



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n: 761999



EasyTV: Easing the access of Europeans with disabilities to converging media and content.

D5.10 Integration and technical test plan final report

EasyTV Project

H2020. ICT-19-2017 Media and content convergence. – IA Innovation action.

Grant Agreement n°: 761999

Start date of project: 1 Oct. 2017

Duration: 33 months

Document. ref.: D5.10

Disclaimer

This document contains material, which is the copyright of certain EasyTV contractors, and may not be reproduced or copied without permission. All EasyTV consortium partners have agreed to the full publication of this document. The commercial use of any information contained in this document may require a license from the proprietor of that information. The reproduction of this document or of parts of it requires an agreement with the proprietor of that information. The document must be referenced if is used in a publication.

The EasyTV Consortium consists of the following partners:

	Partner Name	Short name	Country
1	Universidad Politécnica de Madrid	UPM	ES
2	Engineering Ingegneria Informatica S.P.A.	ENG	IT
3	Centre for Research and Technology Hellas/Information Technologies Institute	CERTH	GR
4	Mediavoice SRL	MV	IT
5	Universitat Autònoma Barcelona	UAB	ES
6	Corporació Catalana de Mitjans Audiovisuals SA	CCMA	ES
7	ARX.NET SA	ARX	GR
8	Fundación Confederación Nacional Sordos España para la supresión de barreras de comunicación	FCNSE	ES
9	Unione Italiana dei ciechi e degli ipovedenti	UICI	IT

PROGRAMME NAME:	H2020. ICT-19-2017 Media and content convergence - IA Innovation action
PROJECT NUMBER:	761999
PROJECT TITLE:	EASYTV
RESPONSIBLE UNIT:	ENG
INVOLVED UNITS:	ENG, CERTH, MV, ARX, CCMA
DOCUMENT NUMBER:	D5.10
DOCUMENT TITLE:	Integration and technical test plan final report
WORK-PACKAGE:	WP5
DELIVERABLE TYPE:	Report
CONTRACTUAL DATE OF DELIVERY:	31-03-2020
LAST UPDATE:	09-04-2020
DISTRIBUTION LEVEL:	PU

Distribution level:

PU = *Public*,

RE = *Restricted to a group of the specified Consortium*,

PP = *Restricted to other program participants (including Commission Services)*,

CO = *Confidential, only for members of the LASIE Consortium (including the Commission Services)*

Document History

VERSION	DATE	STATUS	AUTHORS, REVIEWER	DESCRIPTION
v. 0.1	02/03/2020	Draft	ENG	Table of Contents definition and document structure
v. 0.2	12/03/2020	Draft	ENG	Introduction and first integration tests description
v. 0.3	20/03/2020	Draft	ENG	Service Catalogue tests, Service Registry tests
v. 0.4	30/03/2020	Draft	ENG	SDK library testing description
v. 0.5	02/04/2020	Draft	ENG	Minor changes
v. 0.6	06/04/2020	Ready for internal review	CCMA	Review
v. 0.7	08/04/2020	Ready for internal review	UPM	Review
v. 1.0	09/04/2020	Final	ENG	Minor changes

Definitions, Acronyms and Abbreviations

ACRONYMS / ABBREVIATIONS	DESCRIPTION
API	Application Programming Interface
CD	Continuous Development
CI	Continuous Integration
CP	Crowdsourcing Platform
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
JSON	JavaScript Object Notation
REST	Representational State Transfer
SDK	Service Development Kit
SM	Service Manager
SOAP	Simple Object Access Protocol
SPM	Subtitle Production Module
V&V	Verification & Validation

Table of Contents

EXECUTIVE SUMMARY	8
1. INTRODUCTION	9
2. TEST DESCRIPTION.....	10
2.1. SPM-SM-CP INTEGRATION	10
2.1.1. CO creates a new job on the SM.....	10
2.1.2. SM submits a job to the SPM.....	11
2.1.3. SPM communicates job creation to the CP.....	12
2.1.4. SPM reports job completion to the SM	13
2.1.5. SM reports job completion data to the CO.....	14
2.2. SERVICE CATALOGUE INTEGRATION.....	15
2.2.1. Service Catalogue – Accessibility Services	16
2.2.2. Service Catalogue – Sign Language Ontology	16
2.2.3. Service Catalogue – Audio Subtitles.....	16
2.2.4. Service Catalogue – Crowdsourcing Platform	16
2.2.5. Service Catalogue – Multilingual Subtitles.....	17
2.3. SERVICE REGISTRY IMAGE PULLING FROM PORTAINER.....	17
2.3.1. Crowdsourcing Platform image pulling.....	17
2.3.2. Hyper-personalisation image pulling	17
2.3.3. Service Manager image pulling	19
2.3.4. Subtitles Production module image pulling.....	19
2.4. EASYTV SDK FOR EASYTV SERVICES TESTING	20
2.4.1. Task registration	20
2.4.2. Job start.....	21
2.4.3. Job result	22
3. TOOLS.....	24
3.1. POSTMAN	24
3.2. SELENIUM WEB DRIVER & SELENIUM IDE.....	24
4. CONCLUSIONS	25
REFERENCES	26

List of Tables

Table 1 – CO-SM-001	10
Table 2 - CO-SM-001 request/response	11
Table 3 - SM-SPM-001.....	11
Table 4 - SM-SPM-001 request/response.....	12
Table 5 - SPM-CP-001	13
Table 6 - SPM-CP-001 request/response.....	13
Table 7 - SM-SPM-002.....	14
Table 8 - SM-SPM-002 request/response.....	14
Table 9 - CO-SM-001	14
Table 10 - CO-SM-001 request/response	15
Table 11 - SC-AS-001	16
Table 12 - SC-SLO-001.....	16
Table 13 - SC-AS-001	16
Table 14 - SC-CP-001	17
Table 15 - SC-MS-001.....	17
Table 16 - SR-CP-001	17
Table 17 - SR-HP-001	18
Table 18 - SR-HP-002	18
Table 19 - SR-HP-003	18
Table 20 - SR-HP-004	18
Table 21 - SR-SM-001.....	19
Table 22 - SR-SM-002.....	19
Table 23 - SR-SM-003.....	19
Table 24 - SR-SPM-001	20
Table 25 - SR-SPM-002	20
Table 26 - SR-SPM-003	20
Table 27 - SDK-001.....	21
Table 28 - SDK-001 request/response	21
Table 29 - SDK-002.....	21
Table 30 - SDK-002 request/response	22
Table 31 - SDK-003.....	22
Table 32 - SDK-003 request/response	23

EXECUTIVE SUMMARY

This document is the deliverable D5.10 in WP5. It is the final report on the integration and technical testing plan, which details integration testing choices and execution results obtained that ensure the correct working of the modules involved in the EasyTV platform. In addition to the service modules, tests involving the EasyTV Service Catalogue and the EasyTV Service Registry, along with the EasyTV SDK, are included.

The first chapter is a brief introduction related to the concepts and meaning of “testing” and “integration testing”.

The second chapter (“Test description”) outlines the tests executed. When applicable, details related to the request/response module flow are given.

The third chapter describes the tools used to perform the tests described.

Finally, conclusions are presented in the last chapter.

1. INTRODUCTION

Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results [1]. It is an integral part in software development and it is deployed in every phase in the software development cycle.

Testing is performed for the following purposes:

- to identify defects (bugs or technical errors) and to subsequently correct those identified defects;
- for verification & validation (V&V). According to the IEEE Standard Glossary of Software Engineering Terminology, verification is defined as "The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase" [2]. Validation, on the other hand, is defined as "The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements" [2]. Testing is therefore used as a tool in the V&V process;
- for reliability estimation, where reliability is defined as "the probability of failure-free operation for a specified period of time in a specified environment" [2].

There are many ways to categorize testing methods and testing techniques. By scope, software testing can be categorized as follows: unit testing, component testing, integration testing and system testing. The present document deals with **integration testing**, whose goal is to determine if the independently developed EasyTV components work correctly when they are connected and interacting with each other.

Integration testing assumes that the EasyTV components work as expected if taken **individually** (thus they have already been unit tested) and then groups them on the basis of the system architecture. The integration tests identified and presented here cover the identified information flows and the interaction sequences among the service modules, according to the developments carried out.

2. TEST DESCRIPTION

This chapter describes tests which have been identified and performed in order to verify the proper functioning and the correct interaction flow among the existing EasyTV components. It should be noted that in relation to the tests that were hypothesized and presented in D5.5 (“Integration and technical testing plan mid-term report”) the tests herein described differs in terms of number and type: this is due to the fact that changes to the final module interaction schemes during the development phase have been taken into account, thereby re-shaping the tests proposed in D5.5 in a different way.

2.1. SPM-SM-CP integration

The Subtitle Production Module (SPM) is the module running on the EasyTV platform which performs subtitle translation in multiple languages by interacting with both the Service Manager (SM) and the Crowdsourcing Platform (CP). The former is a module which handles jobs and tasks (on behalf of the Content Owner) involving one or more EasyTV services, while the latter is a platform which allows users to participate in translation tasks and contribute with their knowledge-based skills.

Large efforts were made in software analysis and design phases in order to mitigate risks concerning integration errors between modules: in this regard, the use of Swagger to design and document the different APIs has proven to be very useful.

Frequent deployments have been adopted in a CI/CD fashion to always have a clear outcome of the end results and of the possible side effects and troubleshooting procedures were carried out accordingly to promptly resolve the integration issues that have arisen. Moreover, about 1600 subtitle translation jobs were created for testing purposes, further aiding software error identification and fixing.

The integration tests performed which involve the Subtitle Production Module, the Service Manager and the Crowdsourcing Platform are reported below. A trimmed version of the original request/response data in JSON format is also shown when applicable.

2.1.1. CO creates a new job on the SM

Test ID	CO-SM-001
Description	The Content Owner should register a new job on the Service Manager.
Prerequisites	The original video file and the original subtitle information in EBU-TT-D format are available.
Expected result	The Service Manager successfully creates a new job.

Table 1 – CO-SM-001

Request (HTTP method: POST, target URL: https://sm-api.easytv.eng.it/api/job)
{ "publication_date":156000000,

<pre> "expiration_date":156000500, "tasks":[{ "task_id":19, "input":{ "language_source":"ca", "language_target":"es", "confidence_level":"high", "video_path":"475957_2715392_0157900.mp4", "asset_name":"Crims - La bibliotecària Helena Jubany (capítol 1)", "asset_duration":"00:49:01", "original_title":"capitol 9", "original_text":"... <subtitle EBUTTD escaped>... ", "original_synopsis":"El primer capítol del cas comença la matinada de l'1 al 2 de desembre del 2001, a Sabadell, amb una noia de 27 anys morta. És Helena Jubany i el seu cos apareix en un pati interior, nu i ple de cremades. Ben aviat els investigadors descobreixen que l'han drogat amb somnífers i que algú l'ha tirat des del terrat d'un edifici de pisos. Una de les seves millors amigues explica a la policia que la víctima va rebre dues cartes anònimes abans de morir. En una hi ha una referència a la Unió Excursionista de Sabadell (UES), de la que Helena era sòcia. Els escrits fan que la policia se centri en els amics de la UES i sospita sobretot d'una de les seves amigues, Muntsa Careta, i de la seva parella, Santi Laiglesia. Però ningú troba el mòbil. Per què van matar Helena Jubany?" } }] } </pre>
Response
<pre> { "code":200, "description":"Job created", "job_id":1204 } </pre>

Table 2 - CO-SM-001 request/response

2.1.2. SM submits a job to the SPM

Test ID	SM-SPM-001
Description	The Service Manager should submit a job to the Subtitle Production Module.
Prerequisites	An existing job is available on the Service Manager.
Expected result	The Subtitle Production Module accepts the job.

Table 3 - SM-SPM-001

Request (HTTP method: POST, target URL: https://spm-api.easytv.eng.it/api/Jobs)
<pre>{ "job_id":1025, "publication_date":156000000, "expiration_date":156000500, "content_owner":"ccma", "input":{ "language_source":"ca", "language_target":"es", "confidence_level":"high", "video_path":"475957_2715392_0157900.mp4", "asset_name":"Crims - La bibliotecària Helena Jubany (capítol 1)", "asset_duration":"00:49:01", "original_title":"capítol 9", "original_text":"... <subtitle EBUTTD escaped>... ", "original_synopsis":"El primer capítol del cas comença la matinada de l'1 al 2 de desembre del 2001, a Sabadell, amb una noia de 27 anys morta. És Helena Jubany i el seu cos apareix en un pati interior, nu i ple de cremades. Ben aviat els investigadors descobreixen que l'han drogat amb somnífers i que algú l'ha tirat des del terrat d'un edifici de pisos. Una de les seves millors amigues explica a la policia que la víctima va rebre dues cartes anònimes abans de morir. En una hi ha una referència a la Unió Excursionista de Sabadell (UES), de la que Helena era sòcia. Els escrits fan que la policia se centri en els amics de la UES i sospita sobretot d'una de les seves amigues, Muntsa Careta, i de la seva parella, Santi Laiglesia. Però ningú troba el mòbil. Per què van matar Helena Jubany?" } }</pre>
Response
<pre>{ "code":202, "description":"Job accepted" }</pre>

Table 4 - SM-SPM-001 request/response

2.1.3. SPM communicates job creation to the CP

Test ID	SPM-CP-001
Description	The Subtitles Production Module should inform the Crowdsourcing Platform that a translation job is ready.
Prerequisites	A translation job is ready on the Subtitles Production Module.

Expected result	The Crowdsourcing Platform acknowledges the job existence.
-----------------	--

Table 5 - SPM-CP-001

Request (HTTP method: POST, target URL: https://cp.easytv.eng.it/available/subt)
<pre>{ "job_id":1025, "action":"edition", "testmode":false, "status":"AwaitingForEdition", "data":{ "content_owner":"ccma", "publication_date":156000000, "expiration_date":156000500, "language_source":"ca", "language_target":"es", "confidence_level":"high", "asset_name":"Crims - La bibliotecària Helena Jubany (capítol 1)", "asset_duration":"00:49:01", "link":"https://spm.easytv.eng.it/#editor", "validated_percent":0 } }</pre>
Response
<pre>{ "code":200, "description":"The job was registered successfully" }</pre>

Table 6 - SPM-CP-001 request/response

2.1.4. SPM reports job completion to the SM

Test ID	SM-SPM-002
Description	The Subtitles Production Module should report to the Service Manager that a translation job has been completed.
Prerequisites	The translation job has been completed.

Expected result	The Service Manager acknowledges the job has been completed.
-----------------	--

Table 7 - SM-SPM-002

Request (HTTP method: POST, target URL: <a href="https://sm-api.easytv.eng.it/internal/job/<ID>/finish">https://sm-api.easytv.eng.it/internal/job/<ID>/finish)
<pre>{ "output":{ "translated_text":"... <subtitle EBUTTD escaped>...", "translated_title":"capítulo 9", "translated_synopsis":"El primer capítulo del caso comienza la madrugada del 1 al 2 de diciembre de 2001, en Sabadell, con una chica de 27 años muerta. Es Helena Jubany y su cuerpo aparece en un patio interior, desnudo y lleno de quemaduras. Pronto los investigadores descubren que la han drogado con somníferos y que alguien la ha tirado desde la azotea de un edificio de pisos. Una de sus mejores amigas explica a la policía que la víctima recibió dos cartas anónimas antes de morir. En una hay una referencia a la Unió Excursionista de Sabadell (UES), de la que Helena era socia. Los escritos hacen que la policía se centre en los amigos de la UES y sospecha sobre todo de una de sus amigas, Muntsa Careta, y de su pareja, Santi Laiglesia. Pero nadie encuentra el móvil. ¿Por qué mataron a Helena Jubany?" } }</pre>
Response
<pre>{ "status":200, "description":"The job <ID> was finished" }</pre>

Table 8 - SM-SPM-002 request/response

2.1.5. SM reports job completion data to the CO

Test ID	CO-SM-001
Description	The Content Owner should get status and results of the completed translation job.
Prerequisites	The translation job has been successfully completed.
Expected result	The Service Manager returns translation data to the Content Owner.

Table 9 - CO-SM-001

Request (HTTP method: GET, target URL: <a href="https://sm-api.easytv.eng.it/api/job/<jobID>">https://sm-api.easytv.eng.it/api/job/<jobID>)
N.A.

Response
<pre>{ "code":200, "description":"Success", "job":{ "completion_date":1584993759, "creation_date":1584986620, "current_task":null, "expiration_date":1588883700, "id":1025, "is_canceled":false, "is_completed":true, "output":{ "translated_text":"... <subtitle EBUTTD escaped>...", "translated_title":"capítulo 9", "translated_synopsis":"El primer capítulo del caso comienza la madrugada del 1 al 2 de diciembre de 2001, en Sabadell, con una chica de 27 años muerta. Es Helena Jubany y su cuerpo aparece en un patio interior, desnudo y lleno de quemaduras. Pronto los investigadores descubren que la han drogado con somníferos y que alguien la ha tirado desde la azotea de un edificio de pisos. Una de sus mejores amigas explica a la policía que la víctima recibió dos cartas anónimas antes de morir. En una hay una referencia a la Unió Excursionista de Sabadell (UES), de la que Helena era socia. Los escritos hacen que la policía se centre en los amigos de la UES y sospecha sobre todo de una de sus amigas, Muntsa Careta, y de su pareja, Santi Laiglesia. Pero nadie encuentra el móvil. ¿Por qué mataron a Helena Jubany?" } }, "publication_date":1584999300, "status":"Completed", "tasks":[{ "task_id":19, "task_name":"EasyTV_AutomaticSubtileProductionModule_Docker" }] }</pre>

Table 10 - CO-SM-001 request/response

2.2. Service Catalogue integration

This section describes the integration tests performed in order to verify the correct integration of the existing services with the EasyTV Service Catalogue. The execution of the following tests has been automated by using Selenium IDE in conjunction with Selenium WebDriver.

2.2.1. Service Catalogue – Accessibility Services

Test ID	SC-AS-001
Description	The Accessibility Services API (face magnification, text recognition, character recognition and background sound detection) should be reachable through the Service Catalogue.
Prerequisites	The Accessibility Services components are registered in the Service Catalogue.
Expected result	The Accessibility Services page is loaded correctly.

Table 11 - SC-AS-001**2.2.2. Service Catalogue – Sign Language Ontology**

Test ID	SC-SLO-001
Description	The Sign Language Ontology web service should be reachable through the Service Catalogue.
Prerequisites	The Sign Language Ontology web service is registered in the Service Catalogue.
Expected result	The Sign Language Ontology page is loaded correctly.

Table 12 - SC-SLO-001**2.2.3. Service Catalogue – Audio Subtitles**

Test ID	SC-AS-001
Description	The Audio Subtitles module should be reachable through the Service Catalogue.
Prerequisites	The Audio Subtitles module is registered in the Service Catalogue.
Expected result	The Audio Subtitle module page is loaded correctly.

Table 13 - SC-AS-001**2.2.4. Service Catalogue – Crowdsourcing Platform**

Test ID	SC-CP-001
---------	-----------

Description	The Crowdsourcing Platform should be reachable through the Service Catalogue.
Prerequisites	The Crowdsourcing Platform is registered in the Service Catalogue.
Expected result	The Crowdsourcing Platform page is loaded correctly.

Table 14 - SC-CP-001

2.2.5. Service Catalogue – Multilingual Subtitles

Test ID	SC-MS-001
Description	The Multilingual Subtitles (Subtitles Production) module should be reachable through the Crowdsourcing Platform, which in turn provides access to the Multilingual Subtitles module.
Prerequisites	The Multilingual Subtitles module is registered in the Service Catalogue.
Expected result	The Crowdsourcing Platform page is loaded correctly.

Table 15 - SC-MS-001

2.3. Service Registry image pulling from Portainer

This section describes the tests performed in order to verify the correct Docker image pulling flow from the EasyTV Service Registry onto the running Docker platform.

2.3.1. Crowdsourcing Platform image pulling

Test ID	SR-CP-001
Description	The Crowdsourcing Platform “crowdsourcing-app” image should be pulled from the Service Registry and it should be available in Portainer.
Prerequisites	The image is available in the Service Registry.
Expected result	The image is correctly pulled from the Service Registry.

Table 16 - SR-CP-001

2.3.2. Hyper-personalisation image pulling

Test ID	SR-HP-001
---------	-----------

Description	The Hyper-personalisation module “stmm” image should be pulled from the Service Registry and it should be available in Portainer.
Prerequisites	The image is available in the Service Registry.
Expected result	The image is correctly pulled from the Service Registry.

Table 17 - SR-HP-001

Test ID	SR-HP-002
Description	The Hyper-personalisation module “rbmm” image should be pulled from the Service Registry and it should be available in Portainer.
Prerequisites	The image is available in the Service Registry.
Expected result	The image is correctly pulled from the Service Registry.

Table 18 - SR-HP-002

Test ID	SR-HP-003
Description	The Hyper-personalisation module “hbmm” image should be pulled from the Service Registry and it should be available in Portainer.
Prerequisites	The image is available in the Service Registry.
Expected result	The image is correctly pulled from the Service Registry.

Table 19 - SR-HP-003

Test ID	SR-HP-004
Description	The Hyper-personalisation module “easytv_hp_api” image should be pulled from the Service Registry and it should be available in Portainer.
Prerequisites	The image is available in the Service Registry.
Expected result	The image is correctly pulled from the Service Registry.

Table 20 - SR-HP-004

2.3.3. Service Manager image pulling

Test ID	SR-SM-001
Description	The Service Manager module “easytv_sm_cron” image should be pulled from the Service Registry and it should be available in Portainer.
Prerequisites	The image is available in the Service Registry.
Expected result	The image is correctly pulled from the Service Registry.

Table 21 - SR-SM-001

Test ID	SR-SM-002
Description	The Service Manager module “easytv-sm” image should be pulled from the Service Registry and it should be available in Portainer.
Prerequisites	The image is available in the Service Registry.
Expected result	The image is correctly pulled from the Service Registry.

Table 22 - SR-SM-002

Test ID	SR-SM-003
Description	The Service Manager module “easytv-sm-ui” image should be pulled from the Service Registry and it should be available in Portainer.
Prerequisites	The image is available in the Service Registry.
Expected result	The image is correctly pulled from the Service Registry.

Table 23 - SR-SM-003

2.3.4. Subtitles Production module image pulling

Test ID	SR-SPM-001
Description	The Subtitles Production module “easytvspmapi” image should be pulled from the Service Registry and it should be available in Portainer.
Prerequisites	The image is available in the Service Registry.

Expected result	The image is correctly pulled from the Service Registry.
-----------------	--

Table 24 - SR-SPM-001

Test ID	SR-SPM-002
Description	The Subtitles Production module “easytvspmengine” image should be pulled from the Service Registry and it should be available in Portainer.
Prerequisites	The image is available in the Service Registry.
Expected result	The image is correctly pulled from the Service Registry.

Table 25 - SR-SPM-002

Test ID	SR-SPM-003
Description	The Subtitles Production module “spm-frontend” image should be pulled from the Service Registry and it should be available in Portainer.
Prerequisites	The image is available in the Service Registry.
Expected result	The image is correctly pulled from the Service Registry.

Table 26 - SR-SPM-003

2.4. EasyTV SDK for EasyTV services testing

This section describes the steps executed to test the SDK usage in a test service in order to set up a communication between the EasyTV Service Manager and the test service itself. The test service perform a simple text transformation (lowercase to uppercase text transformation) and a concatenation of the transformed string with an integer number.

2.4.1. Task registration

Test ID	SDK-001
Description	The test service should register a new task in the Service Manager (SDK <i>registerTask</i> method from <i>ServiceClient</i> class).
Prerequisites	A service named “ENG test service” has been registered in the Service Manager and an API

	key has been obtained.
Expected result	A new task has been successfully registered.

Table 27 - SDK-001

Request
<pre>{ "name": "ENG test service test task", "description": "Test task for SDK integration test", "start_url": "https://testservice.easytv.eng.it/start_job", "cancel_url": "https://testservice.easytv.eng.it/cancel_job", "input": { "input_string": "string", "input_number": "int" }, "output": { "result_text": "string" } }</pre>
Response
<pre>{ "code": 200, "description": "The task "ENG test service test task" was successful registered", "task_id": 20 }</pre>

Table 28 - SDK-001 request/response

2.4.2. Job start

Test ID	SDK-002
Description	The test service should start a new job (single task) upon Service Manager call to the configured "start_url" URL.
Prerequisites	A task has been registered in the Service Manager.
Expected result	A new job has been successfully started.

Table 29 - SDK-002

Request
<pre>{ "job_id": 3010, "publication_date": 1583829400, "expiration_date": 1584693400, "content_owner": "Test Owner", "input": { "input_string": "easytv ", "input_number": 2020 } }</pre>
Response
<pre>{ "code": 202, "description": "Job accepted" }</pre>

Table 30 - SDK-002 request/response

2.4.3. Job result

Test ID	SDK-003
Description	The test service should notify the Service Manager the job has finished and should provide the final job results (SDK <i>finishJob</i> method from <i>ServiceClient</i> class).
Prerequisites	A job has been processed.
Expected result	Service Manager acknowledges the job completion.

Table 31 - SDK-003

Request
<pre>{ "output": { "result_text": "EASYTV 2020" } }</pre>
Response

```
{  
  "status": 200,  
  "description": "The job 3010 was finished"  
}
```

Table 32 - SDK-003 request/response

3. TOOLS

This chapter gives an overview of the tools used in the execution of the test presented in the previous chapter.

3.1. Postman

Postman [3] is a collaboration platform for API development. It allows to perform multiple REST, SOAP or GraphQL requests with an intuitive GUI interface: complex requests can be defined and optionally saved in collections for repeated use. Testing automation is simplified through the creation of test suites, which can be used to perform unit testing, integration testing and end-to-end testing. Test workflow can be managed directly from the Postman application or through the command line tool Newman [4].

3.2. Selenium Web Driver & Selenium IDE

Selenium Web Driver [5] is a component that communicates with a browser through a driver which is specific to the browser (e.g. ChromeDriver for Google Chrome browser or GeckoDriver for Firefox browser). Communication takes place in a two-way fashion: WebDriver passes commands to the browser through the driver and receives information back via the same route. Automation test scripts can be written in a wide range of languages (most notably Java and Javascript).

Selenium IDE [6], on the other hand, is a tool available as an extension for both Chrome and Firefox browsers which is used to perform tests on websites based on user interactions. It supports recording and playback of automation tests through an easy-to-use IDE, which also allows to change tests programmatically once they have been recorded. Multiple conditions and expressions can be used to assert and validate each test, thus providing a high level of flexibility in both number and quality of tests performed. Scripts can be exported as Selenium WebDriver test scripts or directly executed on selenium WebDriver servers.

4. CONCLUSIONS

Integration testing plays a key role in the software development lifecycle, as it ensures that the integrated modules are working as expected when combined together. The EasyTV platform, in particular, requires a certain number of integration tests to be performed since it is based on several interrelated modules and components.

This document presented the tests that have been designed and executed in order to ensure the correct integration among the EasyTV service modules. First of all, the integration flow between the Subtitle Production Module, the EasyTV Service Manager and the Crowdsourcing Platform has been presented. Service integration with the EasyTV Service Catalogue has been described, followed by a description of the tests performed to verify the correct image pulling and execution onto the existing Docker platform. Finally, a test service has been developed in order to demonstrate the correct EasyTV SDK usage in the integration with the EasyTV Service Manager.

The ultimate goal was the achievement of a stable platform and reliable communication among the different components of the EasyTV ecosystem. Overall, the testing procedures presented brought satisfactory results: testing performed during the development phase helped to streamline the whole process and to identify software errors in a timely manner. On the other hand, testing performed after each software release contributed to the consolidation of the codebase, allowing to confidently perform the necessary code refactoring and the introduction of new features.

REFERENCES

- [1] William C. Hetzel, *The Complete Guide to Software Testing*, 2nd ed.
- [2] IEEE, *IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990)*.
- [3] Postman. Postman. [Online]. <https://www.postman.com/>
- [4] Postman. Newman. [Online]. <https://github.com/postmanlabs/newman>
- [5] Selenium. Selenium WebDriver. [Online]. <https://www.selenium.dev/documentation/en/webdriver/>
- [6] Selenium. Selenium IDE. [Online]. <https://www.selenium.dev/selenium-ide/>