aal-2014-171

SENIOR-TV

PROVIDING ICT-BASED FORMAL AND INFORMAL CARE AT HOME

Deliverable D2.1
SENIOR-TV

Document information	
Due date of deliverable	15/10/2016
Actual submission date	09/12/2016
Organisation name of lead contractor for this deliverable	IMATIA and CPX
Revision	V1

Dissemination Level		
PU	Public	X
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**
Iulian Anghelache	CPX
Giorgos Kostopoulos	GULK
Sira López	IMATIA
Carlos Rivas	IMATIA/UVigo
Dejan Tomažin	RC-IKTS

Peer Reviewers	
Reviewer	**Partner**
Luis Anido Rifón	UVIGO (subcontracted by IMATIA)
Vojko Strojnik	RC-IKTS

Versioning	
Version	**Summary**
V_1.0	Final version.
V_2.0	Final version including internal reviewers’ comments.

This project has been funded with support from the European Commission.

This document reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Table of Contents

1. Introduction

This document summarizes the work performed from month M1 to M12 in WP2, which final result is the first version of SENIOR-TV platform.

The main objective of WP2 is produce a platform based on interactive TV and mobile technologies according to the requirements of openness and low-cost. The obtained platform has to cover the general project aims: providing formal and informal caregiving services for older adults, focusing on the active prevention and the maintenance of relationships with their friends, their relatives and, the community.

These are the main tasks developed in the first cycle and the objectives have been achieved in each of them:

- T2.1.1. Technology Analysis (M1-M6) and T2.1.2. SMART-TV technology development (M7-M12): the analysis of existing smart-TV software and hardware technologies.
- T2.2.1: First wave (M3-M12): the initial version of most relevant informal services (according the result of WP1 workshops).
- T2.3.1: Formal services first wave (M3-M12): the analysis of potential formal services for the platform.
- T2.4.1: Ergonomic Design. First wave (M4-M12): the analysis of senior necessities regarding graphical interfaces and the design of implemented services according these restrictions.
- T2.5.1. PD&SI first wave (M4-M12): internationalization and external providers integration.

The following sections include a detailed description of performed work in each of these tasks. Sections 2 and 3 include the details of technology analysis. Sections 4, 5 and 6 presents the developed services status and describe the graphical interface advances. Finally, the section 7 includes a review of the required steps to advance on the platform development.

2. Technology analysis

2.1. Smart-TV software

In this section, we introduce the main environments for application development in the smart TV world. While early approaches were based on proprietary systems, present-day, more elaborated solutions are based on Linux, and in some cases, follow the-web-as-a-platform paradigm based on technologies such as HTML5, CSS or JavaScript. The most relevant proposals available at the time of this writing are introduced below, namely AndroidTV, WebOS, FireOS, Firefox OS, Tizen. A final summary is also included to compare the characteristics of the approaches discussed.

- **Android**[1] is a Linux-based operating system developed by the Open Handset Alliance, a consortium of developers led by Google. In June 2014, Google introduced AndroidTV[2] as an evolution of Android version 5.0 Lollipop to be deployed in television environments. Currently, manufacturers like Sony, Sharp and Philips have shown interest in including this operating system in their appliances, being also possible to acquire distributions in USB sticks to be connected directly to a media input (e.g., HDMI).

  Application programming for smart televisions in Android follows the same model as smartphone programming. It is based on the Java programming language and supports user interfaces definition through XML files. Frameworks are also available to support the development of web applications based on HTML5, although these applications do not have unlimited access to operating system services as native applications do. AndroidTV developers must explicitly indicate that their applications are designed for a TV environment. Typical features of a mobile phone that are not available on a TV set (e.g., touch screen, GPS receiver, camera, etc.) may be removed from the interface requirements in the application manifest to facilitate application migration to a TV

[1] Android official site : https://www.android.com/

[2] AndroidTV official site : https://www.android.com/tv/

environment. AndroidTV users will access the Google application repository (i.e., Google play) to download new applications to their TV sets in a similar way to Android smartphones.

- **WebOS**[3] is an operating system developed by Palm also based on the Linux operating system. This platform was acquired by Hewlett-Packard to be integrated into its devices. With the advent of Android, Hewlett-Packard halted WebOS development and eventually sold it to LG, which included it into its smart TV sets.

Thus, WebOS is an operating system designed for integration in smartphones and other devices such as smart TVs. Application development is based on web standards like HTML5, JavaScript and CSS. Its design allows web applications to access the native functionality of the host system. Applications are executed through the webkit renderer, and the system is a multitasking one able to run up to 4 simultaneous applications. As a general rule, a WebOS application is composed of HTML5 code and functional blocks written in JavaScript. While there is an open source distribution named Open WebOS, the version integrated in LG appliances is closed. For application development in this environment, LG offers the Enyo Framework to the developer's community.

TV sets based on WebOS include a remote control equipped with a motion sensor that sets a pointer on the television screen in the style of the Nintendo Wii. Additionally, it implements a voice recognition system that allows voice interaction using certain commands. Users access the LG store to download applications.

- **FireOS**[4] is a Linux-based operating system developed by Amazon for mobile devices (e.g., Kindle) and smart TV devices like FireTV. It is a version of the Android operating system, specifically the Fire OS 3.0 version based on Android Jelly Bean (API level 17). It can be installed in Amazon devices only.

[3] WebOS official site: http://www.lg.com/uk/smarttv/index.html
[4] FireOS official site: https://developer.amazon.com/android-fireos

Applications are written in Java and can be downloaded from the Amazon app store. From the point of view of the application developer, this platform is analogous to Android TV, with the particularity that the applications are available exclusively from the Amazon store. While the operating system is based on an Android distribution, Amazon limits its installation to its own hardware. This, together with the difficulty of downloading apps from other repositories, dramatically reduces the target audience of this platform.

- **Firefox** OS[5] is an open source operating system based on Linux and the Mozilla Gecko technology. While initially designed for mobile devices, some smart TV manufacturers like Panasonic have already shown their interest in this platform. At the time of booting, the Gecko runtime environment is launched, thus allowing the execution of web applications. Developers may use HTML5, CSS and JavaScript standards for their developments. Each application installed on this operating system behaves like a web page. Firefox OS also introduced the WebAPI concept, through which a selection of system calls can be invoked from the web interface. This reduces the gap between current web applications and native applications that can be installed on this operating system.
- **Tizen**[6] is a Linux-based open source operating system supported by the Linux Foundation and companies such as Intel, Samsung, Huawei and Fujitsu, among others. While Tizen was initially conceived as a platform for developing web-based applications, from version 2.0 on it also supports native applications. With this enhancement, support for hybrid applications was also included. These latter applications include native components and web application components. Native application development is done in C++. For web applications, HTML5, CSS and JavaScript technologies may be used. Through a WebAPI web applications may access certain core features (e.g., Bluetooth support). The execution environment is based on webkit rendering. Samsung

[5] Firefox OS official site: https://www.mozilla.org/en-US/firefox/os/

[6] Tizen official site: https://www.tizen.org/

provides an SDK for application design. The ability to run native applications offers many advantages when it comes to access and interact with both the operating system and with other devices that can be connected to the TV set. Similarly, support of standards-based web applications greatly facilitates application design and deployment. The blend of these two approaches in hybrid applications allows programmers to take advantage of both paradigms.

- **Google TV** created high expectations due to the relevance of the developing company (Google Inc.). However, Google abandoned this line in favour of Android TV. In our case, a Google TV device was evaluated and discarded due to the fact that interactivity is achieved through web browsers, which in turn poses significant restrictions on the access to operating system resources.
- **Apple TV** is the TV software created by Apple Inc. In the case of Apple TV, application distribution is based on the Apple Store concept. Convenient support is already available to develop applications for it, but there are also very significant restrictions when connecting external devices. Also, in accordance with the user requirements, in order to provide them with apps with a great interaction experience and a rich set of features, this technology was disregarded.

Android is the base operating system for AndroidTV and FireOS. As a consequence, they can benefit from a rich set of development applications and an active developer's community (in the case of FireOS applications are limited to those available through the Amazon App Store).

Another trend identified is the adoption of web-as-a-platform paradigm by WebOS, FirefoxOS and Tizen. As in the previous case, many tools are ready available to support development. This, together with the Web APIs offered by these systems to provide access to the internals of the operating system, supports the development of complete and rich interactive applications.

All platforms discussed, either Android-based or web-as-a-platform, offer development environments similar to those found in the smartphone world. Thus, support is provided to access peripheral devices that will enhance the user experience (e.g., webcams, voice control, gesture control, etc.). In some cases, the integration of external applications with direct access to native functionalities is also possible to access devices like webcams or microphones.

In the smart TV world, there are application areas that require a more direct control of the operating system and the TV itself. For example, interaction with some applications has to take place while watching a broadcast channel. While this is not a dramatic limitation in the smartphone world, it is a relevant feature in the case of interactive TV (e.g. contests / games applications based on a TV show).

2.2. Smart-TV devices

Smart-TV is a special type of television with an operating system which adds to the TV a set of typical computer characteristics. Most of the smart-TVs on the today’s market have one of the operating systems reviewed in the previous section.

During the last years, each television manufacturer has bet for a particular operating system, even has supported its development economically. Therefore, there is a clear relationship between the TV brand and its software. Next table shows the environments of the main European television brands:

TV Brand	Operating system
LG	webOS
Panasonic	Firefox OS
SAMSUNG	TIZEN
SONY	androidtv

The main risk to buying a smart-TV is the market volatility, because it is evolving continuously and, there is not a clearly dominant technology or brand. Android is the operating system that presents more facilities to developing services for it. However, the best-selling television brands in Europe are Samsung and LG[7] which have their own software - Tizen and WebOS respectively.

Furthermore smart-TVs, there are other devices on the market that allow you to convert any television with an HDMI port, into a smart TV. Most of these devices have an Android operating system, but there are devices with Apple-TV or even Tizen. In general, these devices are cheaper than smart-TVs, therefore they are a great alternative to enjoy the advantages of smart TVs at a low cost.

Here are different software and hardware parameters to take into account when selecting the device to covers the formal and informal service requirement. The formal services are the most demanding because they use special sensors to communicate with devices. The main important issues to review are:

- Operating systems.
- Openness.
- Easiness develop.
- Supports modify the start.
- In/out ports (Bluetooth, HDMI, Wi-Fi USB, etc.).
- Internal and SD storage.
- Memory size and processing capacity.
- External peripherals support (webcam, microphone, remote control, etc.).
- Tv antenna.

The tables below present the general characteristics of the HDMI devices analysed during months M1 to M6.

Hardware characteristics										
Device name	**Memory GB**	**Storage GB**	**Wi-Fi**	**BT**	**HDMI**	**SD**	**USB**	**Webcam**	**External peripherals**	**RF input**
Chromebit CS10	2	16	yes	yes	yes	no	yes	yes	yes	no
Rikomagic v5	2	16	yes	yes	yes	yes	yes	yes	yes	no
Guleek A8	2	8	yes	no	yes	yes	yes	no	yes	no

[7] https://www.statista.com/statistics/267095/global-market-share-of-lcd-tv-manufacturers/

Nvidia Shield	3	500	yes	yes	yes	yes	yes	yes	yes	no
Amazon Fire TV	2	8	yes	yes	yes	yes	yes	no	yes	no
Samsung HomeSync		1 TB	yes	yes	yes	no	yes	no	no	no
Apple TV		32/64	yes	yes	yes	no	yes	no	yes	no
Nexus player	1	8	yes	yes	yes	no	yes	no	yes	no
Raspberry pi 3	1		yes	yes	yes	yes	yes	yes	yes	no
Google TV Box MINIX NEO	2	16	yes	yes	yes	yes	yes	yes	yes	no
MINIX Neo X8-H Plus	2	16	yes	yes	yes	yes	yes	yes	yes	no
Roku4	512Mb	0	yes		yes	yes	yes	no	no	no
Asus Cube v2			yes	no	yes	no	yes	no		no
Boxee Box	1		yes	no	yes	yes	yes	no		no
Udoo	1	SD	yes	yes	yes	yes	yes	yes		no

Software characteristics									
Device name	**OS**	**Cordova**	**SDK**	**app store**	**3rd party libs**	**Openness**	**Easy dev**	**Community**	**Modify start**
Chromebit CS10	Chrome OS	yes	yes	yes	no	yes	yes	no	no
Rikomagic v5	Android, Ubuntu	yes	yes	yes	no	yes	yes	no	yes
Guleek A8	Android	yes	yes	yes	no	yes	yes	no	no
Nvidia Shield	Android TV	yes	yes	yes	yes	yes	yes	yes	no
Amazon Fire TV	Fire OS	yes	yes	yes	yes	yes	yes	yes	yes
Samsung HomeSync	Android	yes	yes	yes	no	yes	yes	no	no
Apple TV	tvOS	no	yes	yes	yes	no	no	yes	no
Nexus player	Android	yes	yes	yes	no	yes	yes	yes	yes
Raspberry pi 3	Windows, Linux	yes/no	no	yes	yes	no/yes	yes	yes	yes
Google TV	Android	yes	yes	yes	yes	yes	yes	yes	yes

Box MINIX NEO		🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
MINIX Neo X8-H Plus	Android	yes	yes	yes	yes	yes	yes	yes	yes
Roku4	RokuOS	no			yes	no	yes	yes	no
Asus Cube v2	Android TV	yes	yes	yes	no	yes	yes	no	no
Boxee Box	Linux	yes	no	no	yes	yes	yes	yes	yes
Udoo	Android, Linux, Arduino	yes	yes	yes		yes	yes		yes

After the market analysis, we have decided to acquire HDMI devices with Android operating system for first piloting cycle. This decision was founded in three main reasons:

- They are low-cost devices whose software can be upgraded. This property reduces the risk of investing a large amount of money in devices that could be obsolete at the end of the project.
- Android supports applications developed using standard web technologies (see section 3) which allows reusing not only the code but also the development experience for other operating systems and devices.
- It is possible to modify the Android boot to run apps when devices start.

3. Development technologies

This section includes a general overview of the different technologies used to develop the initial version of the services includes in SENIOR-TV V1.

As we have seen in the previous section, there are many different technologies in the smart-TV market. Because of that, in this first version, we have decided to use technologies which allow to create different versions of the apps for several operating system with reduced effort.

The following subsections review the characteristics of the different approaches to implement smart-TV apps and, show the characteristics of the selected technologies for this initial version. They also include a detailed description of the implemented services:

- Ionic and Apache Cordova to develop Weather, News and Events services.

- Unity to implement Attentix and Episodix games.

3.1. Smart-TV development paradigms

The Smart-TV market is changing every day. As we have seen in section 2, there is a great variety of Smart-TV technologies which are progressing continuously. Therefore, there is not a unique approach to develop apps for Smart-TV and, the current approaches are based in to reused or extend the mobile app development. This is a valid approach because, as we have seen, the Smart-TV operating systems are almost the same of smartphone software: Android, iOS, Tizen, etc. with some special characteristic oriented to run in televisions.

There are three typical approaches to implement mobile apps and we can use all of these approaches to develop Smart-TV apps:

1. **Native apps**: are developed in a programming language native to the device and operating system, and require one specific app to be created for one target platform.
2. **Cross-platform apps**: use an intermediate language that is not native to the device's operating system, such as JavaScript, to implement the apps. This allows share some code across different platforms – for instance, across iOS and Android.
3. **Hybrid web apps**: are cross-platform apps which rendering the user interface using an embedded web browser and, which are developed using well-known technologies - HTML, CSS and JavaScript.

The choice of one of these options depending on different aspects such us the performance and hardware requirements, supporting devices and operating systems, available resources or economic restrictions.

1. **Native** apps are appropriate if you need a high level performance and speed, for instance graphics intensive apps like games. Or to use advanced features offered by the device and operating system (data storage, memory access or complex networking).

However, this approach is not indicated to support multiple devices, because needs to create different versions for each operating system. So, it requires a bigger investment for development, maintenance and testing.

2. **Cross-platform** is indicated for create apps which must run in different platforms and have high interface performance restrictions. Reduces the development time because the code can be shared between different versions of the apps across devices. However, delegates the access to the device and operating system features to the framework, so, in the end, the performance could be reduced.
3. **HTML5 Hybrid** is the best choice when apps have to run on different devices and operating systems. The base of this approach is the use of basic web technologies such as HTML, CSS and Javascript, but you can use more advanced libraries such us AngularJS or Bootstrap. Therefore, this paradigm allows us to create advanced interfaces with little effort. In addition, you can reuse most of the code, therefore the development time is greatly reduced.

   However, as in the case of Cross-platform, the access to the device and operating system is delegating to the framework, which reduces the response time. So, it is not recommended for implementing services which need a high level of performance.

The great variety of devices and operating systems presents in smart-TV market makes necessary using technologies which cover as many brands as possible. Neither HTML5 Hybrid nor Cross-platform approaches cover all the operating systems in the market. However, we decided to use these technologies because they allow us to achieve a greater number of systems.

We have used some hybrid technologies like Cordova and Ionic to develop the Weather, News and Events services and, Unity - a cross-platform technology - to implement the Attentix and Episodix games.

3.2. Apache Cordova

Apache Cordova[8] is an open-source mobile development framework created by Nitobi. Originally called PhoneGap it was bought by Adobe Systems in 2011, and later released as an open source version called Cordova. Contributors to the Apache Cordova project include, among others, Mozilla, BlackBerry, Google, IBM, Intel and, Microsoft.

Apache Cordova allows software programmers to use standard web technologies (HTML5, CSS3, and JavaScript) for cross-platform development. Instead of relying on platform-specific APIs like those in Android, iOS, etc. The resulting applications are hybrid because use webviews to render the layouts and they are packaged as apps for distribution and have access to native device APIs.

Applications execute within wrappers targeted to each platform, and rely on standards-compliant API bindings to access each device's capabilities such as accelerometer, camera, compass, file system, microphone, network status, etc.

The following diagram shows a high-level view of an Apache Cordova application architecture:

[8] Apache Cordova official site: https://cordova.apache.org/

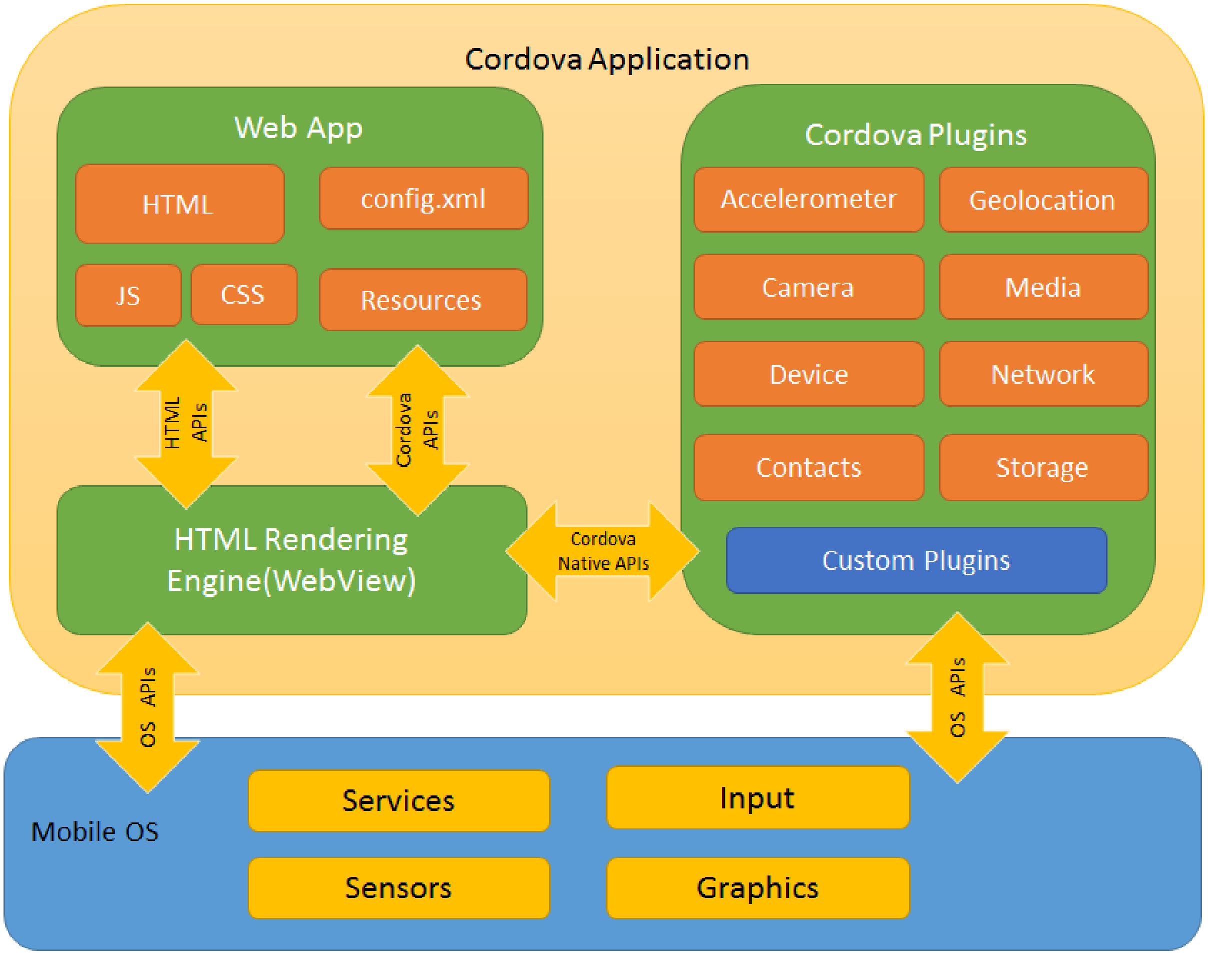


FIGURE 1. APACHE CORDOVA ARCHITECTURE

These are the Cordova architecture main components:

- Web App: the application code. Uses CSS3 and HTML5 to create the user interfaces and JavaScript implement the application logic.
- WebView: minimal browser that delivers web content and rendering the apps. HTML5 provides access to underlying hardware such as sensors, data or GPS. However, browsers' support is not consistent across mobile browsers of the oldest versions of operating systems. To overcome these limitations, Apache Cordova embeds the HTML5 code inside a native webview on the device.

- Plugins: provide an interface for Cordova and native components to communicate with each other, allowing invoke native code from JavaScript. Apache Cordova allows developers to extend its functionalities using native plugins that can be called from JavaScript[9].

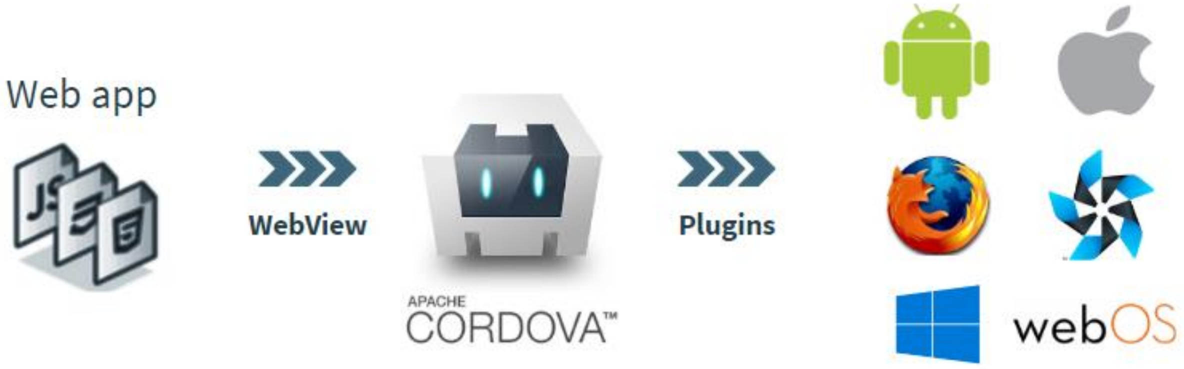


FIGURE 2. APACHE CORDOVA PLUGINS

The supported platforms by Cordova have been changing over the last few years. Previous versions have supported Android, iOS, Samsung Tizen, WebOS, Firefox OS, etc. However, only Android, Blackberry, iOS, OS X, Ubuntu and Windows 8 and 10 are supported in the last version. These limitations are not a deep problem, because Firefox OS is now discontinued. Also, Tizen has similar Cordova tools to develop hybrid apps. So, we can reuse the HTML5, CSS and most of JavaScript code to create Tizen apps.

However, the new Apple tvOS doesn't use webviews, so it is not possible develop apps using web resources, only native code is allowed. Therefore, this could be a potential problem to analyse in future versions of SENIOR-TV platform.

3.3. Ionic Framework

Ionic[10] is a powerful HTML5 SDK that is focused mainly on the look and feel, and UI app interaction. It helps you build native-feeling mobile apps using web technologies like HTML, CSS, and JavaScript. It

[9] Cordova Plugin development Guide: https://cordova.apache.org/docs/en/latest/guide/hybrid/plugins/

[10] Ionic official site: https://ionic.io/

doesn’t replace Cordova but fits in well with Cordova projects in order to simplify one big part of the app: the front-end.

Ionic was created by Drifty Co. in 2013. As Ionic is based on Angular, currently exists two versions of the framework in parallel: a stable version upon first generation of AngularJS (first release released in May 2015) and, a beta version based on Angular 2 (a new version that has not been released as stable until a few months ago).

Ionic includes mobile components, typography, interactive paradigms, and an extensible base theme. It also provides Angular custom components such as collection repeat, scroll-view, etc. Users can build their apps and customize for their favourite operating system and then deploy the apps using Cordova.

The last version supports several systems like Android 4.1 and up, iOS 7 and up, Windows 10 and Blackberry 10.

3.4. Unity

Unity 3D[11] engine is a tool to create games, applications and 3D animations. It is a multiplatform engine that allows you to deploy across major mobile, desktop, console, and TV platforms plus the Web.

The Unity editor is a complex visual editor to develop games. A game in Unity is structured in scenes that can be any part of the game, from a start menu to a level of your game or an area of al level. Unity engine also includes a terrain editor where you can create the terrain of your game using visual tools, painting, etc.

In Unity editor you can create your own objects or you can import them from different 3D platforms like Blender, 3ds Max, SketchUp, Maya and many more. Unity can read 3D formats like “.FBX”, “.OBJ” or “.dae” files so you can import 3D objects from any platforms that can export to these formats. Unity can also read proprietary 3D application files like “.Max”, “.Blend” or “.skp”.

[11] Unity 3D official site: https://unity3d.com/

The behaviour of a Unity game object is defined by his scripts -one or more- that you can develop in JavaScript or C#. All the physics and many other behaviours are also provided from the Unity engine.

Although you can also create your own methods and classes all Unity scripts derive from the base class “MonoBehavior”. This class provides some facilities to interact objects with unity engine including same methods to initialize whatever you want even before the scene of your game starts. The behavior of objects is mainly based on frames, that is, the script is continuously repeated frame by frame. Same methods are called every frame or even more per frame. After initialization you can control physics, input events, game logic or even the rendering of the objects.

Unity 3D is a 3D game engine officially launched on June 2005. This engine allows to create games and other interactive content or 3D animations in real time. Many people interested in development find the difficulty of learning programming languages and the engines that use them. The creation of video games become very difficult without programming studies or computer animation skills.

Unity Technologies is one of the companies that has decided to change this situation. The Unity development team has decided to maintain the source code but offering the user a complete graphical interface so that users can control the source code without creating new elements in the code. This is what has made Unity so popular for video game developers.

Unity is a 3D real time and multimedia application besides being a 3D and physics engine to create games, animations in real time with audio, video and 3d objects content.

This engine does not allow complex modelling but you can create scenes with illumination, terrains, cameras, textures, etc. Unity allows to deploy video games to Windows, Mac OS, iOS, Android, Wii, PlayStation, Xbox 360, Nintendo, Web and same TV platforms.

Unity has several advantages that make it one of the most used video game engines at this moment. Some of these advantages are:

- Allows to import several 3D formats like 3ds Max, Maya, Blender, SketchUp, FBX, etc. and you also can import some resources like Photoshop textures, PNG, TIFF, audios and videos.
- It is compatible with graphic Direct3D, OpenGL and Wii graphic APIs. It is also compatible with QuickTime and use internally Ogg Vorbis format.
- In Unity, the game is built using the editor and a scripting language so user does not have to be a programming expert to create a video game. Scripts can be created in JavaScript or in C# language.
- The Unity game structure is defined by scenes that represent some part of the game.
- Includes a terrain editor that allows you to create a terrain from the beginning, making his geometry, texture and including 3D elements that you can import from other 3D applications or elements already predefined in Unity.
- There are several Unity versions but the simplest is free for personal use and let you create commercial games till you get $100k per fiscal year.
- As we said Unity is multiplatform, so you can easily deploy on multiple mobile platforms, desktop, etc.

4. Informal services first apps

Elderly people will be the main users of SENIOR-TV services. The majority of time, they will access the platform through smart-TV. However, certain applications will allow access to the platform using other smart media such as mobile or tablets. However, secondary users (relatives and caregivers) will normally use their smartphones or tablets access to SENIOR-TV (to communicate with the elders or to access any other particular services). The next diagram shows the principal users of SENIOR-TV and how each kind of user will access to the platform.

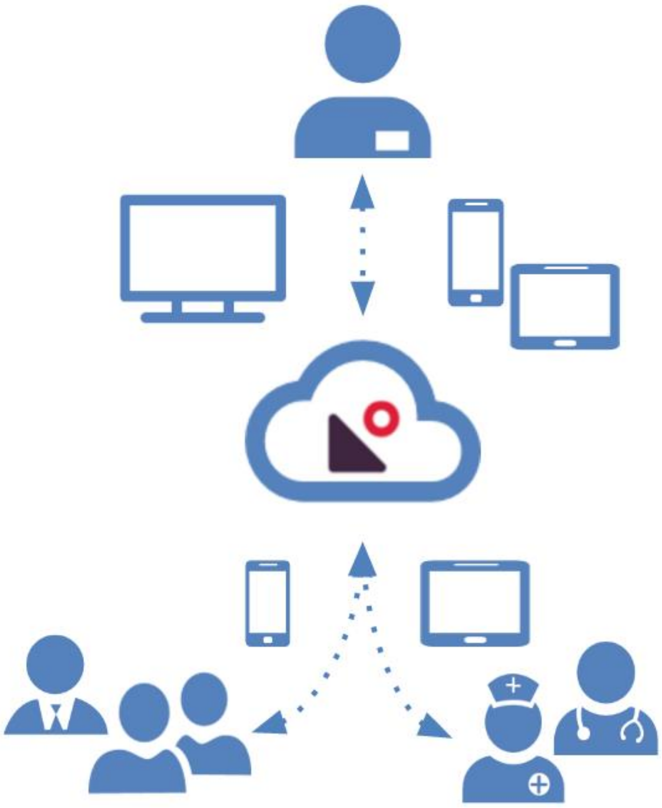


FIGURE 3. PLATFORM INTERACTION

However, the platform obtained after the first cycle only includes services which run in the smart-TV devices, as you can see in the following chart.

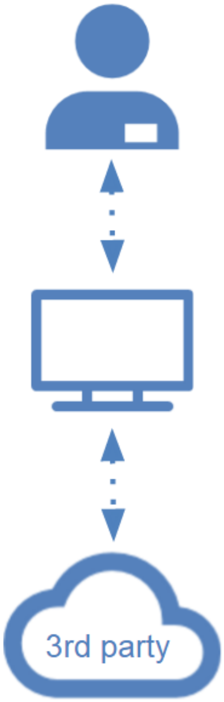


FIGURE 4. CURRENT APPS INTERACTION

The aim of the first version of SENIOR-TV was developed the initial version of a set of informal services to be tested during the first cycle of WP3 pilots by seniors from the three pilot countries (Cyprus, Slovenia and Romania).

This first version does not include complete functional services, but it includes initial versions like clickable mock-ups, which are going to improve in next versions using the feedback obtained from seniors after the pilots. They will evaluate issues such as the look and feel colours, the type and size of fonts, usability, the interaction devices, the contents, etc.

The WP1 workshops provided valuable input from the users, based on which we selected the services to be implemented in the first iteration According to those results, the services most demanded by the elderly, regardless of their sex and age, are: the weather, the news and events, apart from games and entertainment services in general.

The following subcategories show the main characteristics and the current state of development of the services included in SENIOR-TV V1.

4.1. News App

The News App scope is to facilitate the viewing of news in an easily manner on TV, while also taking into consideration personal preferences and interests. The elderly user is able to choose which category of news he wants to follow and which news to read.

The app runs directly on the Set Top Box and fetches the News from different RSS News Websites. The app and news feeds are localized, meaning that the users from different countries see the news from their own countries and in their own language.

The next diagram shows the architecture of the News app:

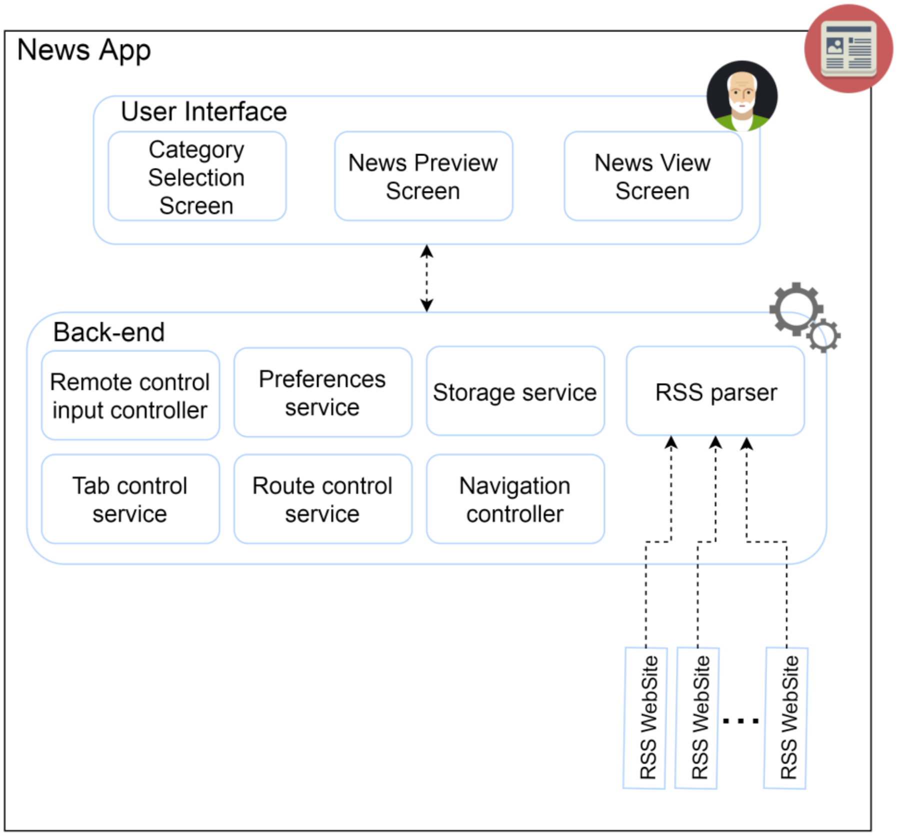


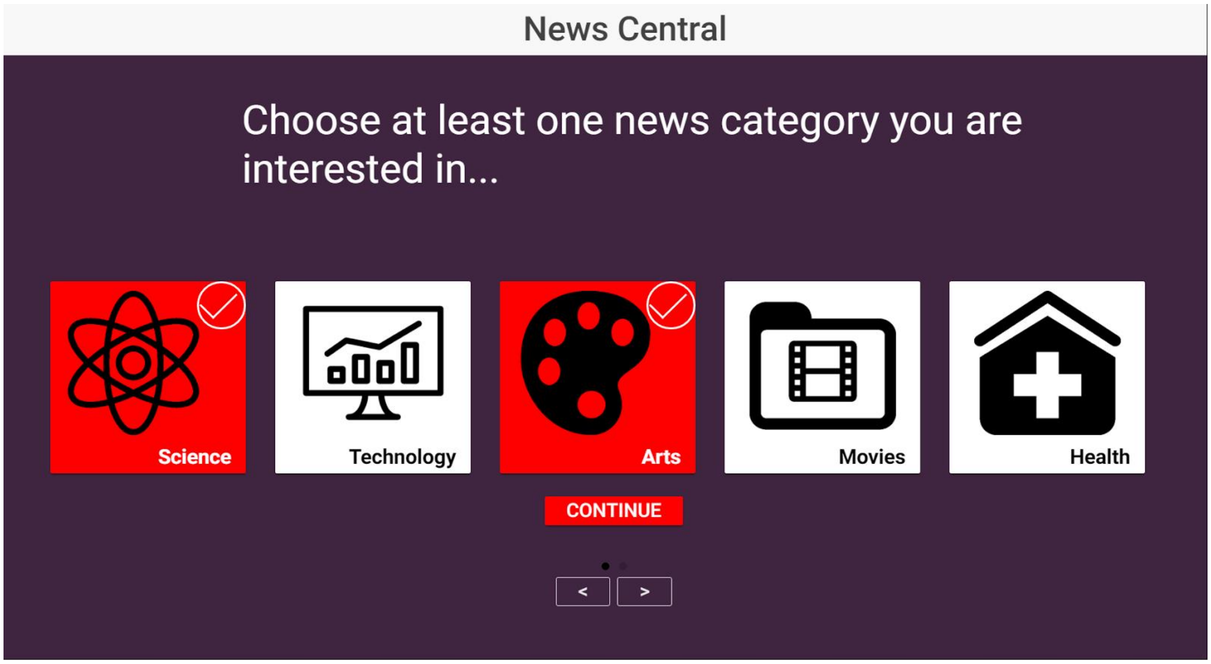
FIGURE 5. NEWS APP ARCHITECTURE

The back-end of the News app has the following components:

- Remote control input controller: is the service that handles the user input and transform the input into actions inside the app.
- Preferences service: is responsible for managing the preferences of the user (e.g. language).
- Storage service: is responsible for storing information, locally on the device, about the interaction with the app and also about each session the user has with the app.
- RSS parser: is in charge of fetching news from RSS feeds and parsing them for presentation on the user interface.
- Tab control service: is responsible for managing the switching between different types of news categories.
- Route control service and Navigation Controller: allow the user to navigate from one page of the app to another page of the app with ease.

The application has one screen for viewing the news information and two main screens for browsing the news and the preferences:

1. **The Category Selection Screen**

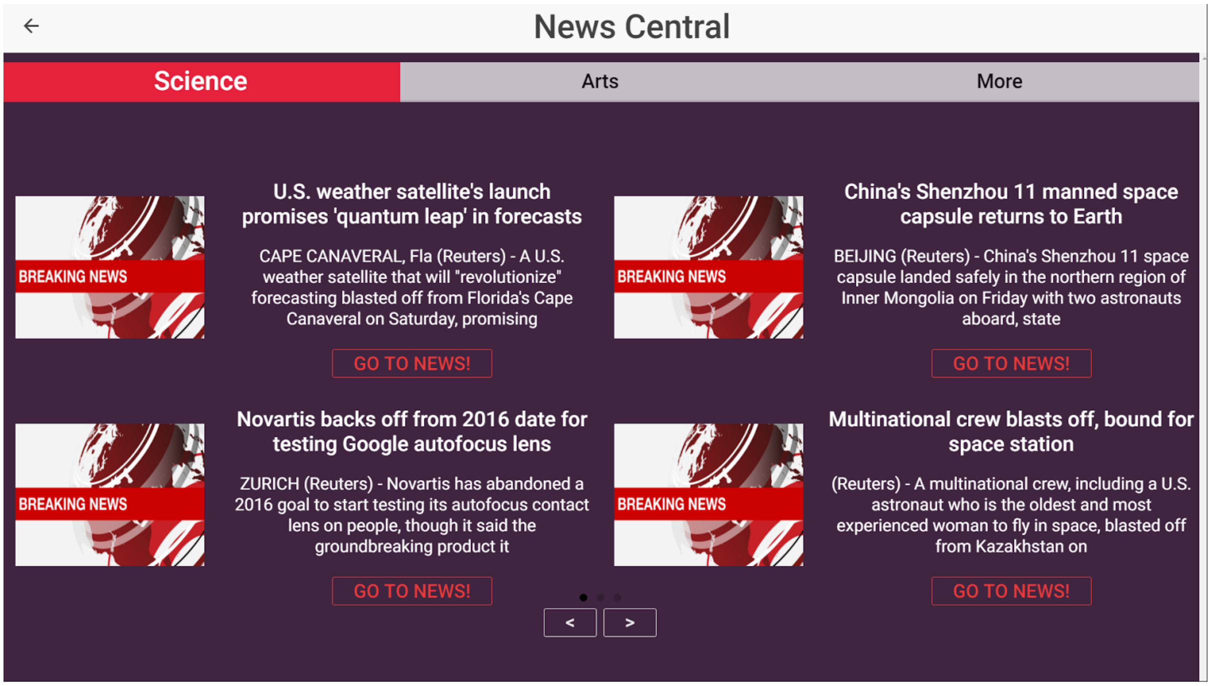


News Central

Choose at least one news category you are interested in...

FIGURE 6. NEWS APP CENTRAL SCREEN

2. **The News View Screen**



U.S. weather satellite's launch promises 'quantum leap' in forecasts

CAPE CANAVERAL, Fla (Reuters) - A U.S. weather satellite that will "revolutionize" forecasting blasted off from Florida's Cape Canaveral on Saturday, promising

China's Shenzhou 11 manned space capsule returns to Earth

BEIJING (Reuters) - China's Shenzhou 11 space capsule landed safely in the northern region of Inner Mongolia on Friday with two astronauts aboard, state

Novartis backs off from 2016 date for testing Google autofocus lens

ZURICH (Reuters) - Novartis has abandoned a 2016 goal to start testing its autofocus contact lens on people, though it said the groundbreaking product it

Multinational crew blasts off, bound for space station

(Reuters) - A multinational crew, including a U.S. astronaut who is the oldest and most experienced woman to fly in space, blasted off from Kazakhstan on

FIGURE 7. NEWS APP CENTRAL SCREEN 2

4.2. Events app

The scope of this app is to enable elderly users to find and participate to events that they are interested in by just selecting an event on the TV screen.

The app is also localized, meaning that it will only show events accessible to the elderly user and in his own language. During the writing of this report we are also investigating the possibility of enabling users to participate to online events.

The following chart shows the architecture of the News app:

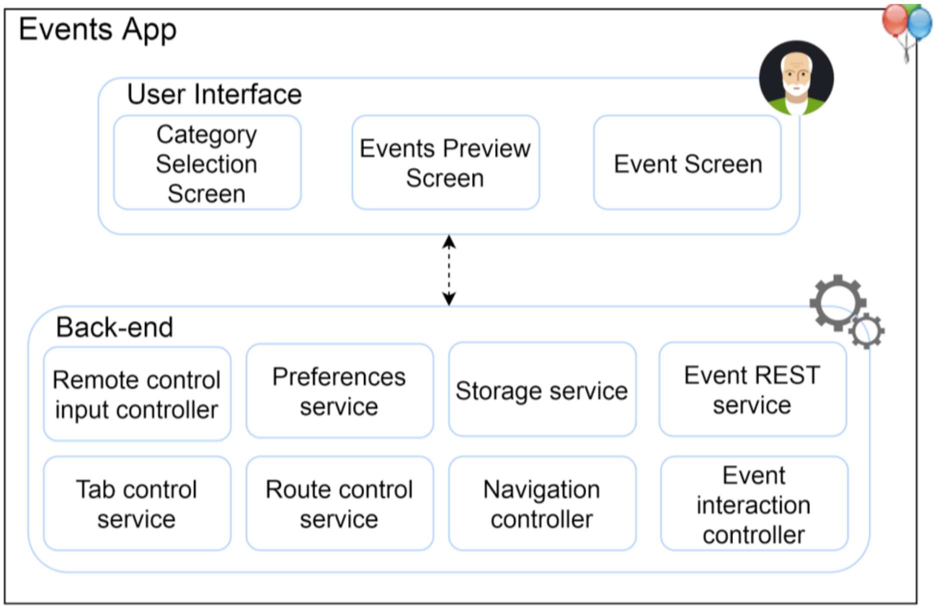


FIGURE 8. EVENTS APP ARCHITECTURE

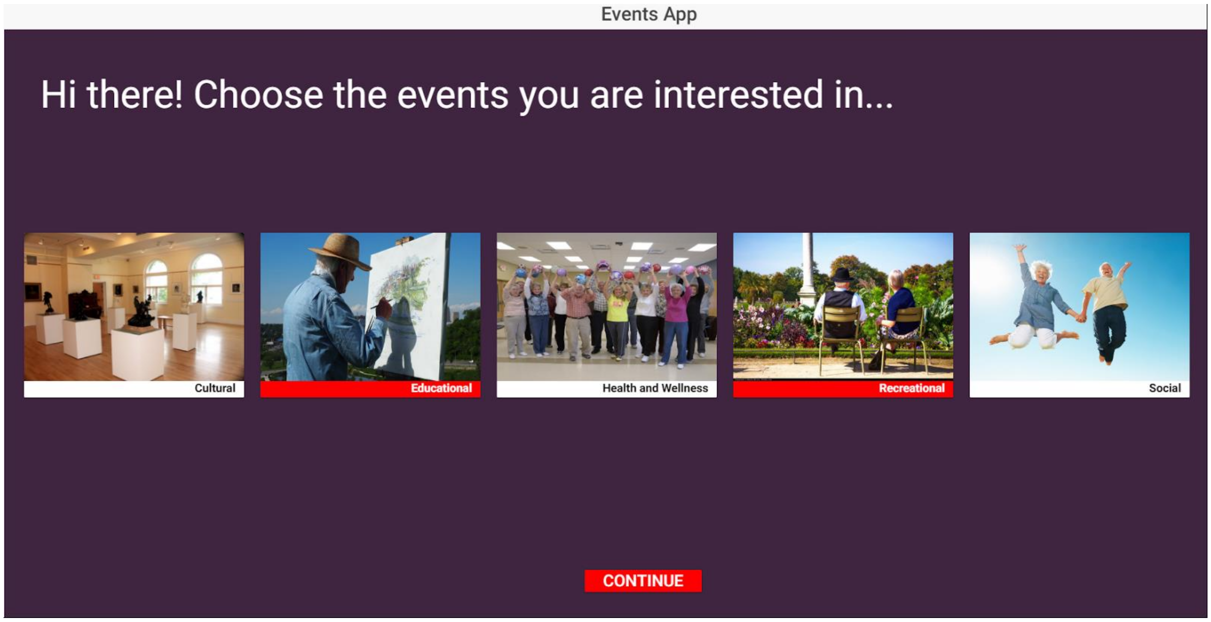
The Back-end of the Events App has the following components:

- Remote control input controller: is the service that handles the user input and transform the input into actions inside the app.
- Preferences service: is responsible for managing the preferences of the user (e.g. language).
- Storage service: is responsible for storing information, locally on the device, about the interaction with the app and also about each session the user has with the app.
- Tab control service: is responsible for managing the switching between different types of news categories.
- Route control service and Navigation Controller: allow the user to navigate from one page of the app to another page of the app with ease.
- Event Interaction Controller: is responsible for enabling the user to interact with a specific event. The user will be able to send a notification for *Participation* to the Event Organisers.
- Event REST service will be responsible for fetching and parsing different event metadata (e.g. location, date, availability, price, etc.). As we are writing this report we are investigating the

implementation of an Event Cloud Platform that will enable different Event Providers for elderly to add their events to the SENIOR-TV Events App.

The main screens of the user interface are as follows:

1. The Category Selection Screen

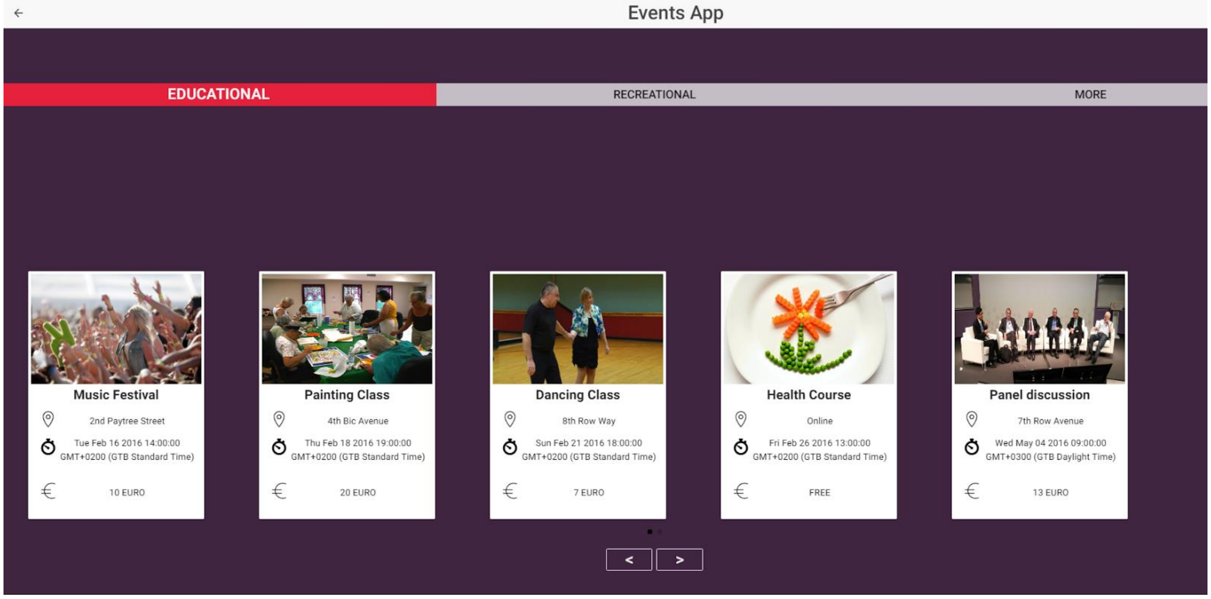


Events App

Hi there! Choose the events you are interested in...

FIGURE 9. EVENTS APP SELECTION SCREEN

2. The Events Preview Screen



Events App

Music Festival
2nd Paytree Street
Tue Feb 16 2016 14:00:00 GMT+0200 (GTB Standard Time)
10 EURO

Painting Class
4th Bic Avenue
Thu Feb 18 2016 19:00:00 GMT+0200 (GTB Standard Time)
20 EURO

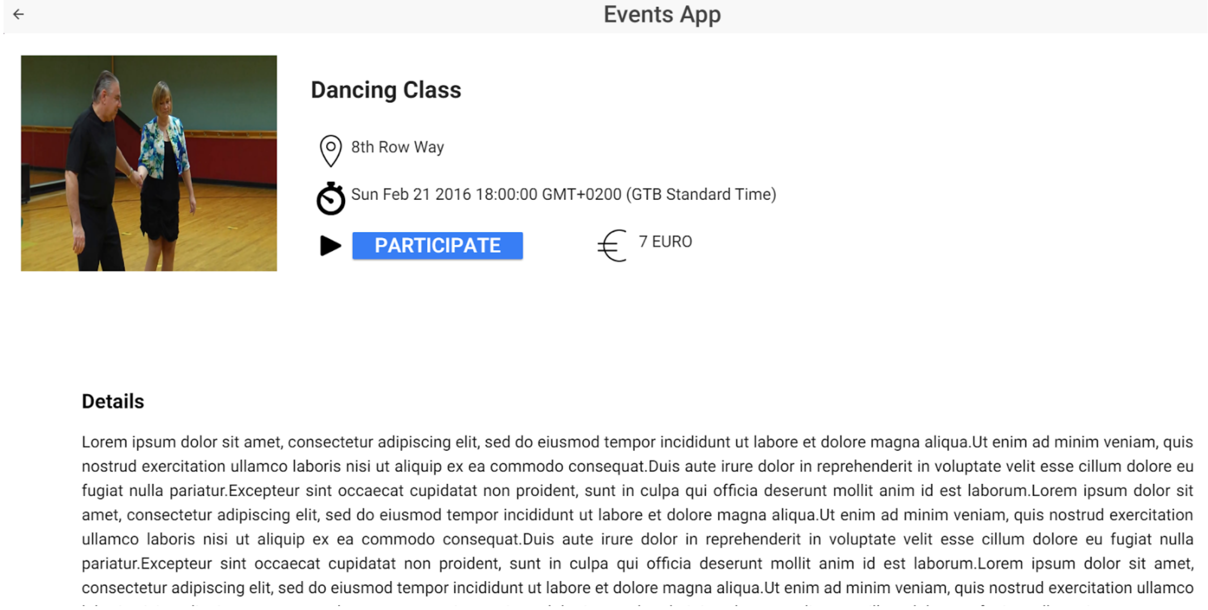
Dancing Class
8th Row Way
Sun Feb 21 2016 18:00:00 GMT+0200 (GTB Standard Time)
7 EURO

Health Course
Online
Fri Feb 26 2016 13:00:00 GMT+0200 (GTB Standard Time)
FREE

Panel discussion
7th Row Avenue
Wed May 04 2016 09:00:00 GMT+0300 (GTB Daylight Time)
13 EURO

FIGURE 10. EVENTS APP PREVIEW SCREEN

3. The Events View Screen



Events App

Dancing Class

8th Row Way

Sun Feb 21 2016 18:00:00 GMT+0200 (GTB Standard Time)

7 EURO

Details

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.Ut enim ad minim veniam, quis nostrud exercitation ullamco

FIGURE 11. EVENTS VIEW SCREEN

4.3. Weather app

This application allows seniors to check the weather conditions for their village or city. The elders don't need to select their location, the app is able to get the users' position and ask the forecast automatically.

The app runs directly on the Set Top Box and fetches the weather data via HTTP requests to an open weather API free service, called *OpenWeatherMap*[12]. The Weather app are localized, meaning that the users from different countries see the information in their own language and the senior's location name is shown on the right top corner.

The next chart shows the architecture of the News app:

[12] OpenWeatherMap official site: https://openweathermap.org/

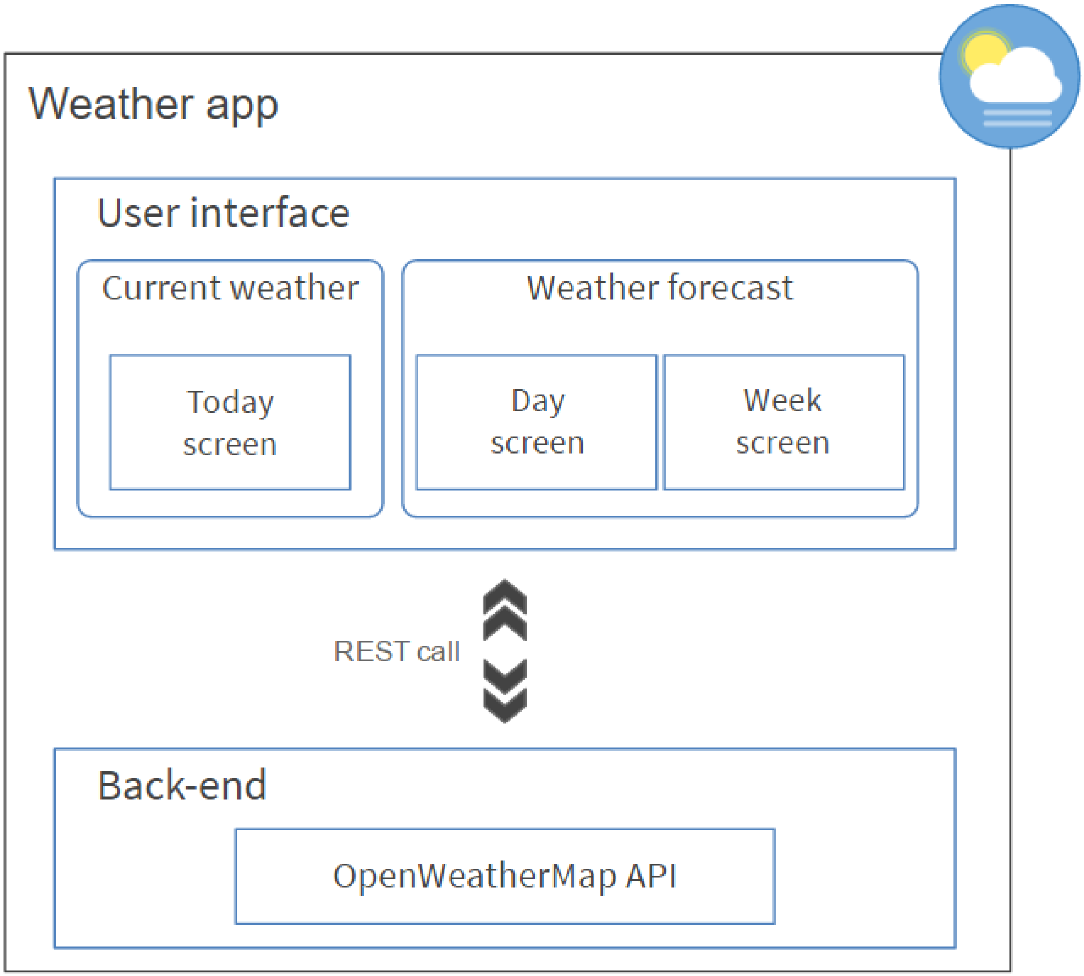


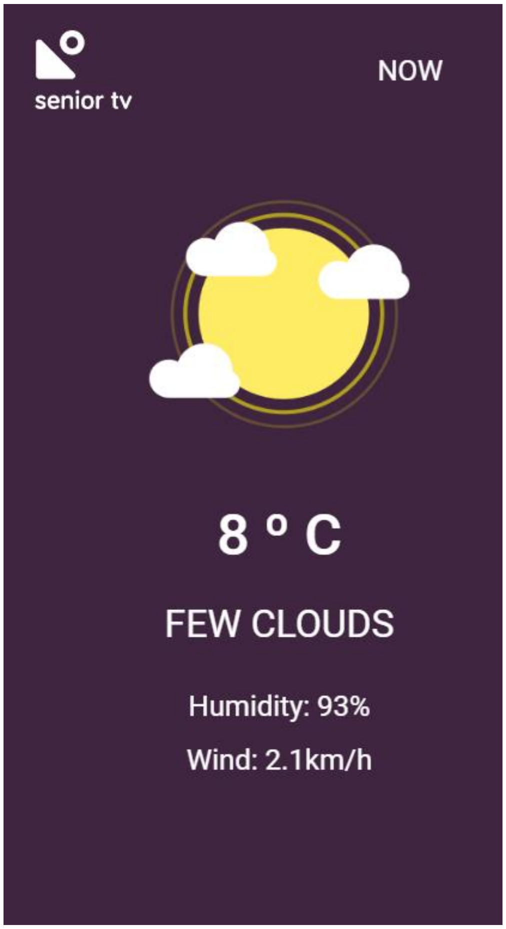
FIGURE 12. WHEATEAR APP ARCHITECTURE

The user interface is divided in two sections:

- On the left: shows the current weather conditions.
- On the right: shows the forecast for the current day or for the next three days.

1. The current weather screen

Shows the temperature, humidity and wind forecast for the senior location, in this moment.



NOW

8 º C

FEW CLOUDS

Humidity: 93%

Wind: 2.1km/h

FIGURE 13. WEATHER APP CURRENT SCREEN

2. The weather forecast screen

Shows the weather forecast for the current day, or for the next three days if we choose so. The user can navigate between this options using the top of the screen buttons: “Day” and “Week”.

The user can switch the screens pressing the corresponding button, or using the right and left key arrows of the remote control.

- **Day screen**

In the case of the current day state, the information shown consists in a line chart with the temperature forecast for the next hours.

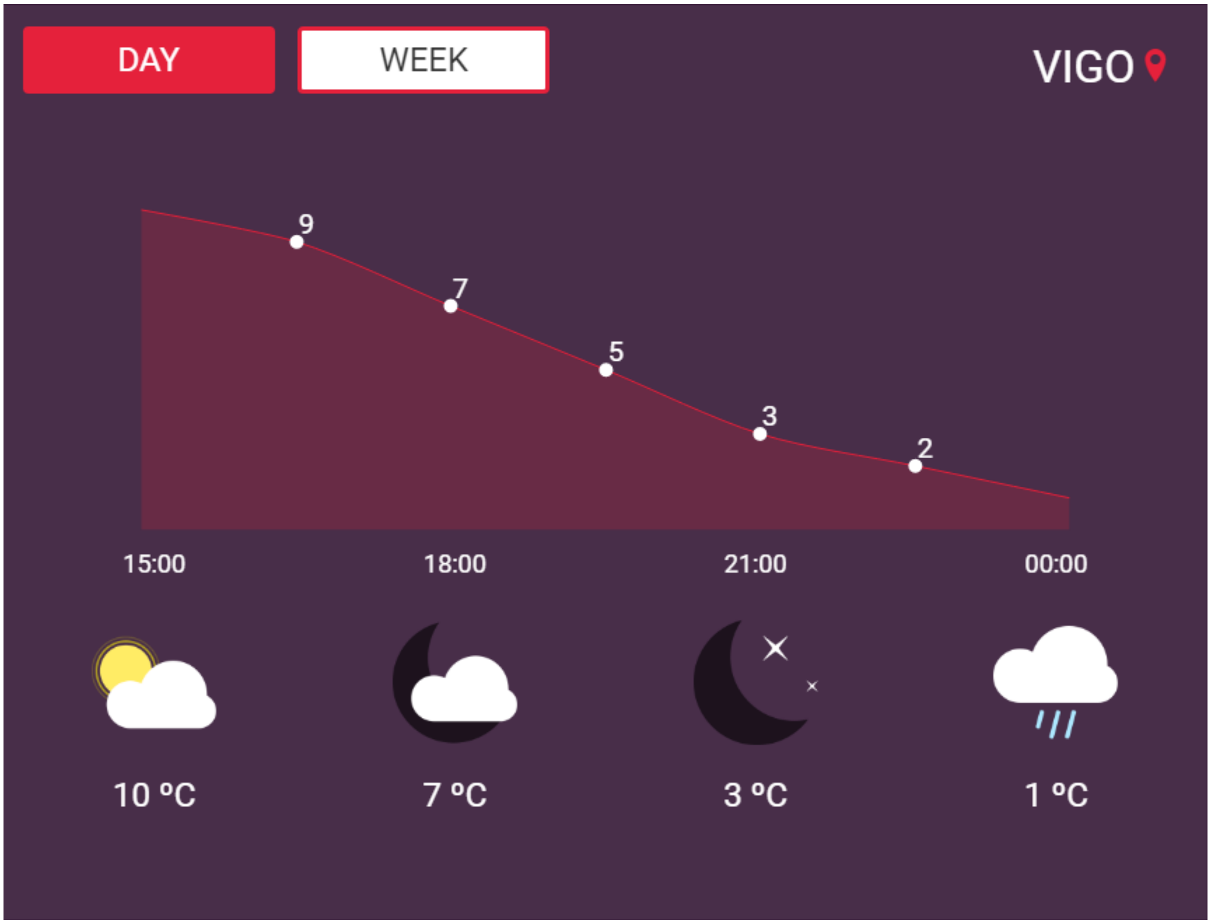
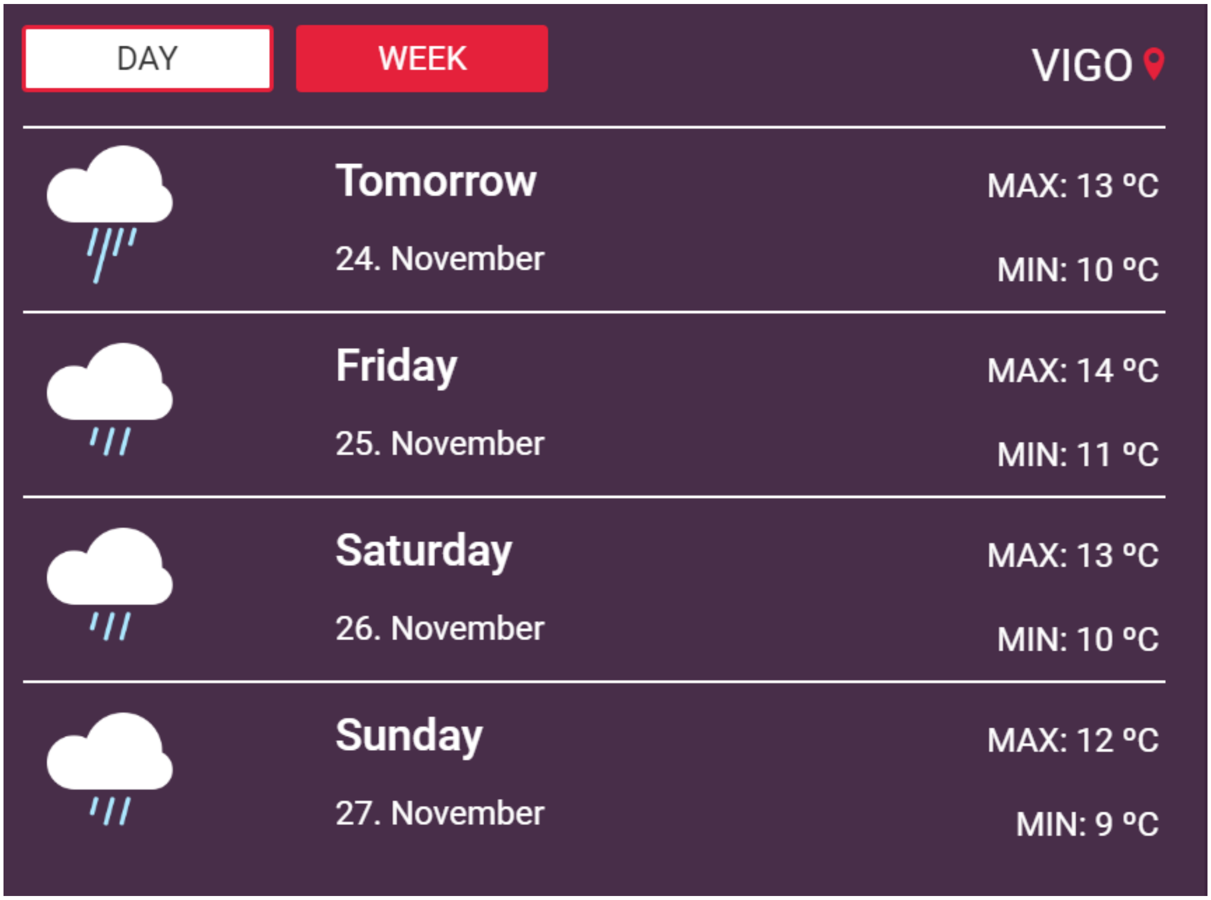


FIGURE 14. WEATHER APP DAY SCREEN

- **Week screen**

Instead, for the week state the application shows the weather forecast and the max and min temperatures forecast.



DAY WEEK VIGO

Tomorrow
24. November
MAX: 13 ºC
MIN: 10 ºC

Friday
25. November
MAX: 14 ºC
MIN: 11 ºC

Saturday
26. November
MAX: 13 ºC
MIN: 10 ºC

Sunday
27. November
MAX: 12 ºC
MIN: 9 ºC

FIGURE 15. WEATHER APP WEEK SCREEN

4.4. Attentix

Attentix[13] is a game for improving memory skills. The game creates sequence of lights and asks the user to repeat it. If user do well, the game will increase number of lights in the sequence one by one. After that, asks a new sequence to the user. At the end, when the series generated. Users can always reach their maximum skill level in the game. You can select different levels. Each level adds new colour box in the screen.

[13] The Attentix game is co-financed within the SENIOR-TV project.

The next diagram shows the architecture of the Attentix game:

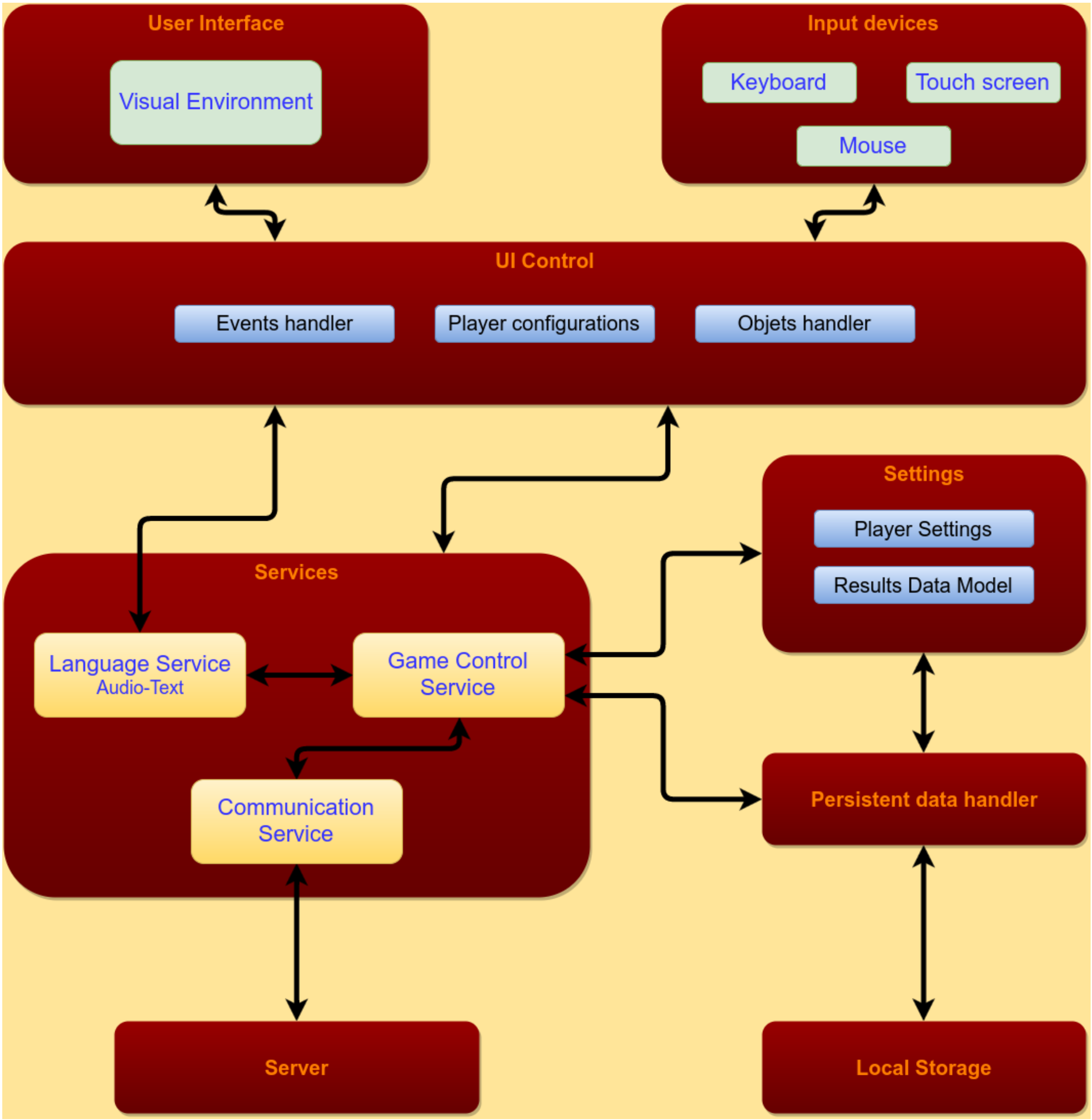


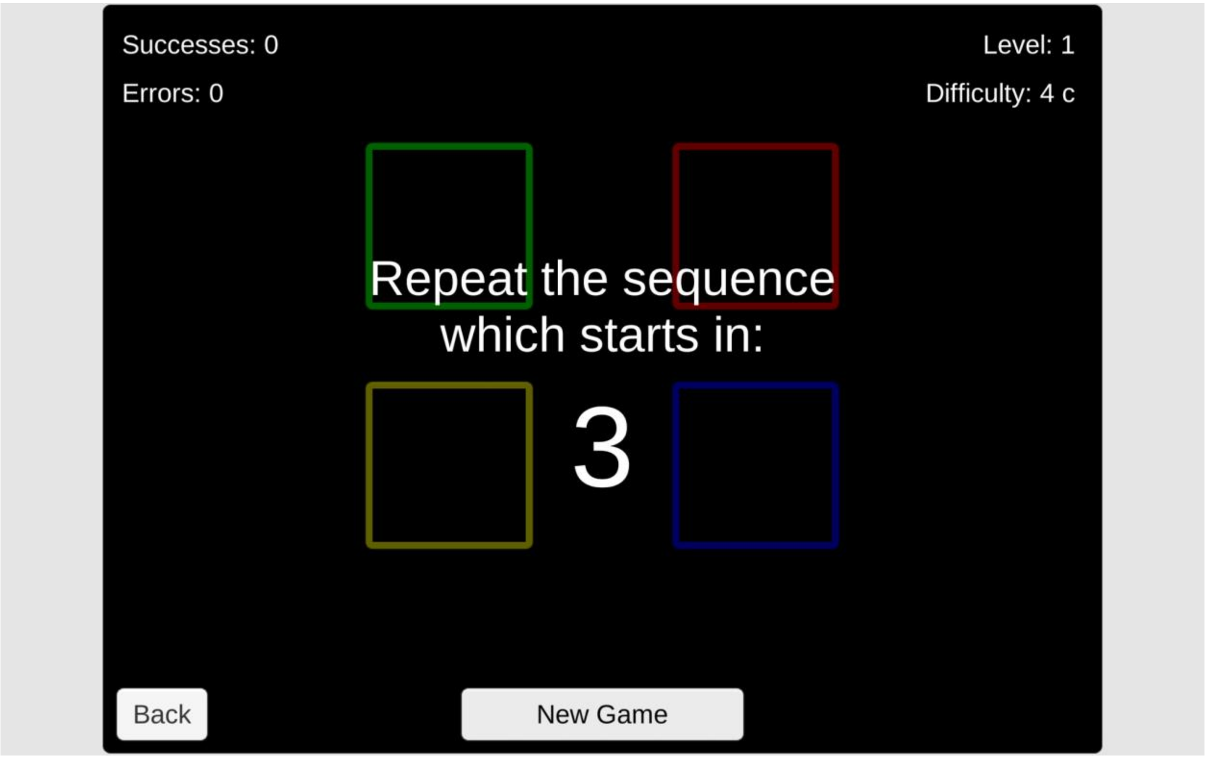
FIGURE 16. ATTENTIX GAME ARCHITECTURE

The previous chart shows the modules of the game and the relationships between them.

- The "Game Control Service" module is the core of the game. It is used by other services or controllers to get current game information or to set it. "Game Control Service" has some different objectives like read current player setting to set the language or to prepare the results to send to the server.

- The “User interface” block are those parts of the game responsible for the rendering of the game objects. This block is connected to the “Game Control Service” through the “UI Control” block to get object's positions for example.
- The “UI Control” is used to interact between user interface and the input. Also, it is responsible for connecting with the “Language Service” to get the current translation for the words or sentences to show on screen. This block has to create the sequence to represent and control if is correctly pressed.
- The “Settings” block is to represent the data model to save player setting or how to send results to the server.
- The “Communication Service” is responsible for sending the game results to the server.
- The “Persistent data handler” is the block responsible for storing results or player settings into the device. The data is saved in a persistent data folder with the package name.

The seniors have to use the pointer to interact with the game, clicking different squares on the screen. The next image shows the main screen of Attentix game:



Successes: 0

Errors: 0

Level: 1

Difficulty: 4 c

Repeat the sequence which starts in:

3

FIGURE 17. ATTENTIX GAME SCREEN

4.5. Episodix

Playing to Episodix[14]game, seniors can see a virtual city and walk around their streets. During their walk, some objects could appear and the senior have to remember which objects have appeared and which not. This game stimulates the elders' memory.

The next chart shows the architecture of the Episodix game:

[14] The Episodix game is co-financed within the SENIOR-TV project.

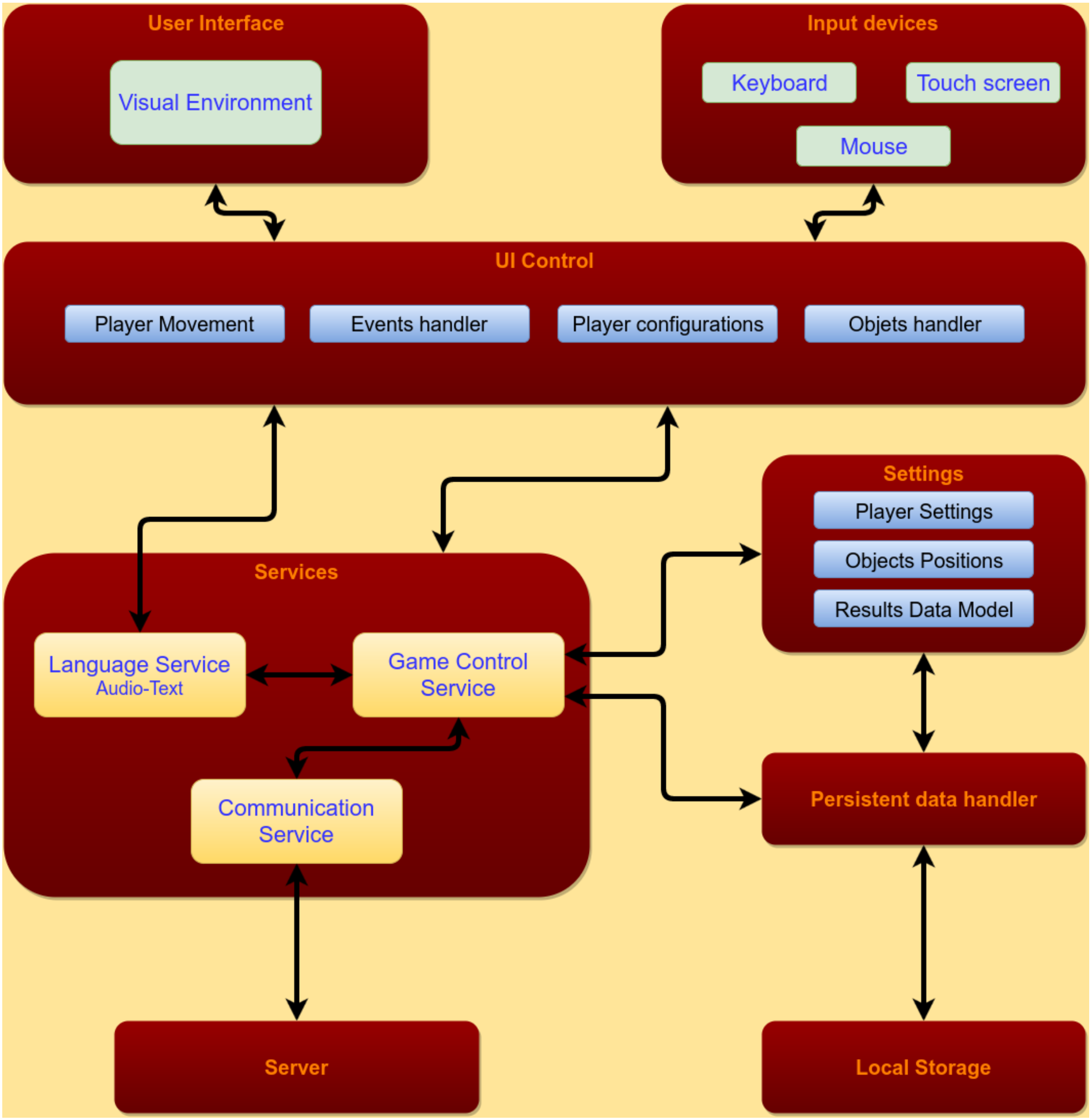


FIGURE 18. EPISODIX GAME ARCHITECTURE

- The "Game Control Service" is the core of the game. It is used by other services or controllers to get current game information or to set it. "Game Control Service" has some different objectives like read current player setting to set the language or to prepare the results to send to the server.
- The "User interface" block includes the parts of the game which are responsible for rendering the game objects. This block is connected to the "Game Control Service" using the "UI Control" block to get object's positions.

- The “UI Control” is used to interact between user interface and the input. Also, it is responsible for connecting with the “Language Service” to get the current translation for the words or sentences to show on screen. The player movement is controlled in this block.
- The “Settings” block represents the data model for saving the player settings.
- The “Communication Service” is responsible for sending the game results to the server.
- The “Persistent data handler” is responsible for storing results and the player settings into the device. The data is saved in a persistent data folder with the package name in the devices SD-card

The seniors have to use the pointer to interact with the game, selecting the up, down, left and right with arrows that appear on the screen. Following you can see the main screen of Episodix game:

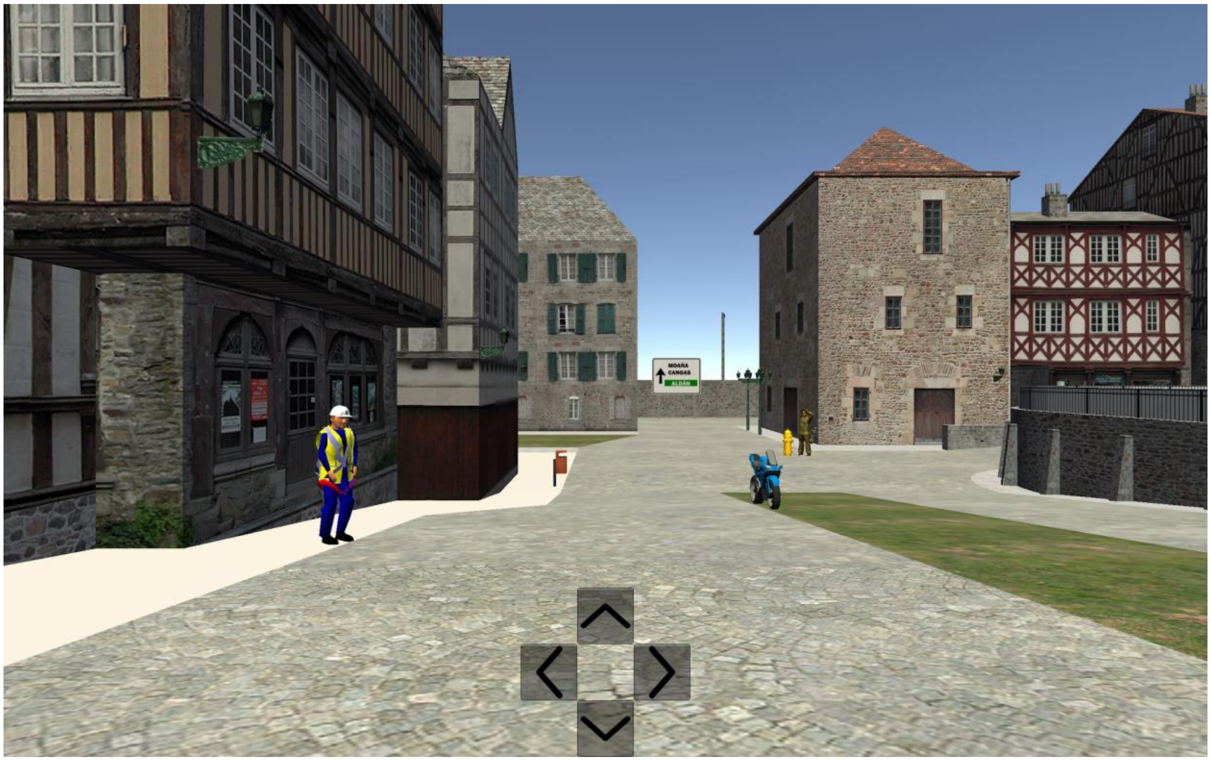


FIGURE 19. EPISODIX GAME SCREEN

5. Formal services

During the first cycle of the development of the formal services we analyzed the potential possibilities for developing formal apps for the SENIOR-TV platform.

This procedure took place taking in mind many aspects, such as:

- Market status and future trends
- End users outcome from the first iteration
- Extensibility of the app with existing technologies and future trends
- Integration of sensors so as to be used as input device in the app
- Data protection, confidentiality and ethics considerations.

During this period the following technical objectives have been identified:

Technical objective	Details
Definition of apps, data collection analysis and predictions	One of the formal bundle that we are interested to develop and consists of subcomponents is the health bundle. These consist of: a) Body Weight App b) Blood pressure app c) Pulse rate app d) Blood glucose level The initial idea is that this bundle will be implemented by using a set of sensors which are connected remotely to the STB (if finally SENIOR-TV will use an STB). The users will have the ability to view the parameters in their screen and if they wish then by just pushing a button they could

	transmit this info to e.g. their doctor. Then, the doctor through a web API could give feedback, monitor the status of the elderly, etc.
Selection of commercial of the shelf sensors for the selected applications	Selection criteria will include: ● Depending on the applications sensor types Power consumption evaluation of sensors and actuators for autonomy evaluation. ● Sampling frequency of sensors, accuracy and sample size. ● Integration possibilities
Selection of wireless technology for connectivity of the remote devices	Selection criteria will include: ● Low-power connectivity solutions. ● Depending on the use case corresponding design parameters such as radio transceivers density per square meter, and wireless link distance. ● Low packet error rates, and wireless capacity to fulfil sensor sampling/data rates. Mesh and star topologies with the corresponding routing protocols.
Integration of lightweight communication protocol COAP of the remote devices	Traditional Web protocols (HTTP/TCP) create heavy traffic for constrained devices, therefore lightweight Web protocols needed. Constrained Application Protocol (CoAP) is a software protocol intended to be used in very simple electronics devices that allows them to communicate interactively over the Internet.
Integration of 6LoWPAN in order the Internet	Low-power devices with limited processing capabilities should be able to participate in the Internet of Things. The 6LoWPAN group has defined

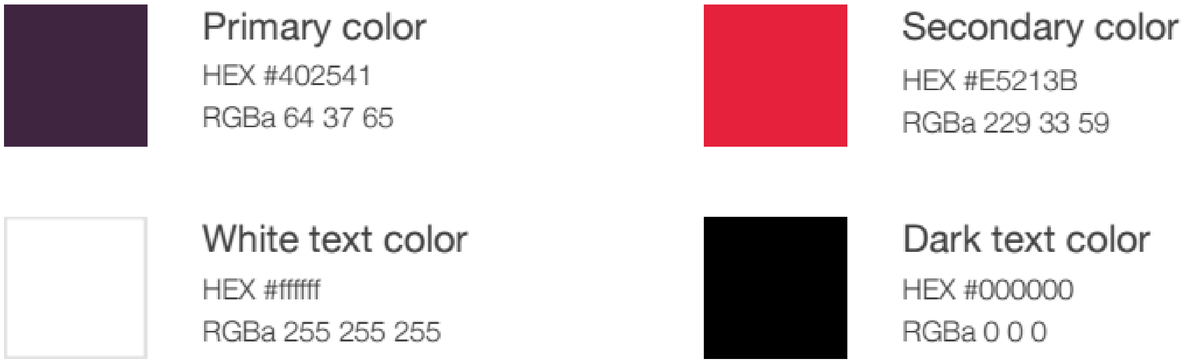
Protocol to be applied even to the smallest devices.	encapsulation and header compression mechanisms that allow IPv6 packets to be sent to and received from over IoT based networks. Other functionalities include: address resolution, routing considerations and protocols for mesh topologies, device, service discovery and security.
Cloud integration	System architecture and development for cloud data acquisition, storage and computation. Data stored in the cloud can be processed for developing business logics. Moreover, the data can be analyzed for anomaly detection, predictions, etc. depending on the use case.
Web API integration	Web API to retrieve the data and other information from the cloud and the network provides ease of integration by other nodes or free application development on the provided data.
Analytics and prediction dashboard	Optional feature – to be implemented in case there is still remaining Budget.

6. Ergonomic design

In this section, we present the user interface style guide used to develop all informal services included in this SENIOR-TV version, except the games.

6.1. General guides

6.1.1. Color swatches



Primary color
HEX #402541
RGBa 64 37 65

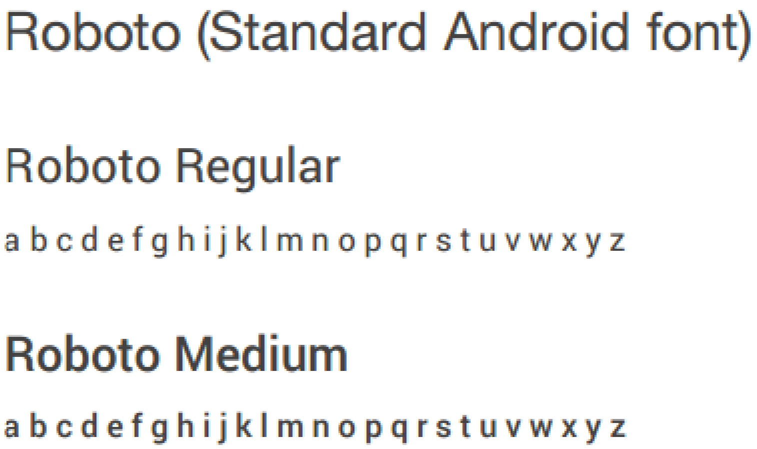
Secondary color
HEX #E5213B
RGBa 229 33 59

White text color
HEX #ffffff
RGBa 255 255 255

Dark text color
HEX #000000
RGBa 0 0 0

FIGURE 20. COLOR SWATCHES

6.1.2. Typography



Roboto (Standard Android font)

Roboto Regular

a b c d e f g h i j k l m n o p q r s t u v w x y z

Roboto Medium

a b c d e f g h i j k l m n o p q r s t u v w x y z

6.1.3. Button states

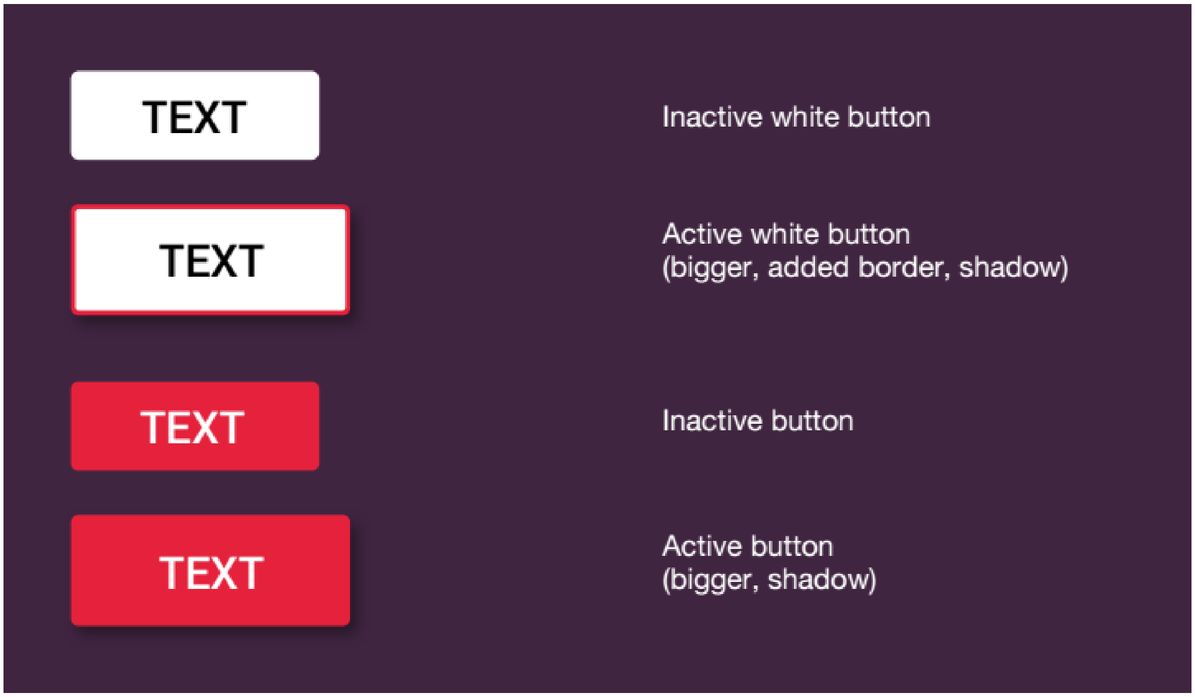


Figure 21. Button states

6.1.4. Navigation, focus and selection

Users navigate APP using a directional pad (D-Pad) and using the AirMouse Pointer. This type of controller limits movement to up, down, left, and right.

A key aspect of making application work well with a D-pad controller is to make sure that there is always an object that is obviously in focus. App must clearly indicate what object is focused, so users can easily see what action they can take. Use scale, shadow brightness, opacity, animation or a combination of these attributes to help users see a focused object.

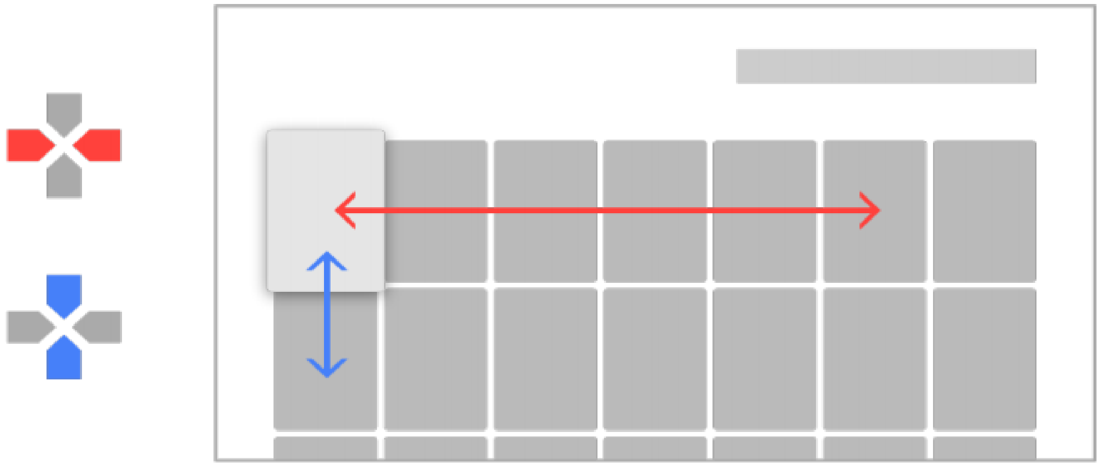


FIGURE 22. NAVIGATION

6.1.5. Overscan

During the evolution of TV technology, overscan originally described an area of TV content outside of a safe zone that most TVs could reliably display.

Minimum margin for the content is 5%. On a 1920 x 1080 screen, this margin should be a minimum of 27 pixels from the top and bottom edges and a minimum of 48 pixels from the right and left edges of the picture.

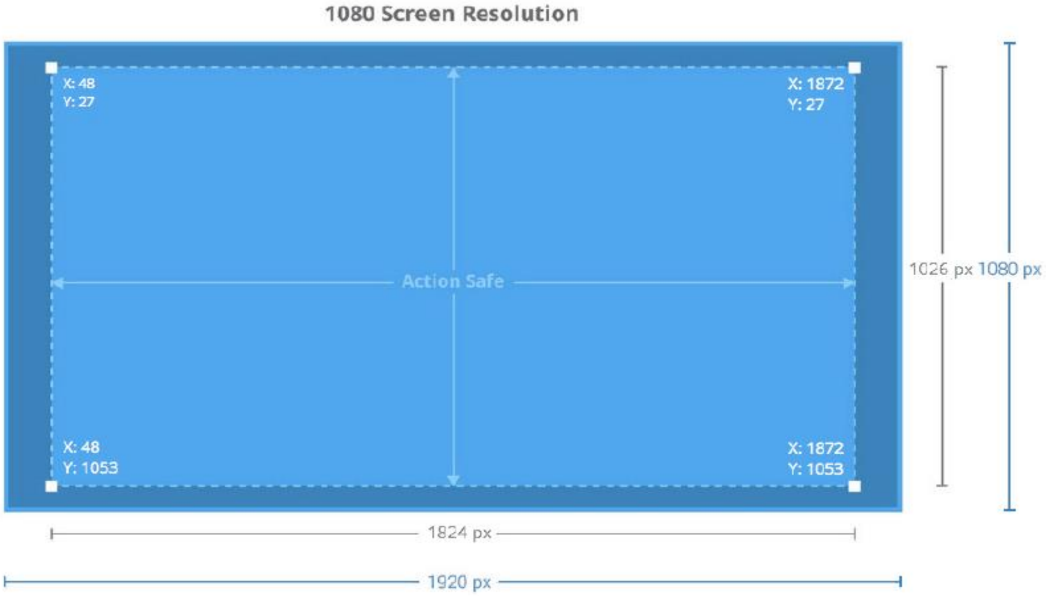


FIGURE 23. OVERSCAN

6.2. Apps design

6.2.1. Weather

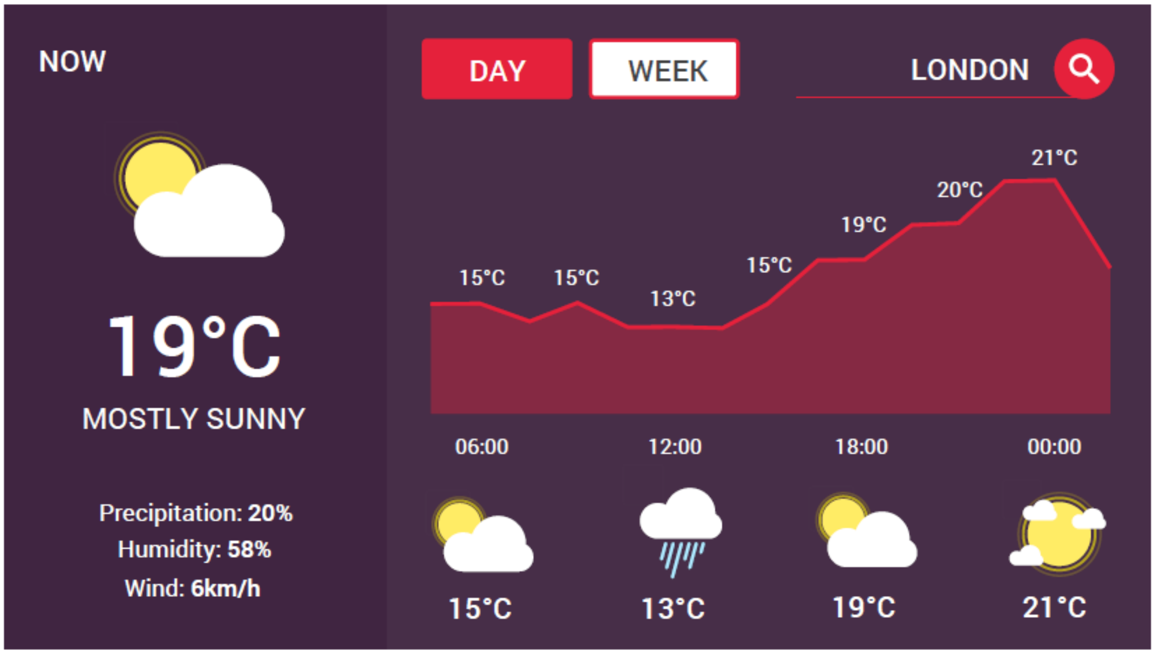
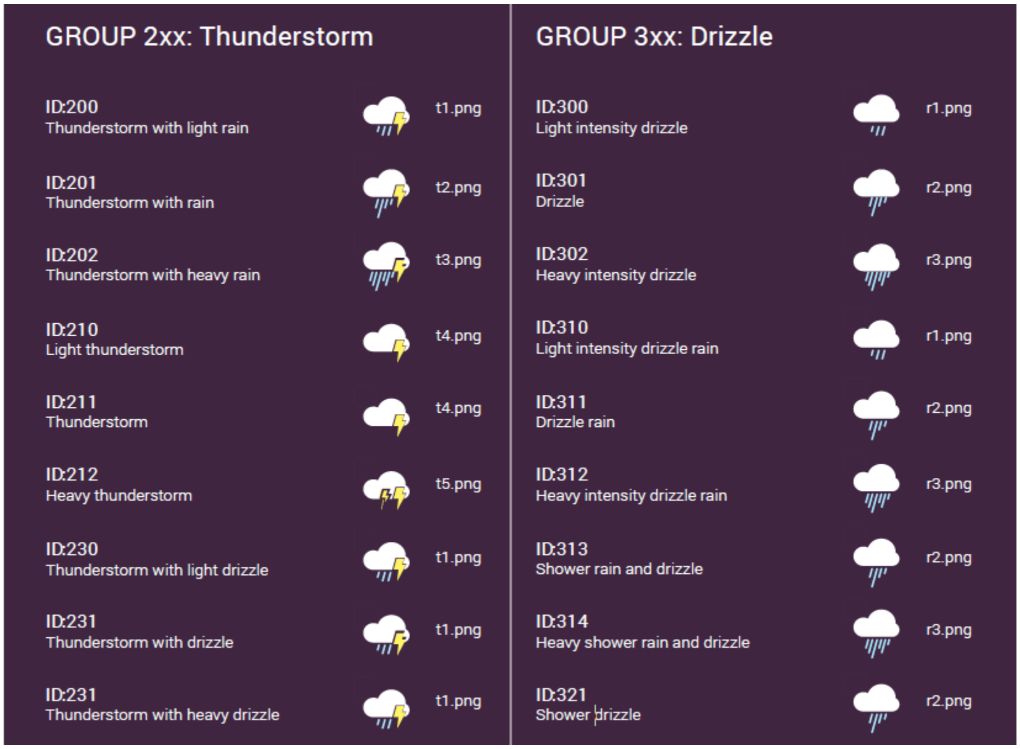
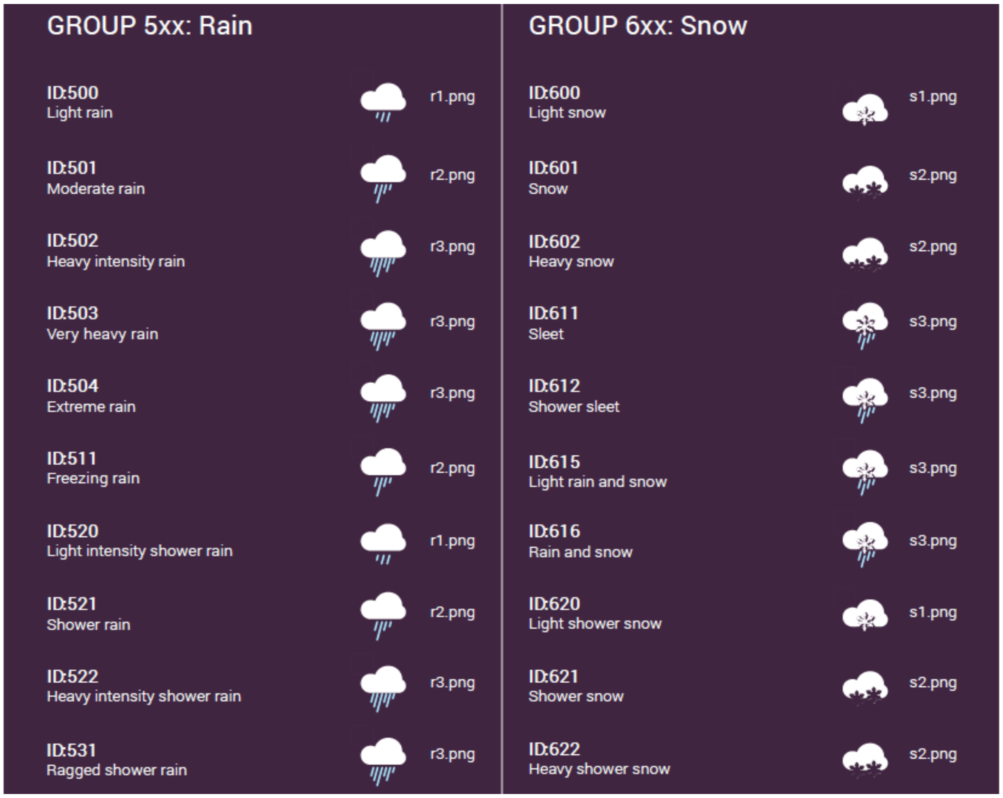


FIGURE 24. WEATHER APP DESIGN



GROUP 2xx: Thunderstorm		GROUP 3xx: Drizzle	
ID:200 Thunderstorm with light rain	t1.png	ID:300 Light intensity drizzle	r1.png
ID:201 Thunderstorm with rain	t2.png	ID:301 Drizzle	r2.png
ID:202 Thunderstorm with heavy rain	t3.png	ID:302 Heavy intensity drizzle	r3.png
ID:210 Light thunderstorm	t4.png	ID:310 Light intensity drizzle rain	r1.png
ID:211 Thunderstorm	t4.png	ID:311 Drizzle rain	r2.png
ID:212 Heavy thunderstorm	t5.png	ID:312 Heavy intensity drizzle rain	r3.png
ID:230 Thunderstorm with light drizzle	t1.png	ID:313 Shower rain and drizzle	r2.png
ID:231 Thunderstorm with drizzle	t1.png	ID:314 Heavy shower rain and drizzle	r3.png
ID:231 Thunderstorm with heavy drizzle	t1.png	ID:321 Shower drizzle	r2.png

FIGURE 25. WEATHER ICONS DESIGN 1



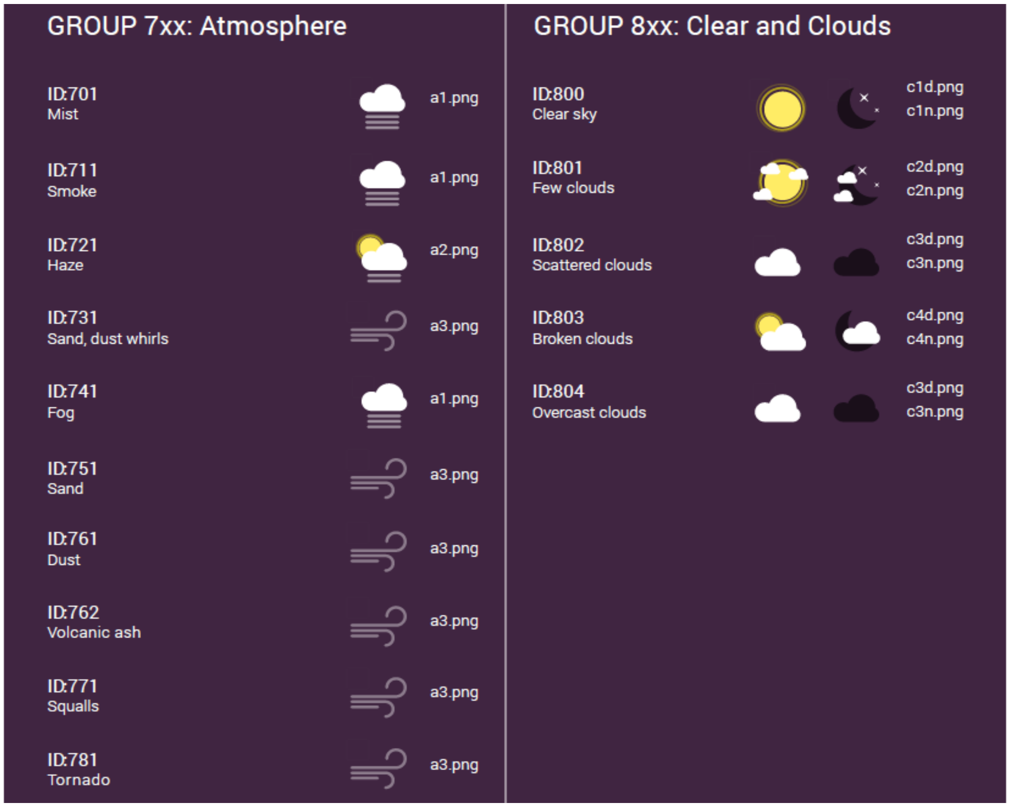
GROUP 5xx: Rain

ID:500 Light rain	r1.png
ID:501 Moderate rain	r2.png
ID:502 Heavy intensity rain	r3.png
ID:503 Very heavy rain	r3.png
ID:504 Extreme rain	r3.png
ID:511 Freezing rain	r2.png
ID:520 Light intensity shower rain	r1.png
ID:521 Shower rain	r2.png
ID:522 Heavy intensity shower rain	r3.png
ID:531 Ragged shower rain	r3.png

GROUP 6xx: Snow

ID:600 Light snow	s1.png
ID:601 Snow	s2.png
ID:602 Heavy snow	s2.png
ID:611 Sleet	s3.png
ID:612 Shower sleet	s3.png
ID:615 Light rain and snow	s3.png
ID:616 Rain and snow	s3.png
ID:620 Light shower snow	s1.png
ID:621 Shower snow	s2.png
ID:622 Heavy shower snow	s2.png

FIGURE 26. WEATHER ICONS DESIGN 2



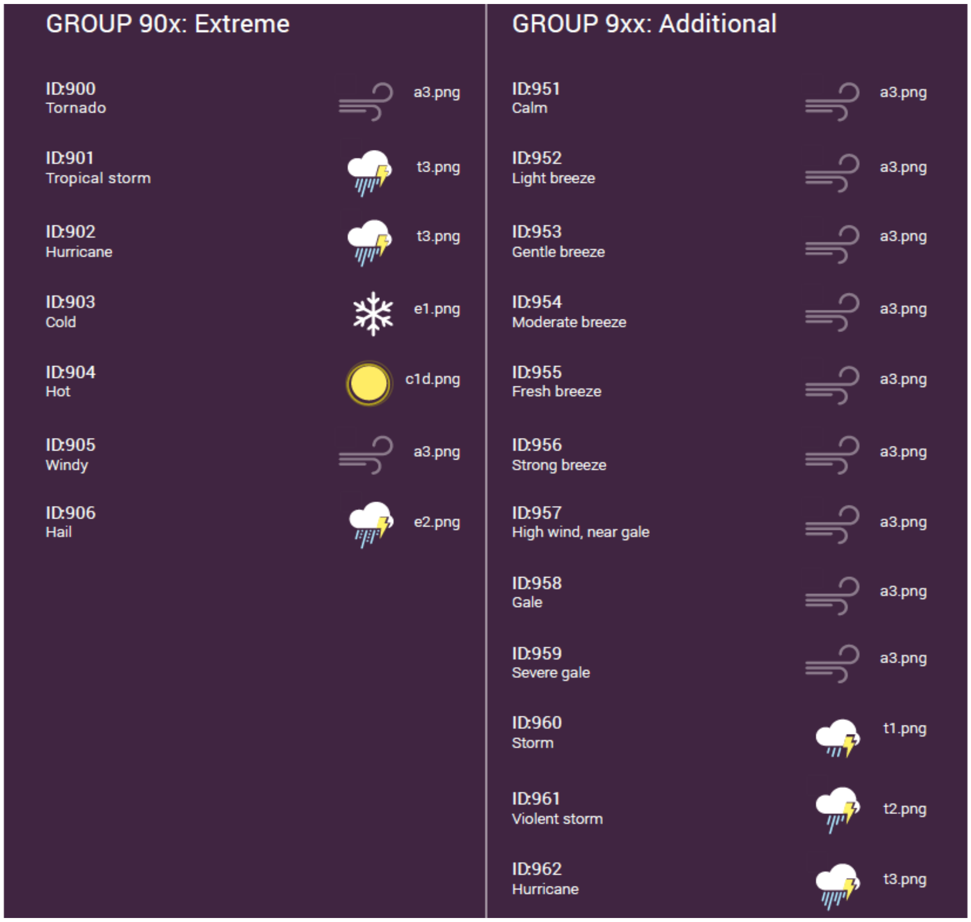
GROUP 7xx: Atmosphere

ID:701 Mist	a1.png
ID:711 Smoke	a1.png
ID:721 Haze	a2.png
ID:731 Sand, dust whirls	a3.png
ID:741 Fog	a1.png
ID:751 Sand	a3.png
ID:761 Dust	a3.png
ID:762 Volcanic ash	a3.png
ID:771 Squalls	a3.png
ID:781 Tornado	a3.png

GROUP 8xx: Clear and Clouds

ID:800 Clear sky	c1d.png c1n.png
ID:801 Few clouds	c2d.png c2n.png
ID:802 Scattered clouds	c3d.png c3n.png
ID:803 Broken clouds	c4d.png c4n.png
ID:804 Overcast clouds	c3d.png c3n.png

FIGURE 27. WEATHER ICONS DESIGN 3



GROUP 90x: Extreme

ID:900	Tornado	a3.png
ID:901	Tropical storm	t3.png
ID:902	Hurricane	t3.png
ID:903	Cold	e1.png
ID:904	Hot	c1d.png
ID:905	Windy	a3.png
ID:906	Hail	e2.png

GROUP 9xx: Additional

ID:951	Calm	a3.png
ID:952	Light breeze	a3.png
ID:953	Gentle breeze	a3.png
ID:954	Moderate breeze	a3.png
ID:955	Fresh breeze	a3.png
ID:956	Strong breeze	a3.png
ID:957	High wind, near gale	a3.png
ID:958	Gale	a3.png
ID:959	Severe gale	a3.png
ID:960	Storm	t1.png
ID:961	Violent storm	t2.png
ID:962	Hurricane	t3.png

FIGURE 28. WEATHER ICONS DESIGN 4

6.2.2. News and Events

- **Category selecting**

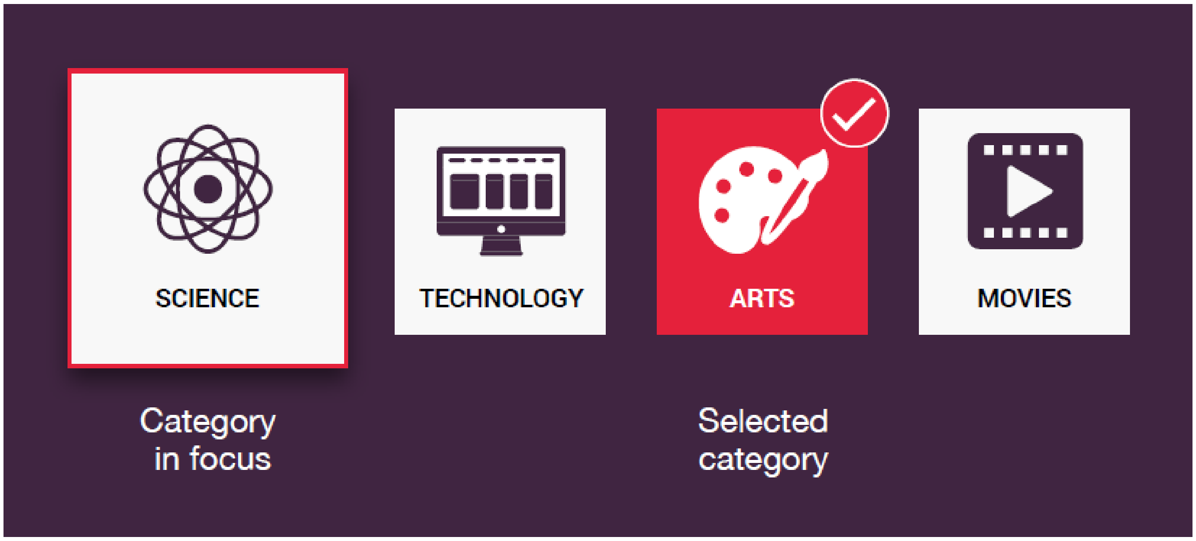
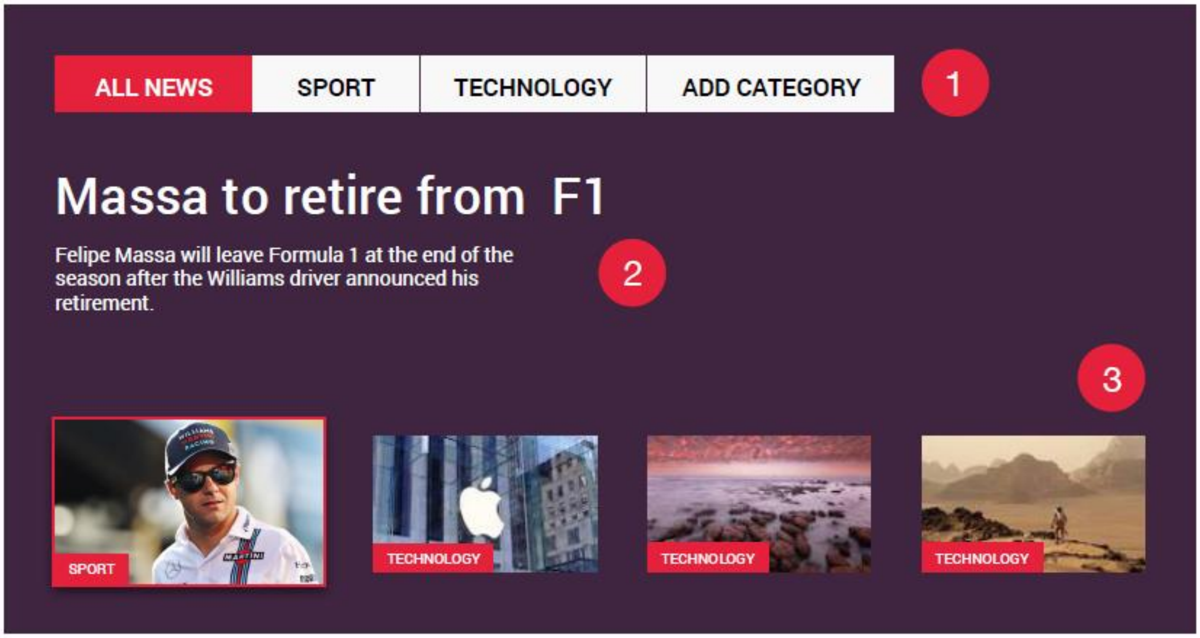


FIGURE 29. NEWS AND EVENT APPS DESIGN 1

- **Browsing News**



Massa to retire from F1

Felipe Massa will leave Formula 1 at the end of the season after the Williams driver announced his retirement.

FIGURE 30. NEWS AND EVENT APPS DESIGN 2

1. **Navigation**: All selected news categories are here. Last button is to add or remove category. So, you go back to first screen where you are choosing categories.
2. **News content**: Place for the content of the selected news. It includes headline and short description.
3. **Available news list**: List of all news. Selected news is larger, with border and shadow. You can go to next or previous news with movement to left or right on TV controller. It includes news image and category.

7. Next steps

This section includes an overview of the main technical issues that will analyze in the following versions of SENIOR-TV platform.

7.1. System Architecture

The current proposal for the Architecture of SENIOR-TV:

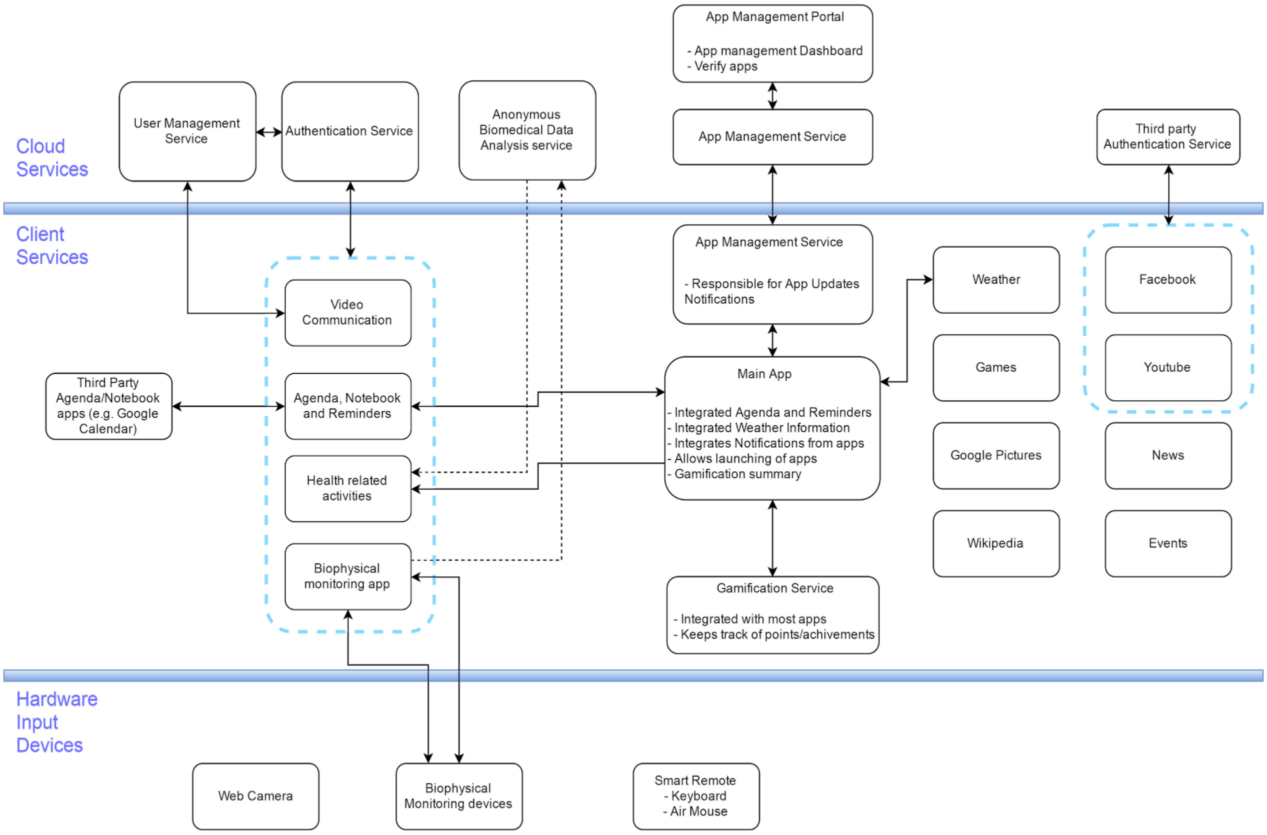

App Management Portal
- App management Dashboard
- Verify apps
Cloud Services
User Management Service
Authentication Service
Anonymous Biomedical Data Analysis service
App Management Service
Third party Authentication Service
Client Services
App Management Service
- Responsible for App Updates Notifications
Weather
Facebook
Video Communication
Games
Youtube
Third Party Agenda/Notebook apps (e.g. Google Calendar)
Agenda, Notebook and Reminders
Main App
- Integrated Agenda and Reminders
- Integrated Weather Information
- Integrates Notifications from apps
- Allows launching of apps
- Gamification summary
Google Pictures
News
Health related activities
Wikipedia
Events
Biophysical monitoring app
Gamification Service
- Integrated with most apps
- Keeps track of points/achivements
Hardware Input Devices
Web Camera
Biophysical Monitoring devices
Smart Remote
- Keyboard
- Air Mouse

FIGURE 31. SYSTEM ARCHITECTURE

The Architecture of the system will be further refined and expanded with each advance of the implementation of each app, service and component.

7.2. Common Components

During recent months, the SENIOR-TV technical team identified possible common client and cloud components and standards.

Cloud core components and standards:

- Login and user management module.
- Data persistence - Multimedia Content module.
- Communication (web - services / third parties) standard.
- Security (encryption) standards.

Client core components and standards:

- Authentication/ Login module
- Internationalization module
- Communication (Between Apps) Standard
- Graphical Components (Angular Material 2) module
- Security encryption standard
- External peripherals interface module
- Local data storage module
- Notification module

In next steps of the development work, we will further investigate these common components and standards in order to enable easy interaction between the various parts of the system.

7.3. Integration of other platforms

The next phase of the project will involve adapting the current and future apps to various other platforms: phones and tablets.

Thanks to the choice of the cross-platform technology (Apache Cordova), the adaptation should be straightforward. The issues we foresee are regarding various screen sizes, user interface interaction and navigation, native components and specific restrictions and limitations for each operating system.

7.4. Next formal and informal services to be developed

The current services that were mentioned in the DOW are regarding:

- Social (Facebook, Twitter).
- Multimedia (Youtube).
- Tele-rehab (games, audiovisual repositories).
- Activity control services (calendars, agenda, notifications, and alerts).
- Smart leisure. Personal recommendations for leisure activities.

- Information systems for the caregivers to support and manage their activities.

The proposal for these services will be presented to users during the next Workshop in order to get an assessment from their point of view.

At the same time the SENIOR-TV team identified the following trends in apps for Seniors, which we will take into consideration when starting the analysis and design for future services:

- Keep their minds active and engaged.
- Stay healthy doing physical exercise and controlling the diet.
- Stay connected with friends and family.
- Keep abreast of news and world events.
- Set reminders for physician appointments and to take medicines.
- Pay bills online.
- Get quick, to-the-point medical information.
- Listen to their favourite music.
- Stay well read.
- Easily take and save notes.
- Be entertained.

7.5. Control & Input devices

For formal and informal services that are designed for TV, the main interface between the user and the TV is a Remote Control.

During the first pilot testing we discovered that users get sometimes confused by too many buttons or buttons that are not marked correctly.

For this issues, the SENIOR-TV team is now investigating a special design for the remote control or an adaptation of the existing remote control and also the possibility of using other control interfaces for the apps (e.g. voice).

For the formal services, we will be investigating the possibility of using a bracelet for monitoring the physiological signals.

8. Conclusions

One of the main challenges to develop smart-TV applications is the variety of operating systems presents in the market. Each manufacturer uses its own technologies, so it becomes essential focusing our efforts on a subset of systems or using hybrid development technologies. Although in this first version of the platform, we have only used Android devices for testing the apps, we have decided use hybrid technologies, for reduce the implementation effort if we will decide to adapt the apps for other operating systems on the next platform versions.

During this first year, only the initial version of the most demanded informal services by the elderly have been developed. In general, the functionalities and the objectives of the apps was easily understood by the seniors but they require some usability improvements. Although they were incomplete apps, we have realized the significance of adapting the interfaces to the needs of elderly people as much as possible. For example, using concise contents, simple and large components, big fonts or neutral colors.

We could appreciate that it is fundamental studying about new ways for navigation between interface elements and new methods for interact with the platform. The buttons of traditional remote controls or the pointer are not easy to use by the elderly. Therefore, it will be necessary to make progress in the design of adapted remote controls or study the use of secondary devices, such as smartphones, to interact with the TV.

In the next platform versions, we will need to advance in the development of apps which promote the activity and autonomy of the elderly, making them attractive for seniors. In addition, it is essential to include preliminary versions of formal services, to get the elderly feedback about this kind of applications which could be more complex than the informal ones.

aal-2014-171

SENIOR-TV

PROVIDING ICT-BASED FORMAL AND INFORMAL CARE AT HOME

Quality Checklist
Quality Control of D2.1

Peer Reviewer	
Reviewer	**Partner**
Luis Anido Rifón	UVIGO (subcontracted by IMATIA)

CRITERIA	VERIFIED
1) Conformity to Standards and Project templates	
Logos (AAL, SENIOR-TV)	**Y**
Project title, reference, author, version, revision, data	**Y**
Mandatory statements (disclaimer)	**Y**
Conformance to the standard structure required by EACEA (ex. Disclaimer, Executive summary, Acknowledgement, Introduction, page numbers, etc.)	**N**
2) Language check (typing mistakes, grammar, etc.)	**Y**
3) Coherence with objectives declared in the Technical Annex	

Obj. 1: To elaborate the project's Quality Plan following well-accepted methodologies tailored to the learning domain and based on a detailed description of projects objectives, success indicators and work plan.	**Y**
Obj. 2: To monitor all project activities and provide quality control of all project results as well as recommendations for improvements and identification of best practices.	**Y**
4) Reliability of data	
Information and sources well identified	**N**
Data and information are free from factual or logic errors	**N/A**
The analysis (if applicable) is reliable, i.e. previous studies have been sufficiently reviewed; qualitative information and quantitative data are balanced and appropriate	**Y**
5) Credibility of findings	
Findings supported by evidence based on data analysis	**Y**
Replicability of findings	**N/A**
6) Validity of conclusions	
Conclusions meet evaluation questions and information needs	**N**
Conclusions supported by proper evaluation findings	**N**
No conclusions missing according to the evidences presented	**N**
7) Please indicate any deviations from contractual conditions (WP objectives declared in the technical annex)	
8) Comments/Suggestions for revision **Mandatory**: - An executive summary is missing - A section with conclusions is required - A better contextualization of the WP objectives for the first period would be required in the introduction - Sections 3.2, 3.3 and 3.4 are not SmartTV specific. Move this info to annexes. - There is insufficient justification on the technology selection. There some vague mentions to Android TV but no strong argument is provided to support this choice. Task 2.1.1 on technology analysis should have produced such conclusions. One of the core objectives	

for the first year of the project is the choice of the technology to support SeniorTV, therefore this should be strengthened in the deliverable

- There is no link with WP1 outputs. Some informal apps have been developed but no information is provided regarding how the users influenced their design or even the decision to develop these and not different ones.
- From the information provided on formal services it is not clear whether or not the applications have been actually developed. T2.3.1 clearly states that some services should have been created during the first year. This needs clarification.
- Section 7 is not fully elaborated. My conclusion is that the final architecture has not been closed yet. A justification is required.
- No references are provided.

Optional:

- There is no data about the current status and market penetration of commercial Smart TVs. Some paragraphs may be added at the beginning of the section on Technology Analysis.
- I would include the analysis of the programming paradigms in this field (section 3.1) before the introduction to the Operating Systems.
- I would add a short sub-section within section 2 including the introduction to the development frameworks (Cordova, Ionic and Unity)

9) Implementation of revisions/modifications suggested and explanation for eventual rejections (performed by the Responsible of the Deliverable)

10) Deliverable accepted

□ **YES**
□ NO

If NO, please state reasons:

Implement the mandatory suggestions above aal-2014-171

SENIOR-TV

PROVIDING ICT-BASED FORMAL AND INFORMAL CARE AT HOME

Quality Checklist
Quality Control of D2.1

Peer Reviewer	
Reviewer	**Partner**
Vojko Strojnik	RC-IKTS

CRITERIA	VERIFIED
1) Conformity to Standards and Project templates	
Logos (AAL, SENIOR-TV)	**Y**
Project title, reference, author, version, revision, data	**Y**
Mandatory statements (disclaimer)	**Y**
Conformance to the standard structure required by EACEA (ex. Disclaimer, Executive summary, Acknowledgement, Introduction, page numbers, etc.)	**Y**
2) Language check (typing mistakes, grammar, etc.)	**Y**
3) Coherence with objectives declared in the Technical Annex	

Obj. 1: To elaborate the project's Quality Plan following well-accepted methodologies tailored to the learning domain and based on a detailed description of projects objectives, success indicators and work plan.	**Y?**
Obj. 2: To monitor all project activities and provide quality control of all project results as well as recommendations for improvements and identification of best practices.	**Y**
4) Reliability of data	
Information and sources well identified	**Y?**
Data and information are free from factual or logic errors	**Y**
The analysis (if applicable) is reliable, i.e. previous studies have been sufficiently reviewed; qualitative information and quantitative data are balanced and appropriate	**Y**
5) Credibility of findings	
Findings supported by evidence based on data analysis	**Y**
Replicability of findings	**Y**
6) Validity of conclusions	
Conclusions meet evaluation questions and information needs	**Y**
Conclusions supported by proper evaluation findings	**Y**
No conclusions missing according to the evidences presented	**Y**

7) Please indicate any deviations from contractual conditions (WP objectives declared in the technical annex)

All significant issues are included.

8) Comments/Suggestions for revision

1. Adding new applications and services to the Senior-TV: Senior-TV as a platform between end users and providers. What about providers? The description is for end users only.
2. Remote control device: it presents an important feature but get rather small attention, especially, because it has been shown as a possible problem. Not only the buttons were the problem but also the aiming itself. Aiming with simultaneous pressing the button is even more demanding action. No references/features and criteria provided as in Technology analysis.
3. Add a new chapter: Conclusions. Some conclusions have been drawn in the individual chapters.

9) Implementation of revisions/modifications suggested and explanation for eventual rejections (performed by the Responsible of the Deliverable)

10) Deliverable accepted

□ YES

□ NO

If NO, please state reasons: