Funded by the European Union's Internal
Security Fund - Police (2014-2020)

Internal Security Fund – Police - Programme (2014-2020)
823701 - YoungRes - ISFP-2017-AG-RAD

Strengthening European Youngsters Resilience through Serious Games - YoungRes

Deliverable D4.3
MANUAL FOR UTILIZATION OF ONLINE PORTAL

Document information	
Due date of deliverable	31 OCTOBER 2019
Actual submission date	29 February 2020
Organisation name of lead partner for this deliverable	OULU
Final Version Number	

Dissemination Level		
PU	Public	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Authors list

Author	Partner
Mourad Oussalah	OULU

Peer Reviewers

Reviewer	Partner
Mourad Oussalah	OULU
Javier Torregrosa	UPM
Raquel Menéndez	UPM

Versioning

Version	Date	Summary
DRAFT	05-09-2019	Online platform design
V1	05-10-2019	Software architecture design
V1	15-11-2019	Management sytem
V1	11-11-2019	Video game application
V1	20-01-2020	Visualization tool
V2	28-02-2020	Final version

The content of this report represents the views of the author only and is his/her sole responsibility. The European Commission does not accept any responsibility for use that may be made of the information it contains.

Table of Contents

1 Introduction

This deliverable provides a manual that describes all the technological applications available in the online platform of YoungRes project. The main objective of the platform is to present the proposed methodology to a wider (online) audience. It will enable educators to report their feedbacks about any outcomes of the educational system. At the same time, it enables easy manipulation of data by the educators who can collect, store and represent anonymised data from the users.

In the following sections, the main applications available in the online platform, that allow users to access to all the resources developed in the YoungRes project, will be described. This manual will contain the description of how to access and use the following applications: the software architecture, the visualization tool, the management system (for administrators) and the videogame platform.

2 Software architecture

The architecture of YoungRes has to satisfy several requirements related to the content offered to the different users who will interact with the online platform (children, administrators and educators/teachers).

As can be seen in Figure 1, there are three different types of interactions depending on the user. Access is managed by three different Application Programming Interface (APIs) that communicate the user with the back-end system and the database. The three different types of users that will interact with the system are the following:

- **Youngsters** are the target users. They will interact with the system by playing the video game designed to improve the resilience.
- **Teachers** interact with the system using the visualization tool that will extract valuable information regarding the performance of the youngsters in the video game.
- **Administrators** are in charge of creating the different users, uploading the video game, and enabling the different licenses in the webpage in order to let educators access the content.

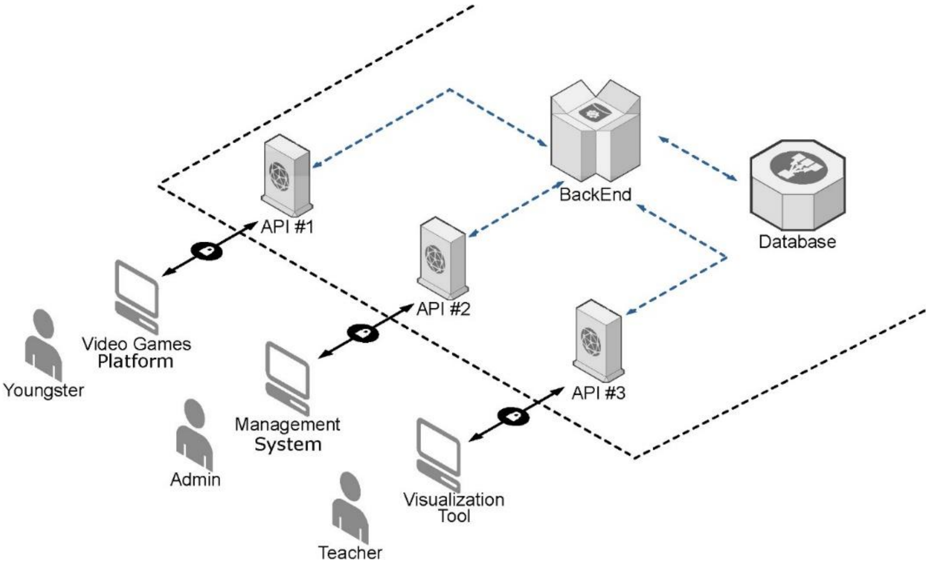


Figure 1. Software architecture of the system designed in YoungRes project. This figure shows the different components of the system, the different users that will work with the system and the different subsystems that these users will interact with.

The software architecture is divided in four modules: videogame, management system, back-end, and visualization tool. All the applications are independent from one another and use the back-end as communication device.

2.1 Video game application

The video game is developed using RPG Maker. Although other programs can be used to create the video games (such as Unity 3D) RPG Maker contains several advantages that can be very useful at this point of the project:

- RPG Maker contains several modules that allow a non-technical user to develop a full video game. It gives the designer the opportunity to easily create an RPG game by using events as basic precoded instances. This characteristic makes that all our effort, at the beginning of the project, is focused on the topics that we want to work with the youngsters and the storytelling of the video games.

- As RPG Maker is focused on creating RPG type games, storytelling can be the focus of the development, without being worried about implementing mechanics on the game. Also, RPG games are really useful due to the possibility of interacting with the characters and define their future action through decisions raised during conversations. Therefore, conversations can be used as a tool to transmit the desired knowledge, but also to extract those factors that allow the teachers to assess the resilience level of the youngsters.

The video game is also connected to the database, which is required to store anonymized information and retrieve it to extract some valuable knowledge. The information retrieved from the avatar of the youngsters and the gameplay will be stored via a REST API. This information will be used to analyse the knowledge of the youngster regarding the topic of the video game. The game will implement a login system that allows identifying a student and selecting a chapter.

2.1.1. Credentials to access to the video game

Before using the video game, the admin team will populate the database with the group info and create as many `studentCodes` as necessary for the whole groups. The available `studentCodes` will be handled by the teachers so each student will be linked to a unique `studentCode`. Only the teacher will have access to the list that link real students to `studentCodes`. During each intervention (video game use) along with the information handled by the teacher, several `chapterCodes` will be handled. Each `chapterCode` will be used in one session of the intervention (Figure 2 shows a description of all the system). The access to each session is structured as follows:

- The teacher communicates the students the respective `chapterCode` for that session and their `studentsCodes`.
- The video game starts with a login page where these two values can be inserted.
- Once the codes are introduced, the video game uses the GAMEPLAY API `chapterStartScene` endpoint to retrieve the scene of the game where the chapter starts.
- The video game loads the scene where the chapter starts and runs normally, using the GAMEPLAY API `storeVariable` and `loadVariable` endpoints to read and write customization data to be used on the game.
- Once an event has occurred the video game uses the GAMEPLAY API `storeDecision` endpoint to store what the students choose.

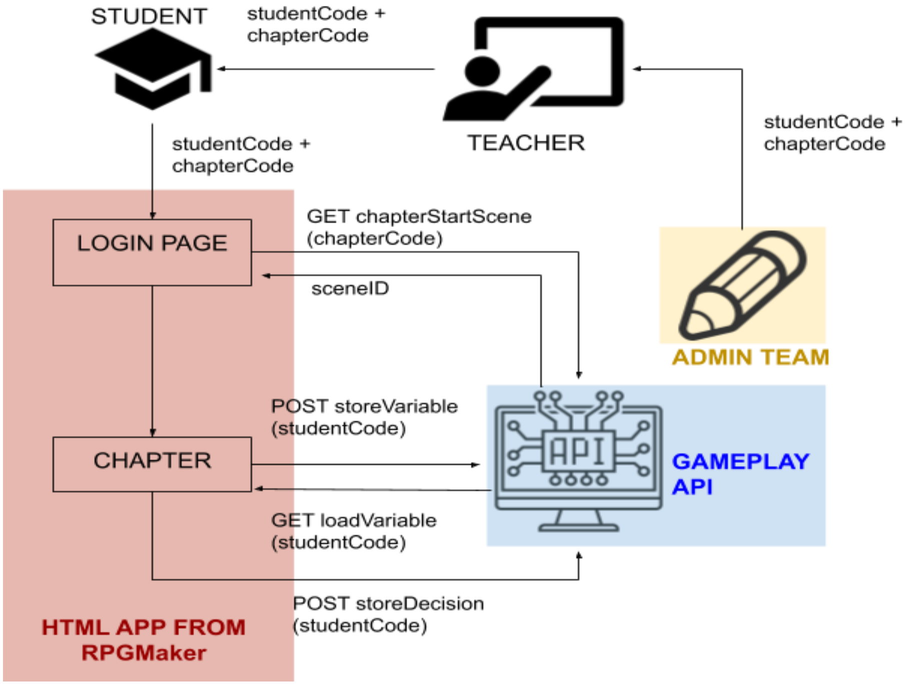

STUDENT
studentCode + chapterCode
TEACHER
studentCode + chapterCode
studentCode + chapterCode
LOGIN PAGE
GET chapterStartScene (chapterCode)
sceneID
ADMIN TEAM
POST storeVariable (studentCode)
CHAPTER
API
GAMEPLAY API
GET loadVariable (studentCode)
POST storeDecision (studentCode)
HTML APP FROM RPGMaker

Figure 2. Description of the architecture of the video game access and stored.

2.2 Management system

This module is used by the administrators for several purposes:

- Register in the system both youngsters and teachers.
- Upload the video game to the Video Games Platform.
- Enable the different sessions of the video game.

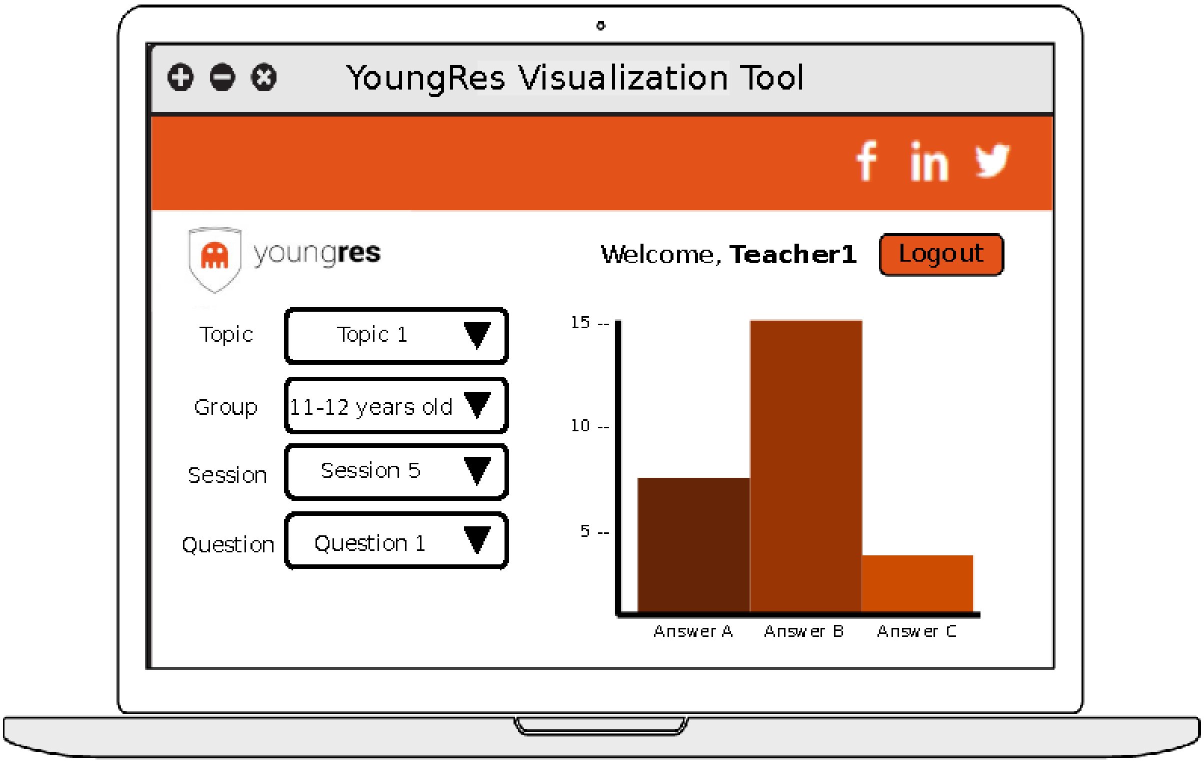
The management system will be a web application. The first task of the administrators is the creation of accounts for youngsters and teachers. This task is required as the platform will not be a public system on the Internet. If any institution wants to participate, they must contact the members of the project using the procedure described on the webpage.

Administrators must enable the different sessions of the video game due to each intervention session with the videogame lasts several days, and each day a different part of the video games is played. This restriction is necessary because once the youngsters plays the video game, teachers have to explain the related theoretical concepts. For this reason, it is extremely important for teachers that all youngsters in the class work on the same part of the video game.

2.3 Visualization tool

The visualization tool is a critical module of the system, as it is in charge of generating reports regarding the behaviour of the youngsters during the gameplay. This system will be only used by the teachers/educators in order to understand the global performance of the group. By using this tool, teachers will check which concepts are correctly understood or should be deeply reviewed with the class. To access the visualization tools, teachers will be provided with a password.

Visualization tool will provide global and anonymized information about the decisions made by the youngsters. For example, given a question asked on the video game (or a decision made), the teacher could easily visualize how many youngsters in the group have chosen each of the options. Figure 3 shows an example of the Graphical User Interface (GUI) of this online tool.



Topic: Topic 1

Group: 11-12 years old

Session: Session 5

Question: Question 1

Figure 3. Graphical User Interface of the Visualization tool of the YoungRes project. In this example, any teacher could easily visualize how many youngsters have selected the different options in a specific question of the video game

Figure 3 is a mockup that shows the information that will be shown in the system. The main idea is to develop a tool able to extract valuable information about the evolution of the group during the different sessions. The outcomes of this tool must be generic, as it must provide information regarding the group, instead of individual information about a specific youngster. This is because all the information stored in the database is anonymized so it is not possible to identify the real student.

2.4 Back-end Application

This application acts as a back-end to store all the data gathered during the different gameplays. Its functionality is twofold: on the one hand, it needs to give support to the game application allowing to store user decisions taken inside the game and reading/writing relevant information needed for the gameplay. On the other hand, it needs to give access to the stored data for later analysis and visualization. This application is administered via the configuration application. Only administrators can access the functions of the back-end application.

Application that acts as a back-end to store all the data gathered during the different gameplays. Its composed of two databases and three REST API. It will be developed it as a docker-compose application for easy migration/workflow.

- Databases: MongoDB will be used as database engine.
    - Game data: In this database all the data related with the game will be stored.
    - Credentials data: In this database the credentials for accessing the REST API are stored.

REST API: OpenAPI specification is being used to describe the REST APIs (Figure 1) and Python Flask as server technology. All endpoints have basic authentication.

Gameplay API: the REST API #1 (Figure 1) contains all the endpoints related to the inner workings of the game. It allows storing game related variables as well as users in-game decisions. Also, it allows retrieving game related variables previously stored.

Data Visualization/Consumption API: The REST API #3 (Figure 1) contains endpoints to access all the data gathered during the interventions and access the configuration information about the game.

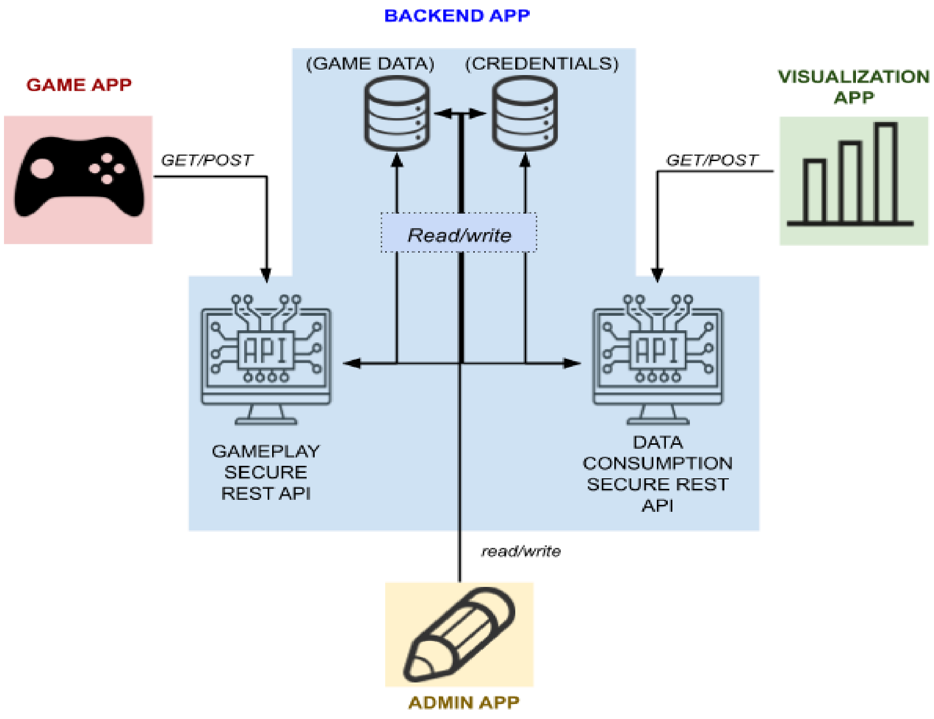


Figure 4. Structure of the back-end application

3 Concluding Remarks

A complete description of the applications included in the online platform is presented in this manual. It offers essential information for the three main users of the project (youngsters, teachers and administrators) about how to access to these applications and what information there will find in each application.

Online platform will contain the three main applications aimed in the YoungRes project: the video game developed in the project, the management system of all the applications, visualization tool and Back-end Application. To access all of these applications a password will be necessary in order to control and management the users and the data gathering. The management system will control all the users and the upload of all the applications.