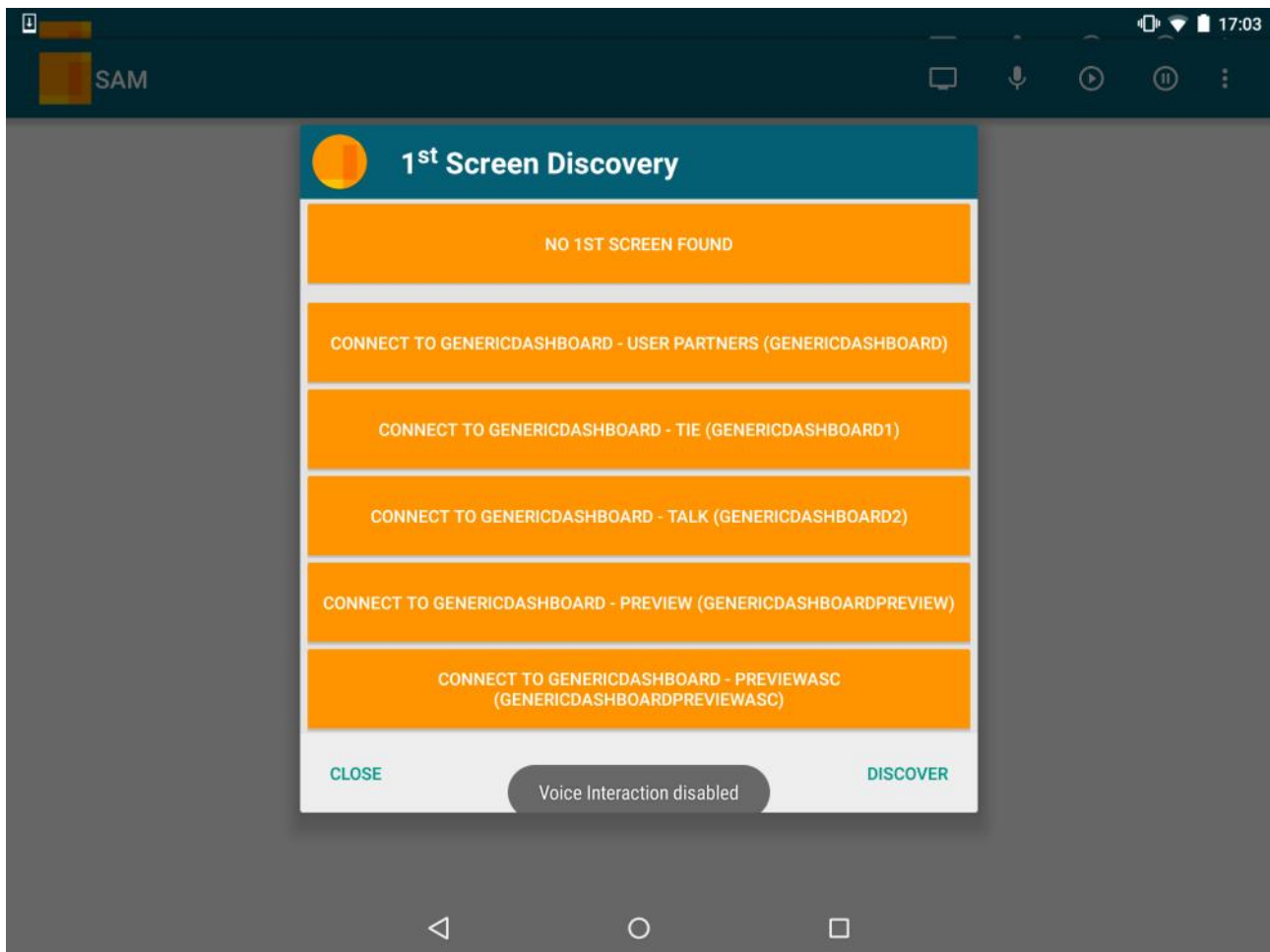


Figure 19: 2nd Screen – Discovery Popup (Debug Mode)

In the same way, the required Voice Backend for the Voice Dialogue to function can be run locally. Then, the field “Override the Voice Dialogue URL” in the App Settings has to point to the address of the local server.

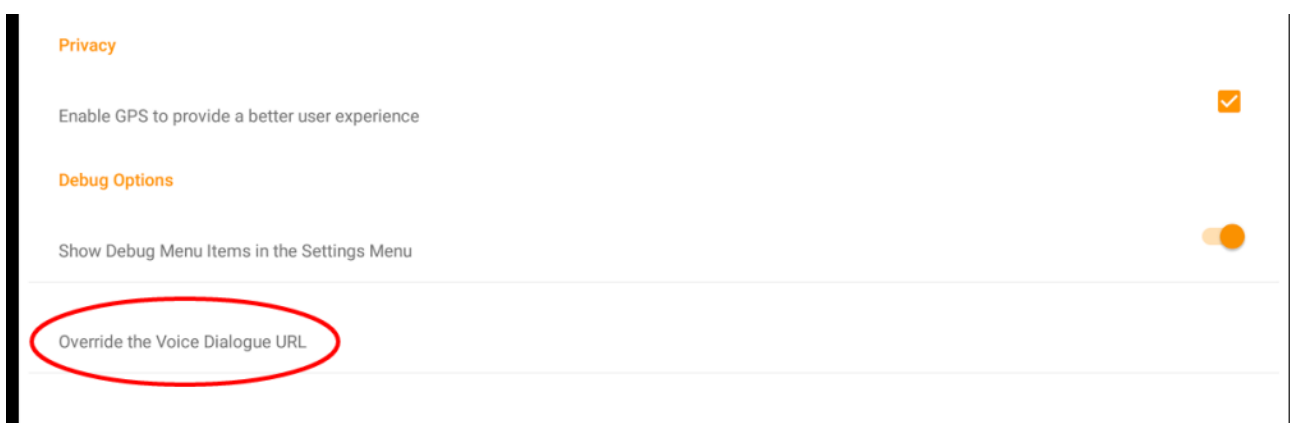


Figure 20: Overriding the Voice Dialogue URL

3.4.2.1 JavaScript Interfaces

This section provides the documentation of the JavaScript interfaces required for the communication between the Dashboard Viewer subcomponent and the within presented website. The JavaScript interfaces are injected by the Dashboard Viewer subcomponent and are available using the namespace “SAM”.

3.4.2.1.1 Get GPS Coordinates - Latitude

Get GPS Coordinates – Latitude	
Description	Returns the latitude value of the GPS sensor. Availability depends on the settings and accuracy depends on the device
Method Header	SAM.getGPSCoordinatesLatitude()
Parameters	None
Return Value	Double
Error Handling	In case of an error, 0.0 is returned
Remarks	Only available on GPS-enabled devices

Figure 21: 2nd Screen JavaScript Interface – Get GPS Coordinates – Latitude

3.4.2.1.2 Get GPS Coordinates - Longitude

Get GPS Coordinates – Longitude	
Description	Returns the longitude value of the GPS sensor. Availability depends on the settings and accuracy depends on the device
Method Header	SAM.getGPSCoordinatesLongitude()
Parameters	None
Return Value	Double
Error Handling	In case of an error, 0.0 is returned
Remarks	Only available on GPS-enabled devices

Figure 22: 2nd Screen JavaScript Interface – Get GPS Coordinates – Longitude

3.4.2.1.3 Get Display Width

Get Display Width	
Description	Returns the width of the display in the current orientation (Landscape/Portrait)
Method Header	SAM.getDisplayWidth()
Parameters	None
Return Value	Integer
Error Handling	None
Remarks	None

Figure 23: 2nd Screen JavaScript Interface – Get Display Width

3.4.2.1.4 Get Display Height

Get Display Height	
Description	Returns the height of the display in the current orientation (Landscape/Portrait)
Method Header	SAM.getDisplayHeight()
Parameters	None
Return Value	Integer
Error Handling	None
Remarks	None

Figure 24: 2nd Screen JavaScript Interface – Get Display Height

3.4.2.1.5 Get Display Density

Get Display Density	
Description	Returns the density value of the display as an integer. "For simplicity, Android groups all actual screen densities into six generalised densities: low, medium, high, extra-high, extra-extra-high, and extra-extra-extra-high." ⁵
Method Header	SAM.getDisplayDensity()
Parameters	None
Return Value	Integer
Error Handling	None
Remarks	None

Figure 25: 2nd Screen JavaScript Interface – Get Display Density

3.4.2.1.6 Get Display Density in DPI

Get Display Density DPI	
Description	Returns the density of the display in DPI (Dots per Inch)
Method Header	SAM.getDisplayDensityDpi()
Parameters	None
Return Value	Integer
Error Handling	None
Remarks	None

Figure 26: 2nd Screen JavaScript Interface – Get Display Density DPI

⁵ http://developer.android.com/guide/practices/screens_support.html#terms

3.4.2.1.7 Start Voice Interaction

Start Voice Interaction	
Description	Calls the voice interaction method in the Voice Dialogue subcomponent
Method Header	SAM.handlePTTPushed()
Parameters	None
Return Value	None
Error Handling	None
Remarks	None

Figure 27: 2nd Screen JavaScript Interface – Start Voice Interaction

3.4.2.1.8 Logout

Logout	
Description	Logs out the user and invalidates the token
Method Header	SAM.logout()
Parameters	None
Return Value	None
Error Handling	None
Remarks	None

Figure 28: 2nd Screen JavaScript Interface – Logout

3.4.2.1.9 Get User Token

Get User Token	
Description	Returns the current user token
Method Header	SAM.getUserToken()
Parameters	None
Return Value	String
Error Handling	In case of an error, null is returned
Remarks	None

Figure 29: 2nd Screen JavaScript Interface – Get User Token

3.4.2.1.10 Get Logged-In Username

Get Logged-In Username	
Description	Returns the name of the current user
Method Header	SAM.getLoggedInUserName()
Parameters	None
Return Value	String
Error Handling	In case of an error, null is returned
Remarks	None

Figure 30: 2nd Screen JavaScript Interface – Get Logged-In Username

3.4.2.1.11 Get SAM ID

Get SAM ID	
Description	Returns the SAM ID of the current user
Method Header	SAM.getSAMID()
Parameters	None
Return Value	String
Error Handling	In case of an error, null is returned
Remarks	None

Figure 31: 2nd Screen JavaScript Interface – Get SAM ID

3.4.2.1.12 Get Device Model

Get Device Model	
Description	Returns the name of the device model (Used by Context Control)
Method Header	SAM.getDeviceModel()
Parameters	None
Return Value	String
Error Handling	In case of an error, null is returned
Remarks	None

Figure 32: 2nd Screen JavaScript Interface – Get Device Model

3.4.2.1.13 Get Device Manufacturer

Get Device Manufacturer	
Description	Returns the name of the device manufacturer (Used by Context Control)
Method Header	SAM.getDeviceManufacturer()
Parameters	None
Return Value	String
Error Handling	In case of an error, null is returned
Remarks	None

Figure 33: 2nd Screen JavaScript Interface – Get Device Manufacturer

3.4.2.1.14 Set Video ID

Set Video ID	
Description	Sets the video ID of the video (used for Voice Dialogue)
Method Header	SAM.setVideoId(string id)
Parameters	String
Return Value	None
Error Handling	In case of an error, video id value is set to null
Remarks	None

Figure 34: 2nd Screen JavaScript Interface – Set Video ID

3.4.2.1.15 Set Video Timestamp

Set Video Timestamp	
Description	Sets the timestamp of the video (used for Voice Dialogue)
Method Header	SAM.setVideoTimestamp(string timestamp)
Parameters	String
Return Value	None
Error Handling	In case of an error, video timestamp value is set to -1.0
Remarks	None

Figure 35: 2nd Screen JavaScript Interface – Set Video Timestamp

3.4.2.1.16 Provide Social Content

Provide Social Content	
Description	Forwards social content to the Voice Dialogue
Method Header	SAM.provideSocialContent(string socialNetwork, string socialContentAsJson)
Parameters	String
Return Value	None
Error Handling	None
Remarks	None

Figure 36: 2nd Screen JavaScript Interface – Provide Social Content

3.5 Limitations

The following limitations for the final prototype are present:

- **Password Recovery User Dialog:** This optional feature has not been implemented.
- **GUI Issues:** At the trials a small amount of Android devices (in particular from Sony and Huawei) had issues displaying buttons in the discovery popup. Due to the heterogeneity of Android devices further research was not conducted.

Additionally, the Voice Dialogue (VD) interaction has the following limitations:

- **Widget Management:** The functionality is no longer supported.
- **Visual Guide:** During the Voice user trials it became apparent that Voice dialogue is missing a visual guide which can inform the users about all available interactions

3.6 Research Background

Due to the nature of the component, no research background is available except of the Voice Dialogue subcomponent, where papers taken into consideration are provided in Figure 37. These papers point to research that led and influenced the creation of Talkamatic Dialogue Manager (TDM) upon which the Voice Dialogue component is built.

The integration of VD with the SAM application is done in a way that highlights TDM's capabilities and thus exhibits the concepts, which exist in TDM.

Source	Subcomponent	Description
[LAR02] Larsson, S., (2002). Issue-based Dialogue Management, Gothenburg Monographs in Linguistic	Voice Dialogue	Dissertation thesis describing dialogue design principles on which TDM was built
[BCL05] B. Bringert, R. Cooper, P. Ljunglöf, A. Ranta, Multimodal Dialogue System Grammars. Proceedings of DIALOR'05, Ninth Workshop on the Semantics and Pragmatics of Dialogue, Nancy, France, June 9-11, 2005, 2005	Voice Dialogue	Shows how mouse clicks can be integrated in GF grammars alongside with speech input

Figure 37: Research Background 2nd Screen

3.7 Target Performance

This section contains the key performance indicators (KPI) (see Section 3) and user experience measurement tasks (see Section 3) for this component.

3.7.1 Component KPIs

For this component, the following KPIs have been defined:

Topic	Description	Target KPI
Discovery duration	The 2 nd Screen application will enable the detection of 1 st Screen devices in the local network. A pairing with available 1 st Screen devices can then be done afterwards to receive additional content regarding the video element running on the 1 st Screen device.	The discovery of all available 1 st Screen devices should be achieved in less than 10 seconds.
Connection success rate	When a 1 st Screen discovers a 2 nd Screen (or vice versa), setting up a connection should always be successful.	The discovery of all available 1 st Screen devices should be achieved in 90% of the test cases.

Figure 38: Target Performance 2nd Screen

These performance targets are reviewed against the performance achieved in the third prototype. The following results have been achieved at the end of the second iteration:

- **Discovery duration:** The average discovery time has been 278.8 ms. This result is based on timestamps extracted from logs. Independently of the discovery time, the discovery popup will be shown after 5 seconds to make sure all available local first screen devices are visible.
- **Connection success rate:** The discovery achievement rate with our test devices is 100%. Issues at the trials, where 1st Screen devices were not found seem to be linked to system customisations of the Android OS by the device manufacturer, i.e. Huawei.

3.7.2 User Experience Measurements Tasks

In addition to the KPIs in Section 3.7.1, this work package provides user tasks, which are input for measuring the subjective user experience in a uniform way. For each of the tasks below the task-specific KPIs, defined in Annex A: User Research, will be measured.

Task	Description	Status
Pairing	The user has to successfully pair the 2 nd Screen device with the 1st Screen device.	Discovery feature have been implemented and used in the trials.
Control Video Element	The user has to successfully pause, and then shift to a specific time in the video element and then start the video element again.	Video control features have been implemented. During the trials, the video control elements have been disabled to prevent interruptions caused by trial participants.
Enable GPS	The user has to successfully enable the GPS location feature in the settings menu.	GPS feature have been implemented and used in the trials.
Enable Notifications	The user has to successfully enable the notifications feature in the settings menu.	Notification have been implemented but due to lower priority in deep testing has been cancelled.
Login	The user has to successfully reset the password by using the recovery feature and login to the SAM application.	Login as well as registration feature have been implemented. Due to prioritising and time issues, the password recovery feature has not been implemented.
Post on a SAM community using Voice Dialogue	The user has to successfully post a comment in a SAM dynamic community using Voice Dialogue	Feature has been implemented.
Control Video Element using Voice Dialogue	The user has to successfully pause, and then start the video element again using Voice Dialogue.	Feature has been implemented.

Figure 39: Target Performance 2nd Screen – Task-Specific Measurements

Both KPIs that regard Voice Dialogue have been extensively discussed in the report from the End User Voice Trials deliverable (D8.4.2, Section 5). In general, the results were satisfactory with some users even actively reporting that the features were easy to use.

3.8 Summary

This section provides a description of the third prototype of the 2nd Screen component developed in task T7.2 2nd Screen Media Interaction. The main outcome of this task is the 2nd Screen Android application and the integrated Voice Dialogue subcomponent including the connected backend system. This prototype is the last of three iterations planned for this component and the goal is to cover the remaining 40% of the requirements of the component (see Section B 1.3.3.7 of the DOW for additional information on the effort distribution for this component in the lifespan of the project).

The requirements necessary for both users and developers have been presented including installation instructions. The last section has been dedicated to describing the limitations and performance data of this prototype.

4 SAM Multi-Device Dashboard

This section describes the software deliverable D7.9.4, which is the fourth prototype release of the SAM Multi-Device Dashboard component.

4.1 Scope and Relationship

The Generic Dashboard provides the end user with a Graphical User Interface for the interaction with the SAM Platform. The Generic Dashboard is presented on both 1st Screen and 2nd Screen devices, on top of which each device adds its specific subcomponents. Figure 40 shows the different subcomponents of the Generic Dashboard, the logical connections that have been established between them and the relations with other components and actors of SAM Platform.

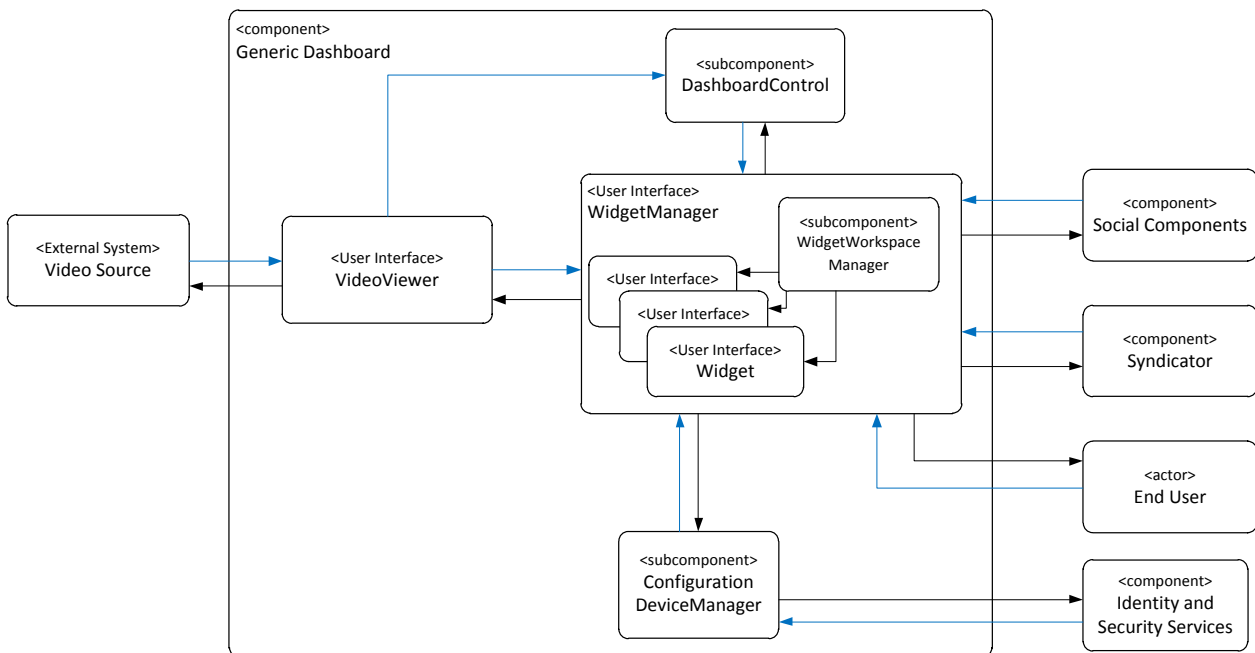


Figure 40 Generic Dashboard Subcomponents and its Relationships

For further description of the functional and technical foundations of these subcomponents, please revisit documents D3.2.1 Section 4.13.2 (Architecture), D3.2.2 Section 4.14.1 (Functional Specification) or D3.3.1 Section 3.14.1 (Technical Specification).

The first prototype of the Generic Dashboard focused on providing basic functionalities to the End User, such as listing the available video streams and playback of the selected one. The Dashboard itself had been populated with several Widgets presenting basic functionalities and sample static content.

A summary of the tasks carried out for each subcomponent of the first version of the prototype is shown in the following table:

Subcomponent	Task
Video Viewer	Obtain a list of available streams from video source, select and stream video
Dashboard Control	Provide JavaScript interfaces for communication
Widget	Display and control user interface widgets on the Dashboard
Widget Manager	Provide JavaScript interfaces for updating widgets content
Widget Workspace Manager	Display and control support of the Widgets. Display SAM settings
Configuration Device Manager	Provide JavaScript interfaces for device information

Figure 41: Tasks Carried Out for the First Prototype of T7.4.

In the second prototype, the Generic Dashboard was extended with a link to the Syndicator component, which allows displaying dynamic Widgets. The number of Widget types was extended with Facebook, Twitter, Related Products and Wikipedia page Widgets allowing providing the End User with a more immersive and rich experience. Adding the interfaces to authenticate the user based allowed for gathering contextual information about user interactions with the Dashboard and consumed assets.

A summary of the tasks carried out for each subcomponent of the second version of the prototype is shown in the following table:

Subcomponent	Task
Video Viewer	No tasks planned for this prototype
Dashboard Control	Improved messaging mechanism between widgets, allowing for more reliable communication between 1 st and 2 nd Screen
Widget	<p>Implemented particular widgets for various functionalities, in particular:</p> <ul style="list-style-type: none"> • Widgets showing relevant social information from Facebook and Twitter related to presented video, allowing to post new posts / tweets • Related Product Widget offering information about possible related marketing product End User can buy • Wikipedia Widget displaying to the End User information from Wikipedia on related topic to currently played video • Related Assets widget displaying past Widgets that are moved away from the central point of the Dashboard, allowing the End User to browse its contents
Widget Manager	<ul style="list-style-type: none"> • Integration with Syndicator to receive and update Widgets with related content to currently viewed video • Integration with Social Components to feed Twitter and Facebook Widgets. This subcomponent is in charge of implementing the communication with the Data API services component • Upgrades in Generic Dashboard services responsible for providing the list of assets related to the main video, obtaining the Widgets html and providing the list of videos to show in the 1st Screen • Implementation of caching mechanism in order to improve the performance of the communication between Generic Dashboard and other SAM components
Widget Workspace Manager	<ul style="list-style-type: none"> • Updating the Widgets accordingly to scheduled timeline obtained from Syndicator component • Advanced Widget handling responsible for prefetching of Widget HTML code & moving it afterwards to Related Assets Widget
Configuration Device Manager	Obtaining user token from 2 nd Screen devices and making it available to all widgets that need this information to track End User preferences and/or activities on the Generic Dashboard on the 1 st and 2 nd Screen
Context Listener	<ul style="list-style-type: none"> • First implementation of an AngularJS client for communicating with the Context Control services. Through this client, contextual information (e.g. location and consumed assets) about the users who interact with the dashboard are pushed to Context Control component • In addition, information on the users' profiles and their behaviour in the SAM environment is available (e.g. consumed assets), which for the proof of concept of the current prototype, provides the Top10 consumed assets

Figure 42: Tasks Carried Out for the second Prototype of T7.4.

In the third prototype, the Generic Dashboard has been modified with new overlay containing all the passed assets, so that user can always go back to the displayed content. Furthermore some optimisations were done in order to support most of the functionality in standard web-browser for true multiplatform support. A couple of layout changes and inclusion of device information (resolution, GPS position) in communication with external components make also considerable impact on delivering good End User experience.

A summary of the tasks carried out for each subcomponent of the third version of the prototype is shown in the following table:

Subcomponent	Task
Video Viewer	No tasks planned for the third prototype
Dashboard Control	Implemented communication to update the Discover Assets Widgets
Widget	No tasks planned for the third prototype
Widget Manager	Social service moved exclusively to 2 nd Screen
Widget Workspace Manager	<ul style="list-style-type: none"> Layout changes Optimised for browser view Recent assets were moved to overlay instead of removing them permanently
Configuration Device Manager	Included device related information in context and syndicator
Context Listener	<ul style="list-style-type: none"> Update of the AngularJS client for communicating with the Context Control services. Useful user interactions (e.g. consume asset, comment asset) are pushed to Context Control component Overall statistics and information on the users' behaviour in the SAM environment is available (e.g. most interactive users, most liked assets), a proof of concept of this prototype

Figure 43: Tasks Carried Out for the Third Prototype of T7.4.

In the fourth and final prototype, the Generic Dashboard has been updated to be in line with the latest SAM styling guidelines. The video carousel area has been simplified and is now only showing video assets. Video and device information widgets have been removed to keep focus on a better user experience. Furthermore the 2nd Screen UI has been modified to have a two-tabbed design rather than having a current and previous assets view. Last major change available in the delivery is the user login functionality for 2nd Screen devices accessing the SAM Platform with a browser instead of an APK.

Subcomponent	Task
Video Viewer	<p>Two new features which are configurable in the App Settings menu of the 1st Screen UI:</p> <ul style="list-style-type: none"> • Implementation of feature “Wait for all connected devices” before starting video playback • Implementation of having the video controls availability <p>Stability improvements:</p> <ul style="list-style-type: none"> • In the original source code displaying loading animation was done by changing CSS properties for the different HTML elements by changing the display value for each element. In the fourth prototype this has been changed to a view state handler function which allows only one element to be visible at a time <p>General changes:</p> <ul style="list-style-type: none"> • Moved syndicator call to retrieve video list from Marketplace V3
Dashboard Control	Further improved messaging mechanism between widgets, allowing for more reliable communication between 1 st and 2 nd Screen
Widget Manager	<p>Introduces error handling for following scenarios:</p> <ul style="list-style-type: none"> • Widgets not shown because beginTime and endTime exceed the duration of the movie. At the same time report in the console that this error occurred and requires correct configuration of the widget data • Moving all widgets back to the current view when the movie ends <p>Introduces additional code to cover the UI changes requested after the first TRIALS:</p> <ul style="list-style-type: none"> • Moving the widgets to the widget tab and moving social widgets to the social tab • Disabled the “move to previous assets” functionality
Widget Workspace Manager	<ul style="list-style-type: none"> • Updating the Widgets accordingly to scheduled timeline obtained from Syndicator component, based on the new code available in the Widget Manager
Configuration Device Manager	Added functionality which is covering for the new user login feature. Removed the default AlexUser credentials (fallback mechanism). Introduced new diversity rules to cover login via web UI versus 2nd Screen APK UI
Context Listener	No tasks for this component

Figure 44: Tasks Carried Out for the Fourth Prototype of T7.4.

4.2 Requirements and Preparations

This section provides information on technical and non-technical requirements for users as well as for developers. The Dashboard has dependencies with the Data API services (Syndicator). For this reason it is necessary to check that the services of the Syndicator component are working properly. The necessary documentation to check the status of these services is D5.9.2 section 6.2.

4.2.1 For Users

The Generic Dashboard is an embedded subcomponent in the 1st and 2nd Screen applications. For specific requirements for running 1st Screen application see document D7.2.3, for 2nd Screen application see D7.3.3.

4.2.2 For Developers

For developers it is strongly recommended to use Apache HTTP server⁶ for hosting the web application. Additionally, using the devices listed in documents D7.2.3 and D7.3.3 is advised, but not a necessity.

4.3 Installation (Deployment)

This section describes the Installation process of Generic Dashboard for Users and Developers.

4.3.1 For Users

It is not necessary to separately install the Generic Dashboard, because it is embedded into 1st and 2nd Screen applications. The installation process of those applications is described in documents D7.3.3 and D7.2.3 respectively.

4.3.2 For Developers

Developers have to download the source code first (Deliverable D7.4.3 contains the required information). The downloaded source code must be placed in the web root Assets folder of 1st Screen application sources. The source code of the Generic Dashboard application contains HTML and JavaScript files and can be edited using any text editor.

4.4 Execution and Usage

This section describes how to use the different subcomponents of the prototype.

4.4.1 For Users

The usage of Generic Dashboard depends on the type of the device. Please consult the corresponding sections describing usage of the 1st Screen (D7.1.3) and the 2nd Screen (D7.2.3) devices.

4.4.2 For Developers

The Generic Dashboard subcomponents have been implemented as AngularJS services⁷ and controllers⁸. Widgets can use these services for messaging, receiving content and social components communication. The communication between Widgets and Syndicator or Social Components is handled by the Widget Manager service. The communication between Widgets, present either on the same Generic Dashboard or across 1st and 2nd Screen devices, is handled by the Dashboard Control service. The following JavaScript methods can be used from Widgets to specify the type of message they would like to receive and define the call back function to receive and process them. The same holds for content update messages. Each Widget can then use the “sendMessage” function to send the message of appropriate type to all subscribed Widgets, irrespectively which screen they are placed on.

⁶ <http://httpd.apache.org/>

⁷ AngularJS Services: <https://docs.angularjs.org/guide/services>

⁸ AngularJS Controllers: <https://docs.angularjs.org/guide/controller>

4.4.2.1 Dashboard Control Service

Subscribe For Message	
Description	A widget can subscribe to receive messages of a certain type
Method Header	dashboardControl.subscribeForMessage(widget, messageType)
Parameters	widget – subscribing widget object reference messageType – type of the message the widget is subscribing for
Return Value	None
Error Handling	None
Remarks	When a message of the messageType is received, the widget controller method receiveMessage is called.

Figure 45: Dashboard Control Service Interface – Subscribe For Message

Send Message	
Description	A widget can send message to other widgets (also other widgets present on other devices)
Method Header	dashboardControl.sendMessage(messageType, message)
Parameters	messageType – type of the message message – message to be send
Return Value	None
Error Handling	None
Remarks	None

Figure 46: Dashboard Control Service Interface – Send Message

4.4.2.2 Widget Manager Service

Subscribe for Content	
Description	A widget can subscribe to receive content of a certain type
Method Header	widgetManager.subscribeForContent(widget, contentType, frequency)
Parameters	widget – subscribing widget reference contentType – type of the content the widget is subscribing for frequency – how often the widget wants to receive the updates (in seconds); 0 = immediately
Return Value	None
Error Handling	None
Remarks	When content of the contentType is received, the widget controller method updateNotification is called

Figure 47: Widget Manager Service Interface – Subscribe For Content

4.4.2.3 Widget controller

Widget controller itself must implement following methods:

Receive Message	
Description	Widget receive message from other widgets (also other widgets present on other devices)
Method Header	widget.receiveMessage(messageType, message)
Parameters	messageType – type of the message message – message received
Return Value	None
Error Handling	None
Remarks	None

Figure 48: Widget Controller Interface – Receive Message

Receive Content	
Description	Widget receive content from Syndicator
Method Header	widget.updateNotification(contentType, content)
Parameters	contentType – type of the content content – content received
Return Value	None
Error Handling	None
Remarks	None

Figure 49: Widget Controller Interface – Receive Content

4.4.2.4 1st Screen components

Due to the nature of SmartTV applications, the Generic Dashboard also includes several subcomponents of the 1st Screen. The Dashboard Viewer of the 1st Screen invokes the SmartTV browser in order to display the Generic Dashboard component on the 1st Screen device. The SmartTV browser communicates with the Inter-Device-Communication and 1st Screen Interaction subcomponents of the 1st Screen (included in the Generic Dashboard) and adds the needed interfaces, as illustrated in Figure 50.

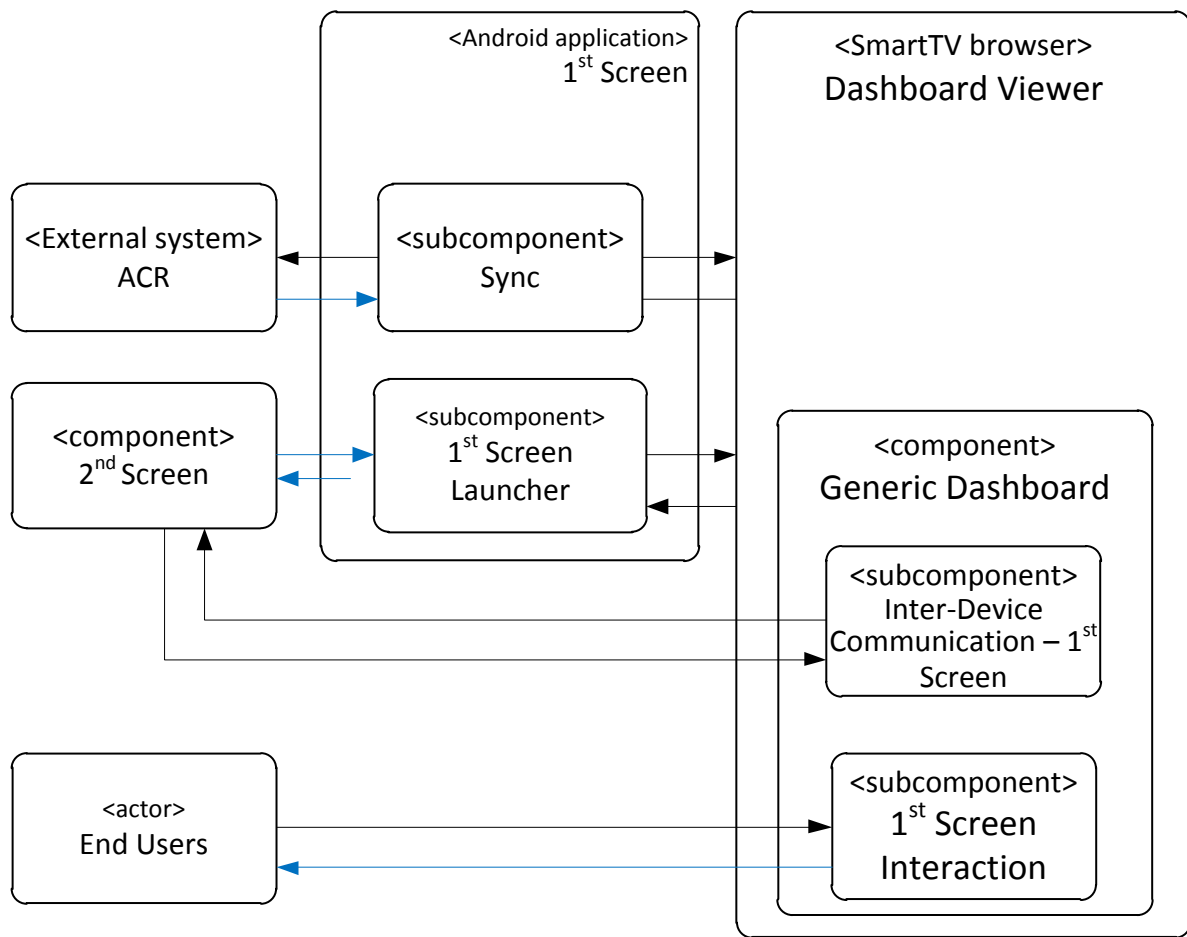


Figure 50: SmartTV Application Architecture

The Inter-Device-Communication is implemented as an AngularJS service. It is responsible for managing the connection to WebSocket server and exchange broadcast messages between all connected instances. Currently WebSocket is a locally embedded server and integrated into the Inter-Device-Communication component, as mentioned in Section 3 of D7.9.3

Each Widget subcomponent that is displayed on the Generic Dashboard is built from two parts: an HTML5 template and its AngularJS Controller (described above). The sample implementation of both is provided in the source code and can be found under the path in the source tree specified in Figure 51 below.

```

<source-root>/html/app/dashboard/widgets/widget.html
<source-root>/html/app/dashboard/widgets/widget.js
  
```

Figure 51: Sample Widget Implementation Source Paths

Those template files are loaded for each widget by the Widget Workspace Manager during the loading process of the Generic Dashboard. By default, the Widget controller is collapsing the Widget on the Generic Dashboard after 2 seconds from the last activity. This timeout can be set per individual Widget inside its controller.

The Video Viewer subcomponent is a specialised Widget, containing an HTML5 video element, used to display videos to the End User and to obtain various information about the video stream. Most importantly, the video ID and current timestamp are obtained via

HTML5 DOM and shared with the Widget Managers of the connected 1st and 2nd Screens using Inter-Device-Communication.

4.4.2.5 Context Control

The Generic Dashboard is integrated with the Context Control component of SAM in order to provide information regarding the users' behaviour during their interaction with SAM system, using 1st or 2nd Screens. This information refers on the users' location and devices, as well as implicit or explicit information about their profiles and preferences such as Assets consumed, scrolled, liked, etc. In that sense, the context control can form sophisticated user profiles and groups, based on their behaviour within the SAM environment. For the communication with the context control and providing or consuming contextual information, the respective REST API is used. However, for the Dashboard and its related components, a JS library has been implemented that provides access to the API through an AngularJS service.

The above process results into a personalisation of the SAM environment based on user profile. Thus, upon each user login to Generic Dashboard and selection of a video, Context Control provides most relevant widgets to the Dashboard, in order to be shown in a "Suggested Content" window, as recommendations.

4.5 Limitations

As for the third prototype, the fourth prototype has been concentrating on further integration and communication with 1st and 2nd Screen components as well as other SAM components (e.g. Syndicator). During the first Trials we noticed issues related to performance. Hence, the following views/functions have been provided as mock-ups:

- **Generic Dashboard settings:** e-commerce, advertising – as currently we don't display any advertisements nor the e-commerce components those options do not have any effect

4.5.1 Planned tasks – Future improvements

The current status is a stable status. However, some unforeseen limitations were uncovered during validations. The behaviour of the 1st Screen application on a TV or a web browser differs per execution platform when one looks at the runtime performance. This is particularly so when there are more and more devices connected to the 1st Screen. There also looks to be a dependency with the number of physical devices connected or multiple tabs from the same devices connecting to the 1st Screen. Further investigation would be needed to make it as a final consumer product which is scalable.

4.6 Research Background

For the current prototype implementation and the overall approach, the following related research work was taken into consideration:

Source	Subcomponent	Description
[BAS13] Bassbouss, L.; Tritschler, M.; Steglich, S.; Tanaka, K.; Miyazaki, Y., "Towards a Multi-screen Application Model for the Web," Computer Software and Applications Conference Workshops (COMPSACW), 2013 IEEE 37th Annual , vol., no., pp.528,533, 22-26 July 2013	General	This paper has been considered to implement Generic Dashboard using HTML5
[ETS15] ETSI TS 102 796 V1.3.1 / HbbTV 2.0 (2015-10)	General	HbbTV specification http://www.etsi.org/deliver/etsi_ts%5C102700_102799%5C102796%5C01.03.01_60%5Cts_102796v010301p.pdf
[WCJ14] Lee, Wei-Po, Che Kaoli, and Jhih-Yuan Huang. "A smart TV system with body-gesture control, tag-based rating and context-aware recommendation." Knowledge-Based Systems 56 (2014): 167-178.	Context Control	Related work on the acquisition of contextual information and its usage
[MAY14] Maynard, G. HbbTV and Multi-Screen Strategy. HbbTV Symposium Asia. Singapore 18 June 2014.	Dashboard	Dashboard principles and examples http://www.ipnetcom.at/upload/News/multiscreen.pdf

Figure 52 Consulted Research References

4.7 Target Performance

The performance measurement of the 1st Screen prototype will be measured accordingly to the defined KPIs (described in section 4.7.1) with additional End User experience measurements (described in section 4.7.2).

4.7.1 Component KPIs

For this component the following Key Performance Indicators (KPIs) have been defined:

Topic	Description	Target KPI	Status
Timeliness of related content	The time it takes for widgets to be updated and for the Inter-Widget Communication will affect the user's experience and thus needs to be low.	Time difference between video trigger and appearance of related content should be less than 5 sec.	Multi-site testing: ~4 sec (measure with 20 clients)
Reusability	The Generic Dashboard component will be used on both 1 st and 2 nd Screen devices, so it needs to be easy to integrate in these components.	70% of the common components of the 1 st and 2 nd Screen should be implemented in the Multi-Device Dashboard	Not measurable in details, but at least 75% is Generic for both 1 st as 2 nd Screen.

Figure 53: Target Performance Generic Dashboard

Timeliness of related content: Measured by using console write lines at start and end of request for widget loading / video play state changes.

Reusability: The availability of code cannot be measured. Most of the UI is based on a Generic Dashboard approach where menus and visible areas are put into areas which are common for both 1st Screen and 2nd Screen components.

4.7.2 User Experience Measurements Tasks

Additional to the KPIs in Section 4.7.1, this work package provides user tasks, which are input for measuring the subjective user experience in a uniform way. For each of the tasks below the task-specific KPIs, defined in Annex A: User Research, will be measured.

Task	Description	Status
Collapse widgets	The user needs to collapse a widget	Users were observed to perform action, using the “More” button
Check related information	The user needs to point out where he can find related information about the video	During user trials, user action has been observed
Send tweet	The user needs to send a tweet about the show he is watching in the Dynamic Social Community	Operation was observed during user trials
Control a Widget	The user has to perform a Widget function (e.g. share the current film on Facebook) successfully	Operation was observed during user trials

Figure 54: Target Performance Multi-Device Dashboard – Task-Specific Measurements

4.8 Summary

This section provides a description of the fourth prototype of the Generic Dashboard component developed in task T.7.4 Multi-Device Dashboard. The main outcome of this task is the software of the Generic Dashboard. This prototype of T7.4 extends user interface elements and their functionalities. This prototype is the last of four iterations planned for this component and the goal of this prototype is to cover 100% of the requirements of the component (see Deliverable D7.4.3 for additional information for this component in the lifespan of the project).

The most important goals reached during this fourth prototype have been:

- Discover Assets Widgets, reacting on user behaviour in real-time
- Recent Assets area, allowing user to go back to the content that moved away from current video timeline
- Optimised experience in browser, for platform where Android app cannot be used
- New UI based on Widget and Social tabs on the 2nd Screen UI. All widget content is being displayed on a single view
- QR code generator for 2nd Screen devices connecting via web browser
- Login functionality for 2nd Screen devices connected via a web browser rather than an APK
- Performance improvements for web socket server behaviour and communication between 1st and 2nd Screen

The requirements necessary for both users and developers have been presented including installation instructions. The last section has been dedicated to describing the limitations of the current prototype and describing the possible extensions for future use.

5 Document Summary

This document, D7.9.4 is the final release of the deliverable series D7.9.x to provide insight of the prototypes of software deliverables D7.1.3, D7.2.3, D7.3.3 and D7.4.3. This document provides information about the performed tasks in Work Package 7:

- T7.2 2nd Screen Media Interaction (Section 3)
- T7.4 Multi-Device Dashboard (Section 4)

The information for each task provided contains:

- Scope of the pilot implementation, its purpose and the main relationships with other modules implemented in SAM
- Information needed to deal with the pilot in terms of technical and non-technical requirements, software to be installed, etc.
- Steps needed to install the pilot software and process to build it from source code
- Different screens and actions implemented at the pilot itself, ways to access it, and to test the different implemented options
- Current pilot limitations and the expected improvements
- Papers and other scientific information considered
- Key Performance Indicators (KPI's) for the SAM component
- Conclusion of the implementation of the last prototype

References

- [WDJ11] Wilson, S., Daniel, F., Jugel, U., Soi, S. (2011). Orchestrated User Interface Mashups Using W3C Widgets. Proceedings of Composable Web 2011.
- [BWS13] Balasubramanee, V. Wimalasena, C. Singh, R. Pierce, M. (2013) Twitter bootstrap and AngularJS: Frontend frameworks to expedite science gateway development
- [KAA07] Christian Kaar (2007), An introduction to Widgets with particular emphasis on Mobile Widgets
- [LAR02] Larsson, S., (2002). Issue-based Dialogue Management, Gothenburg Monographs in Linguistic
- [BCL05] B. Bringert, R. Cooper, P. Ljunglöf, A. Ranta, Multimodal Dialogue System Grammars. Proceedings of DIALOR'05, Ninth Workshop on the Semantics and Pragmatics of Dialogue, Nancy, France, June 9-11, 2005, 2005
- [BRI07] B. Bringert. Speech Recognition Grammar Compilation in Grammatical Framework SPEECHGRAM 2007: ACL Workshop on Grammar-Based Approaches to Spoken Language Processing, June 29, 2007, Prague. 2007.
- [ZIE13] Ziegler, C., "Second screen for HbbTV — Automatic application launch and app-to-app communication enabling novel TV programme related second-screen scenarios," Consumer Electronics Berlin (ICCE-Berlin), 2013. ICCEBerlin 2013. IEEE Third International Conference on, vol., no., pp.1, 5, 9-11 Sept. 2013
- [VAN12] Vanattenhoven, J., Geerts, D. (2012). Second-Screen Use in the Home: An Ethnographic Study. In Proceedings 3rd International Workshop on Future Television, EuroITV 2012 (p. 12).
- [VAN14] Vanattenhoven, J., Geerts, D., De Grooff, D. (2014). Television Experience Insights from HbbTV. In proceedings of the International Workshop on Interactive Content Consumption, Newcastle upon Tyne
- [BAS13] Bassbouss, L.; Tritschler, M.; Steglich, S.; Tanaka, K.; Miyazaki, Y., "Towards a Multi-screen Application Model for the Web," Computer Software and Applications Conference Workshops (COMPSACW), 2013 IEEE 37th Annual , vol., no., pp.528,533, 22-26 July 2013
- [ETS15] ETSI TS 102 796 V1.3.1 / HbbTV 2.0 (2015-10)
- [WCJ14] Lee, Wei-Po, Che Kaoli, and Jhih-Yuan Huang. "A smart TV system with body-gesture control, tag-based rating and context-aware recommendation." Knowledge-Based Systems 56 (2014): 167-178.
- [MAY14] Maynard, G. HbbTV and Multi-Screen Strategy. HbbTV Symposium Asia. Singapore 18 June 2014.

Annex A: User Research

Generic Target Performance KPI's

For assessing the target performance of different components of the Multi-Device Media Representation and Interaction work package (Sections 3 to 4.7.2), 5 Target KPI's were defined as below. Three task-specific target KPI's are defined for every relevant section in this document and in D7.9.3. Next to that, 2 general usability KPI's were defined in order to measure the overall user experience.

Type	Type	Description	Target KPI
Task-specific	First Click score	After a limited amount of usage (familiarisation time, described below), did the user click the correct button for the optimal path to complete the requested task?	80% correctness for "top tasks" should be reached, 65% for other tasks
Task-specific	Task Completion	Was the user able to complete the task within a reasonable time? (A reasonable threshold depends on the task and on the product – it's best for someone who knows it well to set this)	80% correctness for "top tasks" should be reached, 65% for other tasks
Task-specific	Task Confidence	How confident is the user that s/he completed the task requested?	Average score of 5 or above (7-point scale)
Overall Usability	System Usability Scale (SUS) ⁹	This is a survey with 11 statements with an associated 7-point scale. The procedure for executing the SUS is described on the website in footnote 9.	Composite of 70 or higher (100-point scale)
Overall Usability	Usability Adjective Scale	This is a single question that is usually added to the end of the SUS that gives a scale of adjectives to describe whatever system is being measured from "worst imaginable" to "best imaginable"	4 or higher (7-point scale)

Figure 55: Generic Target Performance KPI's

Testing Procedure

A very short description of the testing procedure to follow is described below. The details can be found on the website in footnote 9:

1. Recruit at least 10 research participants with little to no knowledge of the system
2. Give each of these participants a brief (less than 5 minute) introduction to the feature under test. If possible, use the actual materials that a consumer would receive
3. Foresee for each participant a short amount of time to familiarise himself with the feature (usually less than 5 minutes for a consumer is good)
4. The moderator then asks each participant to complete a series of tasks. These tasks should be the most common actions users would take when working with the feature. (For SAM, this would include widget configuration, TV-side navigation, and 2nd Screen app usage.) Within each of those tasks there may be more than one sub-task for the participant. Of these sub-tasks, identify, by observing the behaviour of the participants, the sub-tasks that are absolutely crucial for using the feature

⁹ <http://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>

As a general guideline, the moderators should not guide or assist the participants in any way – they are there only to take notes and ensure the participants attempt each task and fill out the survey.

After each task, each participant answers a question about the ease of the task and how confident he is that he successfully completed the task. These statements are graded on a 5 or 7-point scale.

Task instruction	Ease of task / Confidence
Please collapse the video info widget	<div>Very difficult</div> <div>Very easy</div> <div> <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 <input type="radio"/> 7 </div>
How confident are you that you successfully completed the task?	<div>Not confident at all</div> <div>Totally confident</div> <div> <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 <input type="radio"/> 7 </div>

The moderator should note whether the participant made the correct first click and whether the participant successfully completed the task.

After all tasks are completed, the participants are then asked to complete the SUS¹⁰ including the Usability Adjective Scale. This is a survey with 11 statements with an associated 7-point scale.

¹⁰ <http://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>

