Mestrado
Gestão de Informação
Master in Information Management

ISEGI-NOVA AR Project
Augmented Reality applied to the Universidade Nova de Lisboa

Pedro Miguel Vieira Cardoso

Project work presented as a partial requirement to obtain a Master Degree in Information Management
ISEGI-NOVA AR PROJECT - AUGMENTED REALITY APPLIED TO
THE UNIVERSIDADE NOVA DE LISBOA

by

Pedro Miguel Vieira Cardoso

Project work presented as a partial requirement to obtain a Master Degree in Information Management, Specialising in Information Systems and Technologies Management

Adviser: Miguel de Castro Neto, Ph.D.

November 2012
ACKNOWLEDGMENTS

After more than one year of lonesome hard work and discovery, I am extremely happy to deliver a finished project. The route that brought me to the end result, narrated herein, was shared by some people I wish to acknowledge and thank:

Firstly, to this project’s adviser, Professor Doutor Miguel de Castro Neto, for his ultimate trust in me, for his immense enthusiasm and for the permanent availability.

Secondly, to Catarina, my life companion of several years, for going through the whole experience with me, and for giving me love, space and an economist’s insight.

Thirdly, to my family and friends, they keep “augmenting” as I get older, which is definitely a good sign.

Finally, to Rui Marques (thanks for the tip) and to Rex (you will not be forgotten).
ABSTRACT

In recent years, information technologies have progressed at an accelerated pace. The need to combine the availability of the information with portability has been fostering the evolution of technological devices, while producing some paradigm shifts. Nowadays, mobile computing, in particular, is an area increasingly put to use and recognised as necessary in terms of adding value to an information society. On that subject, the role of the more advanced generation of mobile phones, also called smartphones, has grown in terms of empowerment and importance. Evolving technologically day by day, smartphones can be presented as a communication tool that combines access to a key set of information and location services with pocket-like portability.

At the same time, Augmented Reality – or the act of superimposing virtual elements to the image captured in real time – has been taking advantage of the rapid evolution in terms of the smartphones’ processing capabilities. The result has translated into the development of several technologies which make full use of wireless networks, of the capabilities for detecting positioning, and of the motion sensors. The current state of maturity of these technologies allows us to take Augmented Reality into account, not only as an innovative communication form for the mobile context, but also as a sophisticated method for exhibiting information while relating the presented object, through geo-referencing, with the real world.

The goal of the ISEGI-NOVA AR Project consists in developing and deploying an Augmented Reality system that will assure a multitude of services, directed to the smartphone medium. These services will allow users to interact with multimedia layers (images, sounds, video, text, animations) superimposed, through geo-referencing, to the image captured by the smartphone camera. Besides the ability to provide Augmented Reality contents, another project goal is to develop a web platform that allows dynamic Augmented Reality content creation, through graphical user interfaces. The report includes a technology selection phase, architecture overviews, and the description of all of the developed software components.
The outcome of the ISEGI-NOVA AR Project will deliver new tools to the Universidade Nova de Lisboa, to be used for information publishing and will represent, ultimately, a new era in terms of the relationship between the university and the community.

KEYWORDS

Augmented Reality; Geo-referenced Data; Smartphone
RESUMO

Nos últimos anos, as tecnologias de informação têm progredido de forma acelerada. A necessidade de conjugar o acesso à informação com a portabilidade tem incentivado a constante evolução dos dispositivos tecnológicos e originado algumas mudanças de paradigma. Na época em que vivemos, a computação móvel, em particular, é uma área cada vez mais aproveitada e reconhecida como necessária para a criação de valor numa sociedade de informação. Nesse plano, o papel da geração mais avançada de telemóveis, denominados de smartphones, tem sido de crescente protagonismo e importância. Cada vez mais apetrechados tecnologicamente, os smartphones apresentam-se como uma ferramenta de comunicação que permite aliar o acesso a um conjunto chave de serviços de informação e localização à portabilidade de bolso.

Paralelamente, a Realidade Aumentada - ou o acto de sobrepor elementos virtuais à imagem captada em tempo real - tem aproveitado a rápida evolução da capacidade de processamento dos smartphones. O resultado traduziu-se no desenvolvimento de várias tecnologias que tiram partido das redes sem fios, das capacidades de detecção de posicionamento e dos sensores de movimento. O estado de maturidade actual destas tecnologias permite que a Realidade Aumentada seja encarada não só como uma inovadora forma de comunicação para o contexto móvel, mas também como uma sofisticada metodologia de exibição de informação que relaciona intimamente o objecto apresentado, através da georreferenciação, com o mundo real.

O objectivo do ISEGI-NOVA AR Project consiste em desenvolver e instalar um sistema de Realidade Aumentada com capacidade de assegurar um sem número de serviços para serem usufruídos por intermédio de smartphones. Estes serviços permitirão aos utilizadores interagir com sobreposições multimédia (imagens, vídeo, texto, som e animações), usando georreferenciação, e que serão visíveis através da câmara do smartphone. Para além da capacidade de emitir conteúdos de realidade aumentada, o projecto pretende desenvolver uma plataforma web que permita criar conteúdos de realidade aumentada, de forma dinâmica, através de interfaces gráficos. Este relatório inclui as fases de escolha das tecnologias, abordagem de arquitecturas e descrição de todos os componentes de software desenvolvidos.
O resultado do ISEGI-NOVA AR Project dotará a Universidade Nova de Lisboa de novas ferramentas ao serviço da divulgação da informação e permitirá, por fim, o desenvolvimento de novas formas de relacionamento entre a universidade e a comunidade.

PALAVRAS-CHAVE

Realidade Aumentada; Dados georreferenciados; Smartphone
INDEX

1. INTRODUCTION ....................................................................................................... 21
   1.1. OBJECTIVES AND PROJECT DELIMITATION .................................................... 21
   1.2. MOTIVATIONS AND PROJECT CONTRIBUTIONS ............................................ 21
   1.3. CONTEXTUALIZATION AND STATE OF THE ART ............................................. 22
      1.3.1. The emergence of smartphones ............................................................. 22
      1.3.2. Augmented Reality over the years .......................................................... 23
      1.3.3. Augmented Reality meets the smartphones .......................................... 25

2. CHOOSING TECHNOLOGIES FOR THE AR PROJECT ................................................. 29
   2.1. SELECTING THE AR BROWSER ....................................................................... 30
      2.1.1. AR browsers versus our generic requirements ....................................... 31
      2.1.2. The AR browser for the project ............................................................... 38
   2.2. THE DEVELOPMENT ENVIRONMENT FOR THE AR PROJECT ...................... 38

3. THE LAYAR AR ENVIRONMENT ............................................................................... 41
   3.1. THE LAYAR AR ARCHITECTURE ...................................................................... 41
   3.2. THE LAYAR AR BROWSER ............................................................................. 42

4. THE ISEGI-NOVA AR PROJECT ............................................................................... 47
   4.1. THE AR ARCHITECTURE ................................................................................. 47
   4.2. THE AR WEB SERVICE ................................................................................... 48
   4.3. THE AR LIBS .................................................................................................. 48
      4.3.1. The AR request ........................................................................................ 49
      4.3.2. The AR Response ..................................................................................... 52
      4.3.3. The AR Libs architecture and general overview ....................................... 58
   4.4. THE AR DATABASE AND LOCAL DATA STORAGE .......................................... 64
   4.5. THE AR MANAGEMENT SITE .......................................................................... 68
      4.5.1. AR Management Site: Architecture and concepts ..................................... 68
      4.5.2. AR Management Site: Basic site presentation ......................................... 71
      4.5.3. Layer AR resources ................................................................................. 76
      4.5.4. Hotspots AR resources .......................................................................... 80
      4.5.5. Actions AR resources ............................................................................. 88
      4.5.6. Animations AR resources ....................................................................... 93
INDEX OF FIGURES

Figure 1 - IBM Simon (AugmentedRealityWiki 2011a) ................................................... 23
Figure 2 - Sensorama by Heilig (AugmentedRealityWiki 2011)................................. 24
Figure 3 - Head-mounted display, Sutherland (1968) ..................................................... 24
Figure 4 - Reality-Virtuality (RV) Continuum (Milgram and Kishino 1994)............... 25
Figure 5 – Geo-referenced and superimposed content to a live camera view of the smartphone ............................................................................................................. 26
Figure 6 - Augmented Reality example ........................................................................ 27
Figure 7 - AR generic representation ............................................................................ 32
Figure 8 - AR generic representation - a POI link association ...................................... 32
Figure 9 - AR generic representation – image recognition with 3D model association. 35
Figure 10 - AR generic representation – a language selector ..................................... 36
Figure 11 - AR generic representation – a typology selector ....................................... 37
Figure 12 - AR generic representation - visual filters .................................................... 37
Figure 13 – Layar Platform Overview ........................................................................... 41
Figure 14 – Simulated Layar browser camera view (with legends)............................. 43
Figure 15 - Non-standard POI representation types .................................................... 44
Figure 16 - POI transformation example ....................................................................... 45
Figure 17 - An example of Layar filters ....................................................................... 45
Figure 18 - Layar Vision simulation - example ............................................................... 46
Figure 19 – High level architecture for the ISEGI-NOVA AR Project’s deliverables ...... 47
Figure 20 - AR Web Service Architecture .................................................................... 48
Figure 21 - AR HTTP Request sample extract .............................................................. 49
Figure 22 - JSON structure example ............................................................................ 52
Figure 23 - The AR Response structure (API v6.2) ...................................................... 55
Figure 24 - AR Libs architecture schema ..................................................................... 58
Figure 25 - Basic AR Classes ....................................................................................... 59
Figure 26 - Non-mandatory JSON variable with default value – example ...................... 60
Figure 27 - AR constants defined as resources in the AR Libs ....................................... 61
Figure 28 - AR Libs main JSON class structure (excluding basic AR classes) – composition .................................................................................................................. 62
Figure 29 - High-level sequence diagram for a generic AR request ............................. 63
Figure 30 - AR Database schema ................................................................................ 67
Figure 31 - AR Management Site architecture ............................................................... 69
Figure 32 - AR Management Site structure .................................................................. 69
Figure 33 - AR Management Site layout ......................................................................... 71
Figure 34 - AR site: Home page ....................................................................................... 72
Figure 35 - AR site: navigation area ................................................................................ 72
Figure 36 - Example of additional element info, input validation and error messages . 73
Figure 37 – User authentication in the AR Management Site ........................................ 75
Figure 38 - Composition diagram for AR Webpages and main user controls .............. 75
Figure 39 - Layers page – “list mode” entry screen....................................................... 76
Figure 40 - Layers page- layer “edit screen” ................................................................. 77
Figure 41 - Layer resource managing scenarios (scenario 1- actions) ......................... 79
Figure 42 - Layer resource managing scenarios (scenario 2 - animations) ................. 79
Figure 43 - Layer resource managing scenarios (scenario 3 - hotspots) ....................... 80
Figure 44 – Hotspot resources “list mode” managing scenarios .................................. 81
Figure 45 – Hotspot resources – Basic settings scenario ............................................ 82
Figure 46 - Hotspot resources - Brief information widget scenario ............................. 83
Figure 47 - Hotspot resources - Hotspot indicator scenario ......................................... 83
Figure 48 - Hotspot resources - Layar Vision POI basic settings ................................. 84
Figure 49 - Hotspot resources - Layar Vision POI Vision content settings ................. 84
Figure 50 - Actions resources “list mode” managing scenarios (scenario 1) ............... 88
Figure 51 - Actions resources “list mode” managing scenarios (scenario 2) ............... 88
Figure 52 - Actions resources “list mode” managing scenarios (scenario 3) ............... 89
Figure 53 - Action resources – “edit screen” scenarios ............................................... 90
Figure 54 – Help panel that displays the images associated with each BIW activity type ................................................................................................................................ 92
Figure 55 - Animation resources in “list mode” ............................................................ 93
Figure 56 - Animation “adding” scenarios ................................................................. 94
Figure 57 - Animations "edit screen" scenario ............................................................ 95
Figure 58 - Icon/Biw image resources in "list-mode" scenarios .................................. 98
Figure 59 - Biw Image/Icons "edit screen" scenario ................................................... 99
Figure 60 - Object main resources in "list mode" ....................................................... 101
Figure 61 - Object main resources "edit screen" ....................................................... 102
Figure 62 - Creating an AR world – example ............................................................. 106
Figure 63 - Layer creation are in the Layar publishing site ......................................... 107
Figure 64 – The geo-referenced layer’s filter properties ........................................... 108
INDEX OF TABLES

Table 1 - Generic requirements for the AR project ........................................................ 29
Table 2 - AR Browsers’ compliance with requirement 3 ................................................ 33
Table 3 - AR Browsers’ compliance with requirement 4 ................................................ 34
Table 4 - AR browsers’ compliance with requirement 5 ................................................ 34
Table 5 - AR browsers’ compliance with requirement 6 ................................................ 35
Table 6 - AR browsers’ compliance with requirement 8 ................................................ 37
Table 7 - Compliance level between the AR browsers and the requirements ............... 38
Table 8 - Relevant hardware and software used in the ISEGI-NOVA AR Project .......... 39
Table 9 - Layar request parameters .............................................................................. 51
Table 10 - Supported JSON response elements (API v6.2) ......................................... 56
Table 11 – Layer “list mode” elements description ....................................................... 76
Table 12 – Layer edit/submit screen elements description and JSON association ....... 77
Table 13 – Layer edit/submit screen input elements with applicable validation constraints .............................................................................................................. 78
Table 14 - Layer resource managing scenarios description ...................................
Table 15 - Hotspots “list mode” elements description ................................................. 81
Table 16 - Hotspot edit/submit screen (all scenarios) elements description and JSON association .............................................................................................................. 85
Table 17 – Hotspot edit/submit screen input elements with applicable validation constraints .............................................................................................................. 87
Table 18 - Action "list mode" elements description ....................................................... 89
Table 19 - Actions edit/submit screen (all scenarios) elements description and JSON association .............................................................................................................. 91
Table 20 - Actions edit/submit screen input elements with applicable validation constraints .............................................................................................................. 92
Table 21 - Animation "list mode" element description .................................................. 93
Table 22 - Animation "adding" scenarios description .................................................... 94
Table 23 - Animations edit/submit screen elements description and JSON association 96
Table 24 – Animations edit/submit screen input elements with applicable validation constraints .............................................................................................................. 96
Table 25 - Biw Image/icons "list mode" element description ....................................... 98
Table 26 - BIW Image/Icons edit/submit screen elements description and JSON association ................................................................. 100
Table 27 – BIW image /Icons edit/submit screen input elements with applicable validation constraints .............................................................................................................. 100
Table 28 - Object resources "list mode" element description .............................................. 101
Table 29 - Objects edit/submit screen elements description and JSON association ... 102
Table 30 - Objects edit/submit screen input elements with applicable validation constraints ................................................................. 103
## ACRONYMS AND ABBREVIATIONS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3G</td>
<td>3rd Generation Mobile Communications</td>
</tr>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>AR</td>
<td>Augmented Reality</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BIW</td>
<td>Brief Information Widget</td>
</tr>
<tr>
<td>CIW</td>
<td>Custom POI Indicator Widget</td>
</tr>
<tr>
<td>DBMS</td>
<td>Database Management System</td>
</tr>
<tr>
<td>GLONASS</td>
<td>Global Navigation Satellite System</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
</tr>
<tr>
<td>POI</td>
<td>Point Of Interest</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform Resource Identifier</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>Wireless Fidelity</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

1.1. OBJECTIVES AND PROJECT DELIMITATION

The mission of this project is to develop an Augmented Reality (AR) system which will provide AR content for the smartphone medium. The AR features, to be made available, will allow a user, equipped with a smartphone and the adequate technology, to move within the university campus, point the camera to geo-referenced areas and interact with the superimposed information (images, video, text, sound, and animations), using the phone screen.

The report includes an AR technology selection process which will focus on analysing some of the available AR browsers for smartphones, while considering some basic requirements which the AR system, that we propose to develop, shall meet. The Augmented Reality system, to be created, shall be prepared to act as an interface for the request/response nature of the AR browsers. Additionally, the Augmented Reality system will include means to enable the creation of AR content using a graphical user interface (GUI), thus allowing quicker content creation without requiring knowledge in any programming language.

The Augmented Reality project intends, through demonstrative AR worlds, to define the guidelines in terms of AR content creation. We expect to do so by providing clear examples of the benefits of using Augmented Reality. Ultimately, we propose to create a new communication medium between the university and the community, and to enable the Universidade Nova de Lisboa to develop new strategies to present itself to society.

1.2. MOTIVATIONS AND PROJECT CONTRIBUTIONS

Universities are the cornerstone for achieving excellence in education and should represent a vivid example of modernity. Furthermore, they are a natural platform for presenting new possibilities to society. The project that we propose, in this report, moves and evolves with these ideas in mind.

The Augmented Reality project, applied to the Universidade Nova de Lisboa, derives from a strong motivation, on our behalf, to give something back to the university that may be considered noble and innovative. The idea of being involved in a project of this nature, with such unexplored technologies and concepts, is very appealing. Moreover, given the multidisciplinary and comprehensive nature of the tasks that characterize
the life cycle of the Augmented Reality project. In this regard, the project represents several stimulating challenges:

- a creativity and sensitivity challenge to design relevant and innovative requirements;
- the technical challenge of transforming the requirements into an effective AR management system that is able to produce visually sophisticated and interactive contents, that establish a natural harmony between real and virtual environments;
- the social challenge of using smartphone-directed Augmented Reality as a mean of surprising and reaching out to people, while improving the relation between the university and the community;
- and the challenge of pioneering as far as allowing Universidade Nova de Lisboa to become one of the first universities in Portugal to release contents in an AR platform;

1.3. CONTEXTUALIZATION AND STATE OF THE ART

1.3.1. The emergence of smartphones

The value which information has in today’s society has been translated into a growing number of technologies, communication means and platforms for obtaining and sharing information. In recent years, internet access has evolved from a desktop-based home usage scenario, to a briefcase-transportable mobile laptop environment and, more recently, into a pocket size experience, thanks to the development of the smartphones.

The technology apparatus that is nowadays considered as the first smartphone to be sold, was released in 1993 by the name of IBM Simon (Wikipedia 2011d). IBM Simon (Figure 1) featured several innovative characteristics for its time, like the memory of 1 megabyte and the black and white touch screen, besides accumulating numerous other features (additional to the ability to perform phone calls) like pager functions, calculator, address book, email and fax capabilities.
After this first step, as the price of technology decreased, the concept of “smartphone” has been progressively enriched with other attributes over time. Although there is no exact definition for smartphone, mostly due to the rapid technological changes in this context, according to several sources from Wikipedia (2011e), a smartphone can be described as a communicating device with advanced processing skills, equipped with a video camera and means to access internet, namely 3G technologies (Wikipedia 2011a) and Wi-Fi (Wi-Fi-Alliance 2011). A smartphone device also incorporates motion sensors (Wikipedia 2011b) and a GPS navigation system (Wikipedia 2011c).

These characteristics make mobile phones (or smartphones) a privileged access point for information, and like a 2010 report from the International Data Corporation points out “vendors shipped a total of 302.6 million smartphones worldwide, up 74.4% from the 173.5 million smartphones shipped in 2009” (IDC 2011), smartphones represent a significant share in the global market.

1.3.2. Augmented Reality over the years

The term “Augmented Reality” was initially suggested by Caudell and Mizell (1992) who described it as the act of combining computer generated elements with the real world. However, the first experiences with “augmented” realities date back to 1962, when Morton Heilig recorded the patent of a machine, called Sensorama (Figure 2), which had the ability to reproduce movies while adding wind, vibration, smell and three dimensional images.
Some years later, Sutherland (1968) would invent a special helmet (Figure 3) which he described as being a window into a virtual world.

The helmet had the capability to reflect the users’ movement in the images reproduced, contributing highly to increase the human perception of motion within a virtual environment. This helmet system, closer in concept to what Augmented Reality offers nowadays, would be subject to imitation and further exploring in the nineties by the scientific community, for instance, to act as support tool in the process of learning architecture (Feiner et al. 1995), to help in medical surgery (Wagner et al. 1996), to interact with the real world (Feiner et al. 1997), (Thomas et al. 1998), and to help the manufacturing industry (Chung et al. 1999).

When we talk about virtual elements we can be led to some associated terms like Virtual Reality or Mixed Reality, but it was Milgram and Kishino (1994) who developed
a spectrum called Reality-Virtuality Continuum (Figure 4) which facilitates the understanding of these concepts. The Reality-Virtuality Continuum illustrates the intensity of the combination between virtual and real environments, organizing the Mixed Reality categories by weighing their level of combination.

Figure 4 - Reality-Virtuality (RV) Continuum (Milgram and Kishino 1994)

A few years later, in his research about Augmented Reality, Azuma (1997) defined three major characteristics about AR:

- the mix of the real with the virtual;
- real-time interaction;
- and 3D scenes;

These characteristics, pointed out by Azuma, can certainly be applied to most positions of the Reality-Virtuality Continuum (Figure 4) without losing their validity. It is, however, important to mention that the project we propose approaches the Augmented Reality perspective, meaning that the emphasis is on the real world first, followed by the added virtual elements, second.

1.3.3. Augmented Reality meets the smartphones

Augmented Reality can significantly flourish within the mobility context. In recent years, Augmented Reality investigations took full advantage of the technological evolution happening in the mobility area, like the mobile phone incorporation of motion sensors or the global positioning system (GPS). The results of these investigations were released in the most diverse contexts and, while not being directed specifically for smartphone devices, some proved worthy candidates for usage within the mobile context, for instance:

- games – the user plays using the real world as a background (Thomas et al. 2000), (Cheok et al. 2004);
- exhibitions and futuristic art – virtual art exhibitions where virtual artworks are applied to the exhibition space (Wojciechowski et al. 2004);
- virtual guides for museums – users obtain information about the museums’ points of interest (POI) (Miyashita et al. 2008);
Chapter 1 - Introduction

- instant translators – to instantaneously translate a live-filmed text into several languages (Haritaoglu 2001);
- mapping through recognition – mapping virtual textures in a real environment using predefined recognition patterns (Rekimoto 1998), (Kato and Billinghurst 1999).

The examples referred above were supported, most of the times, by frameworks and hardware developed specifically for scientific purposes, and inaccessible to the general public. However, between 2008 and 2009, new platforms and paradigms emerged to propel AR development in smartphones like Layar (2011), Junaio (2011) and Wikitude (2011a). All of these companies embraced a new concept which consisted in an Augmented Reality browser with a number of features that allowed developers to produce Augmented Reality content according to a specific set of rules, and, finally, allowed end-users to view computer generated elements (audio, video, images and animations) superimposed to the live camera view of common smartphones (Figure 5). These AR browsers are compatible with most mobile operating systems like the Android (2011), the iPhone OS by Apple (2011), or the Symbian (2011).

![Figure 5 – Geo-referenced and superimposed content to a live camera view of the smartphone](image)

The technologies mentioned above are represented officially through Youtube channels - the layarmobile (2011) channel for Layar, the metaioAR (2011) channel for Junaio and the self-titled Wikitude (2011b) channel – which are used to promote and publish news about the technologies, while presenting some examples of their portfolio. The published content spans from the advertisement area to the creation of city/building navigation systems, all based in interactive play.

To provide a basic example of what an AR reality experience may look like in a smartphone context, we will demonstrate a hypothetic scenario in Figure 6. We start,
however, by introducing the term “point of interest” (Wikipedia 2012c) or POI, which is a nomenclature widely used in cartography to define a certain position, when found to be of interest, roughly defined in a map or corresponding to specific coordinates.

As displayed in Figure 6, the basis for an AR interaction is the definition of a POI, geo-referenced using GPS coordinates, to which a certain behaviour and visual characteristics may be associated. The user may be notified, either in a visual or sound form, when POIs are near. Menus can help in the navigation or configuration process and may be accessed through the smartphone touch screen.

Smartphones’ interaction potential, combined with AR technologies, represents the starting point for our AR project that is to be applied to the Universidade Nova de Lisboa. We are focused on creating a new world of mixed realities where access to information and the possibilities for exploring and socialising in the university campus can be enhanced. We wish to take the opportunity to open a new communication medium between the university and the community and to take a chance at becoming one of the first AR-enabled universities in Portugal.

The next section will focus on analysing and selecting technologies and hardware for the AR project, including a few tests with the Junaio, Layar and Wikitude AR browsers.
2. CHOOSING TECHNOLOGIES FOR THE AR PROJECT

Before starting the actual development of our AR system we undertook the task of selecting a set of technologies that could provide a balance between number of features, quality, stability, and ease-of-use characteristics, in order to maximise the development speed. The choosing process, in spite of a deliberate intention to avoid subjectivity, was understandably influenced by our programming experience and/or by financial or time convenience, and is hereby presented with an accompanying justification for all options. It should be noted that the AR project involves not just AR technologies but also web and database technologies.

At this stage we needed to know what technologies to search for, so, a few preliminary guidelines were defined to provide generic indications of what the AR project should be. These are presented in Table 1 and explained in the following sections.

<table>
<thead>
<tr>
<th>Id</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The AR project shall provide a web service so that Augmented Reality content can be experienced using a smartphone across the Universidade Nova de Lisboa campus area</td>
</tr>
<tr>
<td>2</td>
<td>The AR project shall provide an AR Management tool to allow content managers to create, update and delete AR content using a web interface</td>
</tr>
<tr>
<td>3</td>
<td>The selected AR browser technology shall support audio, video, image, 3D elements and animations</td>
</tr>
<tr>
<td>4</td>
<td>The selected AR browser technology shall be compatible with Android and iPhone OS operating systems</td>
</tr>
<tr>
<td>5</td>
<td>The selected AR browser shall be able to function inside buildings</td>
</tr>
<tr>
<td>6</td>
<td>The selected AR browser technology shall be able to recognise image patterns and to map digital content to the recognised patterns</td>
</tr>
<tr>
<td>7</td>
<td>The selected AR browser technology shall be free for the end-user</td>
</tr>
<tr>
<td>8</td>
<td>The AR content made available in the smartphone environment shall be filterable and shall behave accordingly in the browser view by the following contexts: language, typology of POIs, and visual characteristics</td>
</tr>
</tbody>
</table>

We classify the first two guideline items, from Table 1, as being part of our very basic generic AR project proposal, but as we move from items 3 to 8, they are all related with the AR browser technology to be selected. These items will be better understood in the next sections.
2.1. SELECTING THE AR BROWSER

Before selecting an AR browser technology, we have to decide the type of implementation we are looking for the AR project. Presently, multiple solutions exist in the form of a Software Development Kit (SDK) or of an Application Programming Interface (API).

The SDK solutions are libraries organised in a way so that software engineers develop their own mobile applications and, by including the SDK sources, are able to embed Augmented Reality features. The SDK option implies developing and compiling an AR application into several different platforms (iOS, Android, Symbian,...) and then releasing it, after going through the required external (i.e. Apple Store, Android Market, etc.) approval stages, into each operating system’s application provider store. Also, with the SDK solution the developer may have to handle compatibility issues related with variable smartphones’ hardware.

The other considered option is the API-based AR technology, which creates a higher tier that provides an interface between an API-prepared AR browser and the smartphones. Simply put, the API is prepared to send AR content requests and receive AR content responses. Therefore, by opting for an API-based solution, software engineers accept to interact with the smartphones using the API’s lexicon and grammar as intermediary, instead of developing their own AR browser applications. Since the AR browsers are developed to be compatible with several mobile operating systems, the multi-platform and hardware compatibility issues are dealt by the AR browser provider.

From a hardware compatibility point of view and an overall public reach perspective, we believe that the API-based AR browser solution is more adequate for our project since it:

- Aggregates the entities (we intend to be one) that publish AR content and make them accessible through a common public interface. This means that a user searching for Augmented Reality experiences will find our AR contents indexed in a public search engine provided by each AR browser manufacturer;
- Displays harmonised graphical behaviour (i.e. the interface is similar in all smartphones), regardless of the platform and hardware;
- Is extensively tested for bugs and frequently updated on different operating systems and hardware.
For our AR project we, therefore, choose the API-Based AR technology and, in the next section, we advance to the AR browser selection process by performing basic tests using our generic guidelines from Table 1.

2.1.1. AR browsers versus our generic requirements

The process of selecting an AR browser for our project had no intention to compare the AR browsers in a way to decide which one is the best in overall terms or to produce any sort of wide recommendation for a technology. More importantly, time limitations restrict our ability to go through all existing AR browsers. This means that the technology selection process intends to minimise the effort necessary to reach an acceptable verdict. That effort is, nonetheless, related with how aligned the AR browsers are with our guidelines (Table 1) and with our ability to develop within a certain environment.

Using a high-end Android OS smartphone, kindly lent for our project by the university, we performed a mobile web research with the keyword “Augmented Reality browser” - visually filtering instantly recognisable software that would not comply with our project - at the Android Market (2011) on the 27th of December, where we collected download statistics for Junaio (>100 000 downloads), Layar (> 10 000 000 downloads) and Wikitude (>1 000 000 downloads). While these statistics provided some degree of information about the browsers’ popularity, we found an additional element in the form of a comparative study of these browsers by Madden (2011), which acknowledges them as being the most popular and technologically evolved. Considering this comparative analysis, our time limitations, and given the features descriptions we were able to find for each browser, on the Android Market, we acknowledged all technologies as being suitable for our selection process.

The next step was to install and analyse each browser against our AR project requirements (Table 1, items 3 to 8). To conduct the tests, performed during January of 2012, we used a Samsung Galaxy SII smartphone with an Android 2.3.3 operating system and installed the Junaio v3.1, the Layar v6.0 and the Wikitude v6.4 browsers.

After an initial acquaintance phase with the browsers, a scenario based on a fictitious garden was developed, so that our basic concepts (Table 1), associated with AR content representation, could be tested. Figure 7 illustrates a generic live camera view of the smartphone looking at an AR world. The blue dots represent geo-referenced POIs which can be associated with information or may trigger events.
Requirement 3: The selected AR browser technology shall support audio, video, pictures, 3D elements and animations

This requirement was defined so that the user could benefit from varied experiences that could be associated to a geo-referenced POI. The act of associating media content to a POI can be done if the AR Browsers are prepared to support events/actions that can be launched upon user interaction with the POIs. For instance, after a user clicks on the table POI (Figure 7) blue dot, the browser could display a set of actions that could lead to consequences like playing an audio file of relaxing music or opening a web page about table shopping (example in Figure 8).

The following table presents some details about each browser’s compliance.
Chapter 2 - Choosing technologies for the AR project

Table 2 - AR Browsers’ compliance with requirement 3

<table>
<thead>
<tr>
<th>Browser</th>
<th>Behaviour</th>
</tr>
</thead>
</table>
| **Junaio** | The Junaio browser allows associating several elements, through GUI buttons, as events to a POI:  
- Audio: mp3 format;  
- Image: png, jpeg formats;  
- Video: mp4 format;  
- 3D elements: supports *md2* and *obj* 3D models  
- Animations: 2D/3D element animation is supported |
| **Layar** | The Layar browser allows associating several elements, through GUI buttons, as actions to a POI:  
- Audio: mpeg, mp4, mp3  
- Video: 3gp, mp4  
- Image: web based support through an html link  
- 3D elements: supports *l3d* models which can be converted from *obj* or *mtl* models through the use of an available conversion tool  
- Animations: supports 2D/3D animation |
| **Wikitude** | The Wikitude browser allows associating attachments to a POI. This implies downloading a file and opening it with available smartphone software. No 3D native support. |

In this regard, the Wikitude browser posed some limitations in terms of visually associating an action/event different than an attachment or linking to a website. The Junaio and Layar browsers offer sufficient support.

**Requirement 4: The selected AR browser technology shall be compatible with Android and iPhone OS operating systems**

This requirement was defined in order to cover the two major smartphone operating systems. The selection between the Android and the iPhone operating systems was based on an expected behaviour on behalf of the smartphone market that may or may not happen in the near future.

In the United States of America, Nielsen Wire (2011) reports an Android and iPhone OS dominance for the 3rd quarter of 2011, while in Europe, comScore (2011) reports that, on March of 2011, Symbian is considerably ahead of iPhone OS and Android. However, as a consequence of a recent change of strategy from Nokia (2011), the mobile manufacturer responsible for the current Symbian OS legacy, the Symbian OS will be dropped as primary operating system in favour of a Microsoft Windows based platform. Ultimately, since we are choosing AR Browsers which are constantly being updated, any changes in the dominant smartphone operating system scenario will,
predictably, be accompanied by new AR browser releases to assure compatibility with new mediums.

The following table presents some details about each browser’s compliance with the requirement.

<table>
<thead>
<tr>
<th>Browser</th>
<th>Behaviour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Junaio</td>
<td>Support for iPhone OS (iPhone, iPad, iPod Touch) and Android</td>
</tr>
<tr>
<td>Layar</td>
<td>Support for iPhone OS (iPhone 3GS, 4, 4S), Android, Symbian, Blackberry</td>
</tr>
<tr>
<td>Wikitude</td>
<td>Support for iPhone OS, Android, Symbian, Blackberry</td>
</tr>
</tbody>
</table>

All browsers comply with our mobile operating system requirements.

**Requirement 5: The selected AR browser shall be able to function inside buildings**

This requirement was defined so that the user could be able to experience AR content inside buildings. The AR browsers depend heavily on GPS reception and precision, which is normally affected inside buildings, hence the necessity to present some alternatives for displaying AR content indoors with acceptable precision.

<table>
<thead>
<tr>
<th>Browser</th>
<th>Behaviour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Junaio</td>
<td>The browser offers:</td>
</tr>
<tr>
<td></td>
<td>- Junaio Glue technology which allows mapping image/objects to a specific image pattern. This feature may be used to provide indoor content.</td>
</tr>
<tr>
<td></td>
<td>- Junaio LLA Markers technology which can ignore the GPS settings indoor and place the POIs relative to a physical pattern placed on a certain location</td>
</tr>
<tr>
<td>Layar</td>
<td>The browser offers Layar Vision technology which allows mapping image/objects to a specific image pattern. This feature may be used to provide indoor content.</td>
</tr>
<tr>
<td>Wikitude</td>
<td>No support available. The browser always relies on GPS reception.</td>
</tr>
</tbody>
</table>

The Junaio and Layar browsers are the only ones compliant with this requirement, with Junaio offering more possibilities to tackle with poor GPS reception.

**Requirement 6: The selected AR browser technology shall be able to recognise image patterns and to map digital content to the recognised patterns**
Recognising patterns is a feature that may benefit our AR project in terms of variety of presentation and interaction, since it allows relating AR content to a recognised element. When a user points the smartphone camera to a certain image, stored previously as a recognisable element, and the AR browser recognises the predefined pattern, it can trigger the presentation of an Augmented Reality experience. For instance, if the AR Browsers recognised the “wanted” sign on the top of the tree, a 3D model of the suspect could be displayed (example in Figure 9) or the act of interacting with the poster sign could trigger a phone call to the police.

![Figure 9 - AR generic representation – image recognition with 3D model association](image)

**Table 5 - AR browsers' compliance with requirement 6**

<table>
<thead>
<tr>
<th>Browser</th>
<th>Behaviour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Junaio</td>
<td>This browser offers the Junaio Glue technology which allows mapping 2D images and 3D objects to a specific 2D image pattern.</td>
</tr>
<tr>
<td>Layar</td>
<td>This browser offers Layar Vision technology which allows mapping 2D image and 3D objects to a specific 2D image pattern</td>
</tr>
<tr>
<td>Wikitude</td>
<td>No support available</td>
</tr>
</tbody>
</table>

The Junaio and Layar browsers are compliant with our requirement, while the tested Wikitude version provides no support for such features.

**Requirement 7: The selected AR browser technology shall be free for the end-user**

All AR browsers – Junaio, Layar and Wikitude – are free to the end user and downloadable at the app stores associated with each mobile operating system.
Requirement 8: The AR content made available in the smartphone environment shall be filterable and shall behave accordingly in the browser view by the following contexts: language, typology of POIs, and visual characteristics.

This requirement defines important characteristics which can be applied to the AR content. In order to allow a more accurate AR navigation experience, we predicted the utility of filtering features:

- **language**: the ability to support contents in multiple languages;
- **typology of the point of interest**: the ability to filter by content similarity (i.e. distinguish buildings, lamp posts, gardens...);
- **visual characteristics**: the ability to present different icons/images/objects to represent the geo-referenced point;

To be able to display content in different languages is very important, moreover in the context of a university that has a tradition of receiving numerous foreign students, namely the ones that enrol in the Erasmus programme. Figure 10 displays an example of what such capabilities could be like.

![Figure 10 - AR generic representation – a language selector](image)

The ability to filter by typologies of the points of interest is a feature intended to enhance the user experience by providing single or combined types of content filters (classroom, libraries, etc...) which will influence the quantity and the typology of content displayed in the camera. Figure 11 displays an example of a possible typology filter implementation to refine an AR content search result.
Additionally to the user configurable filters, our AR project would benefit from POIs that could be visually distinguishable from one another, like an apple tree being identified by an apple tree icon or a garden bench by the corresponding icon. Figure 12 displays the icon example, which serves as visual aid, but the same example could be applied to the representation of 3D models (requirement 3 in Table 1), instead of icons, applied to the POIs.

Table 6 - AR browsers’ compliance with requirement 8

<table>
<thead>
<tr>
<th>Browser</th>
<th>Behaviour</th>
</tr>
</thead>
</table>
| Junaio  | - Configurable filters (language/typology): It’s possible to add “search box” filtering but we could not easily configure the language and content type filters in our testing phase  
- Visual filters: the browser displays the same icon for all POIs but allows associating different images, inside the POI icon, which allows the intended differentiation |
### Chapter 2 - Choosing technologies for the AR project

#### 2.1.2. The AR browser for the project

As mentioned in section 2.1.1, we did not intend to evaluate which of the AR technologies is the best in the market. Due to time constraints, neither an extreme feature analysis was performed, nor were any stress/performance tests conducted involving the technologies, different hardware or comprehensive usage scenarios.

Table 7 displays the compliance level (None, Partial, Full) between the tested AR browsers and our generic requirements previously defined in Table 1 (requirement 3 through 8).

<table>
<thead>
<tr>
<th>Requirement Id</th>
<th>Junoio v3.1</th>
<th>Layar v6.0</th>
<th>Wikitude v6.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Full</td>
<td>Full</td>
<td>Partial</td>
</tr>
<tr>
<td>4</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>5</td>
<td>Full</td>
<td>Full</td>
<td>None</td>
</tr>
<tr>
<td>6</td>
<td>Full</td>
<td>Full</td>
<td>None</td>
</tr>
<tr>
<td>7</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>8</td>
<td>Partial</td>
<td>Full</td>
<td>Partial</td>
</tr>
</tbody>
</table>

Based on our evaluation of the technologies and the analysis of results from Table 7, we decided to use the Layar browser as our AR technology.

#### 2.2. The development environment for the AR project

Having selected the Layar AR browser, the next step was to prepare a development environment. The Layar AR browser, as detailed in chapter 0, requires a server to handle the request/response nature of dynamic AR content presentation. Universidade Nova de Lisboa provided a Microsoft Windows server to the project, and
considering our relative experience with Microsoft technologies, we opted for Microsoft Visual Studio 2010 as an integrated development environment (IDE) and Microsoft SQL Server 2008 R2 as a database management system (DBMS).

The list of relevant hardware and software, to be used for the ISEGI-NOVA AR Project’s development and testing phases, is presented below.

Table 8 - Relevant hardware and software used in the ISEGI-NOVA AR Project

<table>
<thead>
<tr>
<th>Context</th>
<th>Type</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Hardware</td>
<td>64 bit Intel Xeon CPU at 2.4GHz with 4GB of RAM</td>
</tr>
<tr>
<td>Server</td>
<td>Software</td>
<td>Windows 2008 Server R2 Standard</td>
</tr>
<tr>
<td>Server</td>
<td>Software</td>
<td>Microsoft SQL Server 2008 R2</td>
</tr>
<tr>
<td>Development Environment</td>
<td>Software</td>
<td>Windows 7 Professional</td>
</tr>
<tr>
<td>Development Environment</td>
<td>Software</td>
<td>Microsoft SQL Server 2008 R2 Developer</td>
</tr>
<tr>
<td>Development Environment</td>
<td>Software</td>
<td>Microsoft Visual Studio 2010 Premium</td>
</tr>
<tr>
<td>“High-end” smartphone</td>
<td>Hardware</td>
<td>Samsung Galaxy SII</td>
</tr>
<tr>
<td>“High-end” smartphone</td>
<td>Software</td>
<td>Android 4.0.3</td>
</tr>
<tr>
<td>“High-end” smartphone</td>
<td>Software</td>
<td>Layar 6.0 (Upgraded to 6.2 during project)</td>
</tr>
<tr>
<td>“High-end” smartphone</td>
<td>Hardware</td>
<td>iPhone 4S</td>
</tr>
<tr>
<td>“High-end” smartphone</td>
<td>Operating System</td>
<td>iOS 6.0.1</td>
</tr>
<tr>
<td>“High-end” smartphone</td>
<td>Layar Software</td>
<td>Layar 6.0 (Upgraded to 6.2 during project)</td>
</tr>
<tr>
<td>“Low-end” smartphone</td>
<td>Hardware</td>
<td>Samsung Gio</td>
</tr>
<tr>
<td>“Low-end” smartphone</td>
<td>Operating System</td>
<td>Android 2.3.6</td>
</tr>
<tr>
<td>“Low-end” smartphone</td>
<td>Layar Software</td>
<td>Layar 6.0 (Upgraded to 6.2 during project)</td>
</tr>
</tbody>
</table>
3. THE LAYAR AR ENVIRONMENT

Before we approach all of the developed components for the AR project, we use this chapter to explain how the Layar “ecosystem” is organised and what lies between the smartphone and the AR experience. The description of the Layar AR browser is also included, with supporting illustrations and examples to provide, hopefully, a detailed enough introduction to the browser’s features.

3.1. THE LAYAR AR ARCHITECTURE

The AR project was developed in compliance with Layar’s architecture requirements. Figure 13 displays Layar’s architecture based on the representation available in the Layar site (Layar 2012g). The AR system architecture is based on a multi-server environment on which a part of a request is handled on Layar’s infrastructure, and the other part is handled by the AR service provider (in this case, on our side).

In order to have publicly accessible AR content, the first step is for the content provider to register on the Layar publishing site (Layar.com, item 1 from Figure 13) and define several characteristics which will identify an “AR layer”. An AR layer acts as a container for the points of interest which, ultimately, represent the Augmented Reality experience, made available for the users.

Figure 13 –Layar Platform Overview

A large part of the characteristics defined in Layar’s publishing site are for administrative and legal purposes, although this is where some of the look and feel of the AR browser menus, and also some information filters will be defined (refer to section 3.2). After the AR layer is approved by the Layar team, it becomes publicly
visible to all end-users and eligible to have its content created, updated or deleted by
the administrator. Finally, for a user to be able to access an AR layer, all that’s required
is to have the Layar AR browser installed in a smartphone (iPhone or Android) with
internet access, GPS and motion sensors features.

To explain how the AR content is brought to the smartphone we will use Figure 13 as
an example:

a) The user opens the Layar application on the smartphone, finds our content
layer through the application’s search tool, and opens our AR world. This initial
connection, which retrieves existing layers, is made between the smartphone
(item 3, Figure 13) and the Layar Publishing Site (item 1, Figure 13);

b) Once the world is opened, the smartphone Layar API performs content
requests to the Layar server, forwarded via proxy (item 2, Figure 13). These
requests are redirected to the AR provider’s (we are the provider, in this case)
web service (item 4, Figure 13). The request contains information like the layer
ID, the GPS coordinates of the smartphone which performed the request,
among other attributes;

c) Once the AR provider’s web service receives a request, it prepares an answer in
a JavaScript Object Notation (JSON) format (to be explained in section 4.3.2)
which is returned to the Layar server which, in turn, forwards the response to
the smartphone’s AR browser. The AR provider’s web service response can be
either fixed (hardcoded) for all requests, or it can be variable in nature, in
which case it is recommended to implement an AR content database repository
(item 5, Figure 13) to feed the web service;

d) After the AR browser API on the smartphone receives the JSON response, it can
translate the structured language into visual information, and the user can
finally engage in the Augmented Reality experience. Since the JSON response
may contain portions that point to external links (images, video, sounds,
webpages), for performance purposes, the AR browser will access these links
directly, without using the Layar proxy as intermediary.

3.2. THE LAYAR AR BROWSER

The Layar AR browser is a very important part of the Augmented Reality experience. It
represents the access point to the AR worlds for an end-user. In this section we intend
to provide a simplified overview of the Layar AR browser. All features which concern
the JSON message are approached in more detail in section 4.3.
Figure 14 displays a simulation of several of the browser’s features on screen, approximately as they are experienced in the live camera view of the smartphone. The attributes with the “LPUB” prefix are defined in the Layar publishing site which allows several look-and-feel settings of the browser’s menus and the POIs to be changed. The attributes included with the “JSON” prefix are the ones received in the JSON message, sent by the service provider AR Web Service.

The inner popup associated to the selected POI in Figure 14 - containing the title, description, footnote, image, and actions - is hosted in a logic area, defined in the Layar terminology as the Brief Information Widget (BIW). The BIW contains the information associated with the POI as well as the actions (Layar 2012a) which the POI can trigger.

When a user points to a POI, besides the possibility of highlighting the POI icon with a different colour, it may also open the BIW (if such behaviour is defined in the JSON message) area and display the associated information.

The actions displayed in Figure 14 (“More info” and “Gallery”), at the bottom of the BIW area, can result in activities like opening web pages, videos, and audio files. The action button also supports the inclusion of an icon (the Layar system has a list of...
available icons), placed alongside the action’s name, indicated via JSON message, which suggests the button’s content.

Besides the standard POI look (focus, outer, inner, and background colours) shown in Figure 14, a POI can be represented in the live camera view, like exemplified in Figure 15, by:

- A Custom POI Indicator Widget (CIW) defined in the Layar publishing site by uploading images corresponding to inner, middle, outer, and focus areas (item 1 in Figure 15). The request to use a previously defined CIW set needs to be requested in the JSON message. The CIW option displays behaviour identical to the standard POI, where the POI indicator is highlighted upon focus;
- An icon image (item 2 in Figure 15), where an icon representation URL is sent via JSON message. No inner, middle, outer and focus characteristics are defined. Different icons can be defined for different POIs;
- A 2D image object (item 3 in Figure 15) with a real-world size sent in the JSON message. An object element (2D or 3D) features a real-world size indication in meters which will influence the POI size according to the distance relative to the smartphone (closer equals a bigger object, farther equals smaller object or even no display at all);
- Or a 3D model object with a real-world size (item 4 in Figure 15)

The POI representation also allows the incorporation of animations (Layar 2012b), either previously defined basic API animations (drop, grow or spin effects), or custom animations where the content provider can specify numerous parameters (via JSON) to create more complex animations. This means that, for instance, a POI representation can spin during a number of seconds, rise from the ground, or jump up and down.
Associating an animation to a POI can be done at a layer-level, which defines the same animation for all POIs, or can be set at a POI level, which defines (or overrides layer-level definitions) animation characteristics individually.

Transformations like rotations, scaling, or translations (Layar 2012f) can influence the initial starting position of an object which represents a POI. This means that, for instance, we could rotate a 3D model POI representation, like displayed in Figure 16.

Another important Layar feature that affects the options available in the AR browser is the content filtering capability. This allows the content provider to define the filtering criteria which will be applied to the POIs, like shown in Figure 17. The filters are customised in the Layar publishing site, obey to a set of constraints (refer to section 4.3.1), and require a specifically prepared service provider logic (mentioned in section 4.3) to produce the desired effect. The content filters are accessible to the user through the properties of the selected AR world layer.

Figure 16 - POI transformation example

Figure 17 - An example of Layar filters
Finally, we will approach the image recognition feature, entitled Layar Vision, which is prepared to recognise a previously stored image and, by consequence, to represent associated information or actions. To make use of the image recognition features, the image to be recognised must be added to the Layar server using the Layar publishing site, where the content provider can then set a unique identifier (name) to the image. The image, however, must obey to a set of visual requirements (Layar 2012h), defined to facilitate the recognition process. Figure 18 shows an example of a user navigating through an AR layer, enabled for Layar Vision POIs, and the smartphone’s AR browser camera is recognising a poster and displaying the associated AR content.

Figure 18 - Layar Vision simulation - example

All of the Layar features referred in the section will be better understood as the AR components, developed for our AR project, are introduced in the next chapter.
4. THE ISEGI-NOVA AR PROJECT

4.1. THE AR ARCHITECTURE

Considering Layar’s architecture requirements in section 3.1, and given the generic guidelines 1 and 2, declared in Table 1 (at the beginning of chapter 1, in page 29), our AR project defined the following deliverables which are presented through a high-level architecture in Figure 19:

1. An AR Web Service prepared to handle the AR request/response nature. The AR Web Service handles the requests and relays the JSON responses, which are built using the logic contained within the AR Libs;
2. An AR Class Library (AR Libs) made available as a Dynamic-Link Library, commonly known as DLL (MSDN 2012b), to be used by the AR Web Service and the AR Management Site. The AR Libs deliverable contains logic and data tiers to provide the Augmented Reality content;
3. An AR Management Site which allows the content provider to create Augmented Reality experiences, without requiring programming skills, using a graphical interface. The AR Management Site makes use of the AR element constants and the data tier available in the AR Libs.
4. A database management system to act as repository for all AR content with the purpose of serving both the AR web service and the AR Management Site.

In the next sections we will approach each one of these four components in a more detailed way.

Figure 19 – High level architecture for the ISEGI-NOVA AR Project’s deliverables
4.2. The AR Web Service

The AR Web Service was developed with the capability to handle HTTP requests and to return a response in a JSON format. As illustrated in Figure 20, the access point (interface) for the AR Web Service is the AR HTTP Handler, which is, in fact, a generic ASP.NET Generic Handler class. This class implements the IHttpHandler interface (MSDN 2012c) and defines a default ProcessRequest method which receives the HTTP requests and, in our case, returns the JSON output to the browser responsible for the request. The mentioned ProcessRequest method receives an HttpContext class object as argument, which, in turn, encapsulates several elements associated with the request: the request parameters, the application state, and information about the session, to name a few.

When the AR HTTP Handler receives a request, the AR Web Service uses the AR Libs’ request and response classes to acknowledge all of the request’s parameters and to return a structured JSON message, respectively.

As it will be reinforced in section 4.3, the AR Libs is where all of the AR “intelligence” is implemented. The AR Web Service serves, therefore, as a mere intermediary to receive the request and to forward the response produced by the AR Libs.

4.3. The AR Libs

The AR Libs class library is one of the core components in the ISEGi-NOVA AR Project as it is responsible for producing the JSON structured message, with all the information which will translate into the Augmented Reality world. The JSON structured response is delivered to the AR browser and encompasses all that is to be seen or experienced on
the smartphone screen: the GPS coordinates, the animation effects, the actions available to be triggered through POI interaction, the rotation, translation, and scale effects, etc.

Before explaining how the AR Libs class library is organised, we’ll first provide some information about the structure of the HTTP requests and JSON responses, according to the Layar architecture.

### 4.3.1. The AR request

The HTTP request, produced by the smartphone’s AR Layar API and relayed by the Layar Server, implements the standard HTTP request format (W3C 2012) represented by:

- an HTTP address (underlined grey section in Figure 21);
- a question mark (purple section in Figure 21);
- subsequent parameters (light blue elements in Figure 21), which are separated by ampersands (orange elements in Figure 21) and use the equal sign (red elements in Figure 21) to display the value (green elements in Figure 21).

```plaintext
http://IsegArProject.com/ArIsegI.ashx?countryCode=PT&lat=52.377544&lon=4.887339&userId=ed48067cda8e43j34h393da4fd490a37&layerName=ArLayer (...)
```

**Figure 21 - AR HTTP Request sample extract**

The API documentation section on the Layar site (Layar 2012d) defines the structure for the POI request as well as the parameters used in the request. Table 9 contains all of the request parameters available for Layar API v6.0, along with a short description and other details. All parameters are acknowledged (processed) by the AR Libs, although only the following will actually be supported by our AR project implementation to define/filter the content returned in the JSON response:

- “layarName” – this parameter indicates which layer’s content is to be searched in the AR Database;
- “CUSTOM_SLIDER” – this is a parameter we will use in our AR demonstration world (refer to section 5.2.2) to define how far, in meters, we will see into the world.
- “RADIOLIST” - this is a parameter we will use in our AR demonstration world (refer to section 5.2.2) to allow the user to change the language of the displayed AR content.
The filter types available in the Layar publishing site can be configured in different ways and according to different contexts. Like mentioned in section 3.2, every filter configured in the Layar publishing site will require the creation of a business logic prepared to implement the expected filter behaviour. For instance, when a request is received, it is necessary to analyse the filter types and parameters and to transform them into a database query that returns the desired data. Therefore, the act of adding a filter (or changing the logic) in the Layar publishing site will always require a new AR Libs software release in order to deal with the rationale behind the filter. So, for the AR project described in this report, the AR Libs will only support the filters defined in the demonstrative AR implementation (refer to chapter 5).
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>userId</td>
<td>integer</td>
<td>yes</td>
<td>The unique ID of the user’s phone in a hash code format. Since this is created during the installation of the Layar browser, as long as the AR browser is not completely removed from the smartphone device, the ID will remain the same.</td>
<td>userId=13344069379</td>
</tr>
<tr>
<td>layerName</td>
<td>string</td>
<td>yes</td>
<td>The layer unique ID like defined in the Layar publishing site</td>
<td>layarId=ArLayer</td>
</tr>
<tr>
<td>version</td>
<td>string</td>
<td>yes</td>
<td>The Layer browser version installed in the smartphone</td>
<td>version=v6.0</td>
</tr>
<tr>
<td>lat</td>
<td>Decimal</td>
<td>yes</td>
<td>The latitude of the smartphone at request time</td>
<td>lat=56.3434334</td>
</tr>
<tr>
<td>lon</td>
<td>Decimal</td>
<td>yes</td>
<td>The longitude of the smartphone at request time</td>
<td>lon=23.450590</td>
</tr>
<tr>
<td>countryCode</td>
<td>string</td>
<td>yes</td>
<td>A two letter code provided by the SIM card on the phone or by the locale value set on the phone (when no SIM card information is found)</td>
<td>countryCode=PT</td>
</tr>
<tr>
<td>lang</td>
<td>string</td>
<td>yes</td>
<td>The language setting as defined on the phone</td>
<td>lang=EN</td>
</tr>
<tr>
<td>action</td>
<td>string</td>
<td>no</td>
<td>Parameter indicates whether the client is requesting a full layer refresh or just an update to be performed in the current context.</td>
<td>action=update</td>
</tr>
<tr>
<td>accuracy</td>
<td>integer</td>
<td>no</td>
<td>The accuracy in meters of the GPS coordinates as given by the smartphone device.</td>
<td>accuracy=15</td>
</tr>
<tr>
<td>pageKey</td>
<td>string</td>
<td>no</td>
<td>The Layar browser has an interface where the POIs can be seen in a list format (just a text, a small icon, and a distance value) which can spread over several pages. With this parameter the Layar Server can request the next pages.</td>
<td>pageKey=asdad232</td>
</tr>
<tr>
<td>oauth%</td>
<td>variable</td>
<td>no</td>
<td>Uses the OAuth (<a href="http://oauth.net/">http://oauth.net/</a>) protocol to enforce a secure authorization mechanism for the http requests, through the parameters: oauth_consumer_key, oauth_signature_method, oauth_timestamp, oauth_nonce, oauth_version, oauth_signature, oauth_body_hash</td>
<td>N/A</td>
</tr>
<tr>
<td>radius</td>
<td>integer</td>
<td>no</td>
<td>This value is retrieved from the default range slider filter, defined in each layer at the Layar publishing site, which gives the radius in meters on which to obtain POIs.</td>
<td>radius=50</td>
</tr>
<tr>
<td>localCountryCode</td>
<td>string</td>
<td>no</td>
<td>Value based on the country of the mobile network provider in use. If the value is not available the Layar browser will try to determine the value using GPS location (through reverse geo-coding lookup) if available.</td>
<td>localCountryCode=UK</td>
</tr>
<tr>
<td>alt</td>
<td>integer</td>
<td>no</td>
<td>The altitude (above sea level) value in meters</td>
<td>alt=20</td>
</tr>
<tr>
<td>recognisedReferenceImage</td>
<td>string</td>
<td>no</td>
<td>The ID, as defined in the Layar publishing site, of an image recognised by the smartphone. Only when working with the image recognition features</td>
<td>recognisedReferenceImage= myImage</td>
</tr>
<tr>
<td>RADIOLIST</td>
<td>string</td>
<td>no</td>
<td>A content filter parameter: radio button list value according to the filters defined in the Layar Publishing site and accessed through the Layar browser settings associated with the layer. Only one radio button list filter per layer is supported on the Layar publishing site (refer to Figure 17 in page 45 for a visual example of the filters).</td>
<td>RADIOLIST=POR</td>
</tr>
<tr>
<td>CHECKBOXLIST</td>
<td>string</td>
<td>no</td>
<td>A content filter parameter: displays the comma-separated ids of the values selected in a check box list filter defined in the Layar Publishing site and accessed through the Layar browser settings associated with the layer. Only one check box list filter per layer is supported on the Layar publishing site (refer to Figure 17, in page 45 for a visual example of the filters).</td>
<td>CHECKBOXLIST=1,2,3</td>
</tr>
<tr>
<td>SEARCHBOX, SEARCHBOX_n</td>
<td>string</td>
<td>no</td>
<td>A content filter parameter: search term entered by the user to filter content. Up to three searchboxes can be defined. The searchboxes will be numbered SEARCHBOX_N, where n is the sequential id of the searchbox as it is ordainly defined in the Layar publishing site (refer to Figure 17, in page 45 for a visual example of the filters).</td>
<td>SEARCHBOX_2= trees, SEARCHBOX_3=tables</td>
</tr>
<tr>
<td>CUSTOM_SLIDER, CUSTOM_SLIDER_n</td>
<td>string</td>
<td>no</td>
<td>A content filter parameter: a sliding range value entered by the user to filter content. Up to three custom sliders can be defined. The custom sliders will be numbered CUSTOM_SLIDER_N, where n is the sequential id of the custom slider as it is ordainly defined in the Layar publishing site (refer to Figure 17, in page 45 for a visual example of the filters).</td>
<td>CUSTOM_SLIDER_2= 10, CUSTOM_SLIDER_3=3.5</td>
</tr>
</tbody>
</table>
4.3.2. The AR Response

The JavaScript Object Notation (JSON) response output is the key element behind the Augmented Reality experience as it represents all that the user will see on the smartphone. If, for proof-of-concept purposes, we took the time to build a JSON response by hand, with the intention of making it public as quickly as possible, besides the Layar site registration steps, we would only require a simple web service to pass the hand-built JSON message to the browser requesting Augmented Reality content.

The JSON format (JSON 2012) was developed as a text language-independent-data-interchange-format with the intention of being human-readable and at the same time easily parseable and generated by computers. The JSON structure is organised by a name/value logic and also supports ordered lists (arrays) of elements.

```json
{
    "id":13,
    "name": "Jonathan Onion Potato",
    "age": 21,
    "hobbies": ["walk the dog", "practice good deeds"],
    "email": "j.o.potato@emailsnov.com",
    "eatsVegetables": false,
    "family": [
        {
            "kinship":"father",
            "name":"Edward Cabbage Potato"
        },
        {
            "kinship":"mother",
            "name":"Selma Carrot Onion"
        }
    ]
}
```

Figure 22 - JSON structure example

Figure 22 shows an example of a “person” object using the JSON notation, with the variable names highlighted in red. If we make a quick analysis of the example, it is considerably explicit that:

- the name/value pairs are separated by colons;
- the variable names and the text type values are placed within quotation marks;
- a “complex” type is written between curly brackets;
- and an array content is written within square brackets.
The documentation section in the Layar site (Layar 2012e) provides information about the expected JSON response structure supported by the Layar AR browser.

The AR Response schema (Figure 23) illustrates the JSON structure hierarchy with added legends to facilitate some distinctions. The schema presents several types of data which can be returned:

- the “simple” types - string, integer, boolean, and decimal - represented by the name/value notation;
- and the “complex” types (i.e. arrays of “simple” types, arrays of dictionary types, or just dictionary\(^1\) types).

Highlighted in the AR Response schema (Figure 23) are also:

- the mandatory elements;
- the elements which are applicable only to:
  - geo-layers: layers which contain geo-referenced POIs;
  - vision-enabled-layers: layers which contain POIs that use the Layar Vision image recognition technology;
- the items not supported in our AR system (by design).

The JSON response shall obey to a minimum required elements list in order to be considered valid. If the mandatory elements are not returned within the JSON response, the response will be discarded and no content will be displayed in the smartphone. Like exemplified in the AR Response schema (Figure 23) – for instance, the axis (x, y and z) mandatory coordinates in the animation>axis path – child items may only be mandatory once the parent node (which may or may not be mandatory) is written in the JSON response.

For our JSON response, we chose not to support a small number of Layar’s response items (highlighted as “not supported” in the AR Response schema), mainly because some of them would require complex hardcoded logic in the AR Libs class library to be compatible with our flexible AR Management Site dynamic-content-creation approach. Basically, the decision principle we applied is the same as the one we took towards the restrictions for the supported filters in the Layar Publishing site (mentioned in section 4.3.1). By using an AR Management Site to create content, we sacrifice the prospect of defining a “fixed” artificial-intelligence-fuelled AR world, in favour of the possibility of

\(^1\) A dictionary type is a data structure which encapsulates “simple” types and/or “complex” types
allowing the creation of an extensively customisable AR world, with a more linear behaviour (refer to section 4.5 for more details).

To complement the AR Response schema (Figure 23), we present the description of the supported response elements in Table 10, which also provides a closer look at the raw elements that make use of the AR browser features (section 3.2). The information in Table 10 is presented in a succinct manner, without all the details associated to each variable, which are available in the Layar site documentation area (Layar 2012e), but with enough insight to allow some perception of the features and JSON response structure. Section 3.2 can also be used as a reference to facilitate the variable’s understanding.

It should be noted that during the development of our AR project, we upgraded the Layar API version from version 6.0 to 6.2, which resulted in some AR variables and features changes. Consequently, all of the presented data, both in both Figure 23 and Table 10, reflects the response structure for version 6.2 of the Layar API.
Figure 23 - The AR Response structure (API v6.2)
### Table 10 - Supported JSON response elements (API v6.2)

<table>
<thead>
<tr>
<th>Path/parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>root.layer</td>
<td>The unique ID of the layer, as registered on the Layar's publishing site.</td>
<td>&quot;layer&quot;: &quot;ArProjectTest&quot;</td>
</tr>
<tr>
<td>root.errorCode</td>
<td>Error code equals 0 for &quot;ok&quot; and is within the 20-30 range for other errors. Range 20-29, can be used to send a message back to the AR browser. Error code 30 can be used to indicate that this layer requires authentication.</td>
<td>&quot;errorCode&quot;: 0</td>
</tr>
<tr>
<td>root.errorString</td>
<td>The message that is to be sent for the AR browser, in case of error.</td>
<td>&quot;errorString&quot;: &quot;No POIs found. Please adjust the filters&quot;</td>
</tr>
<tr>
<td>root.refreshInterval</td>
<td>The attribute tells the AR browser how many seconds to wait until a content refresh request is made.</td>
<td>&quot;refreshInterval&quot;: 100</td>
</tr>
<tr>
<td>root.refreshDistance</td>
<td>This tells the AR browser to refresh when the user is moving above the specified range in meters.</td>
<td>&quot;refreshDistance&quot;: 50</td>
</tr>
<tr>
<td>root.fullRefresh</td>
<td>Attribute tells the AR browser what type of refresh should be done: if true, the entire layer is refreshed; if false, the JSON response will just update the current POI list, meaning that POIs returned are added (for new ids) or replaced (for existing ids) POIs.</td>
<td>&quot;fullRefresh&quot;: false</td>
</tr>
<tr>
<td>root.biwStyle</td>
<td>Attribute tells the AR browser which BIW style should be used. &quot;classic&quot; style displays the entire &quot;text&quot; object when a POI is in focus while &quot;collapsed&quot; style only shows the title parameter.</td>
<td>&quot;biwStyle&quot;: &quot;classic&quot;</td>
</tr>
<tr>
<td>root.hotspots</td>
<td>An array of POIs with several location and behaviour parameters.</td>
<td></td>
</tr>
</tbody>
</table>
Axis where the animation revolves around.

The attribute which indicates the path to the action that is to be triggered.

The content type of the action can be "audio/mp3" for audio files, "video/mp4" for video files, "text/html" for web pages, and "application/vnd.layar.internal" in conjunction with "tel:", "mailto:", "sms:", "layer://" to add phone, email, SMS, and external layer opening (using an animation to change between different layers).

The request type ("GET" or "POST"). In our AR system it shall always be "GET".

Determines the icon that should be used in an action button. Layar provides a predefined set of icons for selection.

For POI actions only. If true, the Layar AR browser will close the BIW after the user clicked the button of the action.

For Vision POIs. Indicates if the action can be auto-triggered by reaching the POI. Note: not available for the layer level actions.

Indicates whether the action can be invoked manually through the use of the action button. When set to "false" it will allow manual triggering; otherwise it cannot be manually triggered.

Implements "animation" behaviour which will be associated to the POI. Three default animation types: "drop" (POI drops from the sky), "grow" (POI grows from nothing until it reaches the real size), "spin" (POI spins, slowing gradually until it stops).

Implements "animation" behaviour which will be associated to the POI. Several types of animation events can be defined:
- onCreate, onUp, onUpdate, onDelete, onFocus, onClick -> onDocumentLoad, onDocumentUnload, onDocumentVisibilityChange
- If implemented at a POI level will override settings made at a layer level actions.
- If implemented at layer level actions.

Can contain "rotate", "scale", or "translate" values which imply the type of movement around within the axis (x, y, z).

The duration in milliseconds between "from" and "to" statuses. Animation will stay at "from" status for the given delay, and only then it will move to the "to" status. The definition of elements "from" and "to" vary according to "interpolation".

Parameter used in the interpolation algorithms.

Interpolation parameter contains animation behaviour. Several types can be defined:
- 1- "linear" (constant animation rate);
- 2- "accelerateDecelerate" (animations speeds until the halfway the animation, then slows until it stops);
- 3- "accelerate" (starts slowly then it speeds up);
- 4- "decelerate" (starts quick and slows down);
- 5- "bounce" (the animation is reversed in the final state);
- 6- "cycle" (repeats for a number of times);
- 7- "anticipateOvershoot" (starts out reversed, then accelerates in the specified direction, then decelerates, overshoots the target value, reverses again and finally goes to the specified value);
- 8- "anticipate" (starts reversed, then accelerates in the specified direction towards the specified "to" value);
- 9- "overshoot" - (starts fast, overshoots the specified "to" value then reverses to reach the specified "to" value)

If set to true, the POI animation is maintained even when the POI loses focus.

Specifies if the animation is repeated in a loop.

Specifies the start state of the animation.

Specifies the end state of the animation.

Axis where the animation revolves around.
4.3.3. The AR Libs architecture and general overview

The AR Libs class library was developed using the .NET environment and C# as a programming language. Figure 24 portrays a high level architecture of the AR Libs class library, with some indication of the information flow. Briefly explained, in four steps, the AR Libs:

1. receives an HttpContext .NET object which encapsulates the request relayed by the Layar server;
2. retrieves all information from the database, concerning the request, using the logic and data tiers;
3. instantiates the AR classes in the logic tier;
4. and builds a JSON response based on the AR classes’ instantiation.

As mentioned in previous sections, the JSON message is one of most important outputs in our AR project since it will contain all of the AR experience, condensed in one message, to be interpreted by the Layar AR browser, installed in the smartphone. Regarding the JSON messages’ length and characteristics, the AR Libs obey to the following criteria:

1. For performance purposes, the JSON message shall be as short as possible:
   a. Classes and/or class variables with null values shall not be written in the response;
   b. Classes and/or class variables which contain values recognised as default (to be explained in the next paragraphs) by the Layar AR API shall not be written in the response;
2. By Layar design, the URLs included in the JSON response shall produce a text element, which escapes the forward slash (“/”) with the backslash (“\”)
character (i.e. “http://link.com/hello.jpg” shall be represented in the JSON message as “http:\/\link.com\hello.jpg”).

Even though the .NET framework provides classes designed to manipulate information in JSON format, like the `DataContractJsonSerializer` (MSDN 2012a) class, we opted for developing our own implementation of a JSON serialiser. This option helped us to tackle some limitations, imposed by the JSON serialisation requirements, to handle the complex class structure (with defaults and boolean flags) which we intended to build, with the purpose of delivering the shortest and most dynamic possible JSON message.

Figure 25 - Basic AR Classes

Figure 25 displays the basic classes which were created to encapsulate the basic types of variables used in the JSON response. The `ArBool`, `ArString`, `ArDecimal`, and `ArInt` types are classes derived from the `ArVariable` base class, each with their own serialisation method. The `ArVariable` class contains, besides the variable names, several boolean flags which will influence the JSON message generation.

These variables help in a way, such as, when a variable is requested to produce a JSON response, the JSON output will only be produced when the value is not null or when the variable’s value is different from the default value defined in the Layar API (example in Figure 26).
The variable defaults, which are part of the Layar API specifications (Layar 2012e), are embedded in the AR Libs logic resources and concern the parameters contained in Table 10. The AR Libs’ resource structure (Figure 27) is organised to provide information linked by different contexts, and separated by Layar API versions, for each AR variable, like:

- **ArElementDatabaseName** – this element defines the name (column name) for which a given AR variable is known in our AR Database;
- **ArElementDefaultValue** – this element defines the default value associated to the variable according to the Layar API specifications (Layar 2012e);
- **ArElementFrameworkName** – this element defines the name for which a given AR variable is known within the AR Libs;
- **ArElementHasDefault** – this element indicates whether or not an AR variable has a default value specified by the Layar API specifications (Layar 2012e);
- **ArElementIsMandatory** – this element indicates whether or not an AR variable is mandatory for inclusion in the JSON response message, specified by the Layar API specifications (Layar 2012e);
- **ArElementVariableName** – this element defines the JSON name of the AR variable as specified by the Layar API specifications (Layar 2012e).
The AR resources are organised into fixed hardcoded resources instead of being saved, for instance, in a configuration file, even though they may change value or logic after each Layar API release. We opted for this approach because no evident advantages were found in opting for a configuration file solution, mostly due to the fact that changing or adding an AR variable always implies updating the applicable logic for the AR Database schema, the AR Libs or the AR Management Site’s pages. So, it is defined, by design, that, for each new Layar API release, given the cascade effect on several components, all changed AR variables shall be reviewed, the impact shall be measured in all components, the necessary logic alterations shall be performed, and a new release of our AR system’s affected components shall be produced.

Figure 28 displays a class diagram (by composition) that highlights the hierarchy and structure (basic classes in Figure 25 are not included) responsible for generating a JSON message. Each class may encapsulate basic types (Figure 25) or other “complex” classes (Figure 28), where each class and variables are equipped with their own serialisation methods.
Figure 28 - AR Libs main JSON class structure (excluding basic AR classes) – composition
We can use the class diagram in Figure 28 and the high-level sequence diagram in Figure 29 to approach the instantiation process, which starts when the HTTP request is received and the \texttt{ArRequest} object is created. This is followed by the instantiation of an \texttt{ArResponse} object, which receives the \texttt{ArRequest} object as a parameter, and fetches the information from the database. After all of the information is retrieved, an instance of \texttt{ArLayer} class type is created and the previously gathered database information is sent across the JSON class hierarchy through the logic instantiation chain.

![Figure 29 - High-level sequence diagram for a generic AR request](image-url)
The process of obtaining the JSON message is done by producing a serialisation request on the ArLayer, which represents the top of the JSON classes’ hierarchy. Every JSON class will then call each of its attributes’ serialisation methods, with the result being our structured, and optimised in length, JSON message, ready to be experienced in the smartphone.

4.4. THE AR DATABASE AND LOCAL DATA STORAGE

The AR Database is where the complete Augmented Reality experiences are stored. The information is added to the AR Database by using the AR Management Site (to be detailed in section 4.5) which provides a graphical user interface that organises the main AR components into a flexible logic of content creation. Once the information is fed into the AR Database, and duly associated to a layer id defined in the Layar publishing site, the AR Web Service can then use the AR Libs logic to retrieve the database information and produce the JSON response understood by the AR browser.

Some of AR Database’s table columns hold paths (like Uniform Resource Identifier [URI]), which point to a directory structure, defined in our AR server and created to store the binary files (audio, video, image, icon, and 3D model) used in the AR worlds. The AR Management Site is responsible for maintaining the physical file structure, creating and deleting files by user request.

Figure 30 illustrates the AR Database schema where it can be observed that the poi table – the point of interest, also known as hotspot – represents the main element of the AR experience, as it is the one with the most connections with other tables. Below we present a short description of each table, which should be combined with the logic learned in sections 3.2 (AR browser), 4.3.2 (AR Response), and 4.5 (AR Management Site) to obtain a more comprehensive understanding:

1. **language** – defines the supported languages, according to the language filters defined in the Layar publishing site and the multi-language features available in the AR Management Site;

2. **transformation** – stores the transformation data which can be applied to a POI. The act of creating a table dedicated for transformations was done by design, instead of including the fields directly in the poi table (item 4), to provide future means for managing transformations independently of the points of interest;

3. **image** – stores the images visible in the brief information widget, which are managed independently from the points of interest;
4. **poi** – stores the data concerning the points of interest like the coordinates, the brief information widget behaviour, etc;

5. **icon** – stores the icons visible in the live camera view of the smartphone, which are managed independently from the points of interest;

6. **custom_animation_type** – defines the custom types of animation (rotate, translate, scale), supported by the latest Layar browser, which can be applied to an animation;

7. **interpolation_type** – defines the types of interpolations (linear, accelerate, etc...), supported by the latest Layar browser, which can be applied to an animation;

8. **multi_lang_title** – defines the actual title instance, for a POI, relating it with a specific language;

9. **title** – stores unique identifiers that point to multiple language definitions of the brief information widget’s titles which apply to the POIs (item 4);

10. **poi_type** – defines the types of POIs as far as using georeference or image recognition technology;

11. **poi_animation_event** – relates an event type with an animation and a POI, establishing the order on which the animation sequence will occur for a given stage (on create, on click, etc...);

12. **multi_lang_label** – defines the actual label instance, for each action, relating it with a specific language;

13. **description** – stores unique identifiers that point to multiple language definitions of the brief information widget’s descriptions, which apply to the POIs (item 4);

14. **object** – stores data which concerns 2D and 3D objects used to represent the point of interest indicator in the live camera view;

15. **event_type** – defines the allowed event types (on create, on click, etc.) which represent the current stage of an animation in the point of interest’s lifetime;

16. **animation** – stores the information which concerns the animations’ characteristics like duration and behaviour;

17. **multi_lang_uri** – defines the actual URI instance (a path to a resource), for each action, relating it with a specific language;

18. **multi_lang_description** – defines the actual description instance, for a POI, relating it with a specific language;

19. **layer_poi** – association table between the points of interest and the layers;

20. **label** – stores unique identifiers that point to multiple language definitions of action (item 30) button names;
21. **content_type** – defines the types of contents and their Layar terminology (e.g. image type equals to “image/vnd.layar.generic”), applicable to actions and/or objects;

22. **layer_action** – relates layers with actions;

23. **layer_animation_event** – same definition as in item 11, except the relation is set at a layer level;

24. **uri** – stores unique identifiers that point to multiple language definitions of action (item 30) paths;

25. **multi_lang_footnote** – defines the actual footnote instance, for a POI, relating it with a specific language;

26. **biw_style** – defines the types of brief information widget styles supported by the Layar browser;

27. **layer** – stores the information which concerns the Augmented Reality worlds, including the relation between a database layer and a layer defined in the Layar publishing site;

28. **simple_animation** – defines the basic system-default animations (grow, spin, etc.), natively supported by the Layar AR browser;

29. **footnote** – stores unique identifiers that point to multiple language definitions of the brief information widget’s footnotes which apply to the POIs (item 4);

30. **action** – stores all information which concerns the actions triggered by the POIs;

31. **layer_type** – defines the layer types in terms of content (e.g. 2D Layer, 2D/3D layer, and 2D/3D layer with image recognition support).

32. **activity_type** – defines the supported icon terminology id list (e.g. open, close, score, etc...) which is mapped by the Layar browser into icons presented alongside an action button;

33. **poi_action** – relates an action with a point of interest.
4.5. THE AR MANAGEMENT SITE

Even though the AR Libs component, developed within our project, is the crucial artefact behind the Augmented Reality experience we aim to provide, there is yet another very important element: the AR Management Site. The AR Management Site was built with the purpose of “boosting” the AR content creation process. Any entity with the intention of creating AR content will be able to do it in an accessible manner, without possessing any knowledge in programming languages or database structures.

The AR Management Site was developed using .NET technologies, Framework 4.0, the Ajax Control Toolkit, and C# as a programming language. The site was designed for a single-user scenario (the content provider actor), with user authentication, and within an intranet context.

Since we predicted the AR Management Site would be the most time consuming element to develop, we decided to keep the approach as simple and practical as we could. The main goal of the site was to provide simple and suggestive enough interfaces, whose ultimate interaction output would result in the adequate population of our AR Database with distinct Augmented Reality experiences.

In the next subsections we will approach the architecture and page organisation of the AR Management Site.

4.5.1. AR Management Site: Architecture and concepts

The site was designed to benefit from some of the features provided by the AR Libs component: the Data Tier which provides the SQL execution methods, and the Logic Tier which provides the AR Constants element that contains information about each AR variables’ JSON name, affected database column names, default values, mandatory flags and default flags (like explained in section 4.3.3). We also developed a “local” Logic Tier equipped with methods capable of building the database queries necessary to display information and to manage (insert, update, delete) the existing AR data accordingly, in the AR Database.
With the mission of defining the site’s structure, we performed an analysis on the JSON response parameters in Figure 23, from page 55, and classified Layer, Hotspot, Action, and Animation as our main elements. The Layer element represents the AR world which contains several elements of type Hotspot (also known as points of interest or POIs). Both the Layer and Hotspot elements can contain Action elements which, when triggered, may add more content to the AR experience. The Animation element defines animated behaviour against each Hotspot, which can be set at a Layer-level or at a Hotspot-level, as explained in the AR browser overview (section 3.2). Additionally, we’ve identified two more supporting resource groups, in the form of BIW Images/Icons and 2D/3D Objects. The BIW Images/Icons section will allow centralised management of the images used in the BIW area, as well as the custom icons used to represent a point of interest. The 2D/3D objects, which stand for real-world-size elements that represent POIs, shall also benefit from a dedicated management site area.
Using our main elements as a starting point, we defined the site structure displayed in Figure 32 which will allow users to:

- Create, edit and delete layers (AR worlds) in the Layers resource page;
- Create, edit and delete hotspots (POIs) via:
  - The Hotspots’ general resource page which shall contain all hotspots that exist in the AR Database;
  - An existing parent layer node that shall display only the hotspots associated with the selected layer;
- Create, edit and delete actions via:
  - The Actions’ general resource page which shall contain all actions that exist in the AR Database;
  - An existing parent layer node that shall display only the actions associated with the selected layer;
  - An existing parent hotspot node that shall display only the actions associated with the selected hotspot;
- Create, edit and delete animations via:
  - The Animations’ general resource page which shall contain all animations that exist in the database;
  - An existing parent layer node that shall display only the animations associated with the selected layer;
  - An existing parent hotspot node that shall display only the animations associated with the selected hotspot;
- Create, edit and delete BIW Images and POI indicator icons via:
  - The BIW Images and Icons main resource page which shall contain all the BIW images and POI indicator icons which exist in the database;
  - An existing parent hotspot node that shall display only the BIW images and Icons associated with the selected hotspot;
- Create, edit and delete objects via:
  - The Objects’ main resource page which shall contain all the objects which exist in the database;
  - An existing parent hotspot node that shall display only the objects associated with the selected hotspot.

By implementing a “resource” concept, we enforce the reuse possibilities in terms of the AR contents:

- a hotspot can be used by multiple layers;
- an action can be used by multiple layers and hotspots;
Chapter 4 - The ISEGI-NOVA AR Project

- an animation can be used by multiple layers and hotspots;
- a BIW image can be used by multiple hotspots;
- an icon can be used by multiple hotspots;
- a 2D/3D object can be used by multiple hotspots;

The AR site structure in Figure 32, as well as some of the concepts mentioned in the current section, will be further understood in the following sections when we present the overview of the site’s pages and associated features.

4.5.2. AR Management Site: Basic site presentation

In this section we will highlight the main characteristics of the AR Management Site, including the layout organisation, the most commonly used web user controls, the validation techniques, a schematic view of the main classes, and some of the look and feel aspects.

Although the developed site represents a “beta” software release, it is delivered as fully working and comprehensively tested version. A lot of priority was given to the ease-of-use dynamics which resulted in a, hopefully, simple looking site with powerful features, advanced maturity and flexibility.

The simple three-zone site layout concept in Figure 33 can be visually translated into Figure 34, which displays the home page for the AR Management Site.
Chapter 4 – The ISEGi-NOVA AR Project

Figure 34 - AR site: Home page

For the AR Management Site, we created several custom web user controls that encapsulate logic that is reused throughout the webpages. Besides the “heavier” web controls developed for the AR-related logic (layers, hotspots, actions, animations, images/icons and objects), several supporting controls were developed to provide the navigation area, to display error messages, to inform about input logic, and to perform custom validations.

Figure 35 - AR site: navigation area

The navigation area, displayed in Figure 35, exhibits the following behaviour:

- Highlights the currently active page link through a colour and design shape change;
- Highlights the page link item, through a colour and design change, when it is hovered;
- Opens the affected link page when clicked.
Figure 36 shows an example of several user controls developed to validate the graphical interfaces’ input and facilitate the displayed elements’ understanding. Since the input elements are a mix between the AR elements, necessary to produce a JSON response, and our own additional “metadata”, and that both require clarification, an ExtendedLabel control (item 1.1 in Figure 36) was created to display a dialog popup (item 1.2 in Figure 36), triggered when the user clicks the element’s text, which informs about the nature of the expected input on behalf of the user. Regarding the JSON response elements, the information provided is very similar to the Layar site’s API documentation (Layar 2012c).

The input/submit validation is performed using:

- Assertive input restrictions in the text boxes (ExtendedTextBox and MultilangExtendedTextBox controls, displayed in items 2.1 and 3.1, respectively, in Figure 36) according to:
  - Integer type which allows only numeric input;
  - Floating point type which allows only numbers and commas;
  - Text type which allows all characters except quotes;
All-or-none logic for the multiple language items (3.1 in Figure 36, when input exists for one multiple language variable, it must be done for all language input boxes);

- Informative messages and alerts, through a dynamic panel (StatusInformation control, item 2.3 from Figure 36):
  - Incorrect input values;
  - Missing input values for mandatory fields;
  - Database insertion errors;

- Upload restrictions (ResourceUpload control):
  - Extension validation;
  - Write-to-server capabilities.

The centre piece in Figure 36 shows an input validation scenario, triggered after the user presses the “save” (item 2.6) button. When warnings exist, either due to problems submitting the data to the AR Database, or due to user input errors, a warning panel (StatusInformation control, item 2.3) is displayed, with an informative message (item 2.4), and a panel “closing” button (item 2.5).

In the validation process, the site includes:
- the .NET controls of type:
  - RequiredFieldValidator - for missing values detection;
  - and CompareValidator - for the data type integrity checks;
- the Ajax Control Toolkit’s FilteredTextBoxExtender to restrict text box inputs;
- a panel, just above the input area (item 2.3 from Figure 36), to display all error/warning messages;

Regarding the AR Management Site’s security aspects, user authentication is required. This means that, whenever there is no session initiated, all pages are blocked and the web navigation is always redirected to the login page (refer to Figure 37).
Figure 37 – User authentication in the AR Management Site

Figure 38 helps to demonstrate the level of reusing of user controls in the AR Management Site, through a composition diagram of the “main” elements. On the right bottom area, the highlighted (in orange colour) “support” controls do not display the connections to the other controls in Figure 38 to avoid a number of excessive connection lines, although they are embedded throughout most of the site’s controls and pages. The act of reusing controls enforces a consistent graphical and validation behaviour across the site. The reuse advantages will become clearer in the next chapters, as we provide the resources’ overview with supporting illustrations.

Figure 38 - Composition diagram for AR Webpages and main user controls
4.5.3. Layer AR resources

Conceptually, a layer is a virtual container for all of the AR content. In the AR Management Site, the Layers resource represents the highest level in the content hierarchy. It not only allows the user to create layers, which represent the AR worlds, but it is also an entry point for subsequent content creation (hotspots, animations, actions, etc.).

![Figure 39 - Layers page – “list mode” entry screen](image)

Figure 39 displays the entry screen for the Layers page which shows the currently existing layers in “list mode”. The description of each numbered item is provided in Table 11.

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A panel which hosts the page content</td>
</tr>
<tr>
<td>2</td>
<td>A button which opens a new panel for creating a new layer</td>
</tr>
<tr>
<td>3</td>
<td>A button which allows the user to edit the settings of a layer</td>
</tr>
<tr>
<td>4</td>
<td>A button which allows the user to delete a layer. This action does not delete any animations,</td>
</tr>
<tr>
<td></td>
<td>actions or hotspots resources because these are all independent and can exist without a layer</td>
</tr>
<tr>
<td>5</td>
<td>The header section which refers to the available options applicable to the layer list</td>
</tr>
<tr>
<td>6</td>
<td>The header section which refers to the name which identifies the layer. This name exists to help</td>
</tr>
<tr>
<td></td>
<td>the user to better identify the layer</td>
</tr>
<tr>
<td>7</td>
<td>The header section which refers to the type of layer. It is included for informative purposes</td>
</tr>
<tr>
<td></td>
<td>as it is conceptually linked with the layer properties defined in the Layar publishing site</td>
</tr>
<tr>
<td>8</td>
<td>The header section which refers to the number of actions specifically associated with the layer</td>
</tr>
<tr>
<td>9</td>
<td>The header section which refers to the number of animations associated with the layer</td>
</tr>
<tr>
<td>10</td>
<td>The header section which refers to the number of hotspots associated with the layer</td>
</tr>
<tr>
<td>11</td>
<td>The header section which refers to the association between the AR Database layer and the layer</td>
</tr>
<tr>
<td></td>
<td>defined on the Layar publishing site.</td>
</tr>
</tbody>
</table>
Figure 40 displays the layer “edit screen” (the “submit screen” version, available when in creation mode, replaces the “save” button for “submit”) for the Layers page, which is the result of the user pressing the edit button (item 3 in Figure 39) when in layer “list mode”. The settings page allows the user to edit all content associated with the layer. The description of each numbered item as well as the association with the JSON response elements (refer to Table 10) is provided in Table 12.

### Table 12 – Layer edit/submit screen elements description and JSON association

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Associated JSON response element (refer to Table 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A button to move the current page back to the layer list mode (Figure 39)</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>A web user control which encapsulates all information associated with the layers, including other child web user controls to be mentioned in the following sections</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>The basic settings button link which comprises the information associated to a layer</td>
<td>N/A</td>
</tr>
<tr>
<td>4</td>
<td>The animations button link which, when clicked, opens the list of animations to be set at a layer level (to be applied to all hotspots within the layer)</td>
<td>N/A</td>
</tr>
<tr>
<td>5</td>
<td>The actions button link which, when clicked, opens the list of actions</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Chapter 4 – The ISEGI-NOVA AR Project

The hotspots button link which, when clicked, opens the list of hotspots associated with the layer

The “layer type” element refers to the layer type as defined in the Layer publishing site. This element is used to aid the user about the type of hotspots which should be added to the layer

The “layer name” element refers to the name used for information purposes. It is used to aid the user in the layer identification process

The “layer general description” element provides a more extensive description to aid the user to understand the logic behind the layer

The “associated service provider layer id” element defines an association between our AR Database layer and the unique layer id defined in the Layar publishing site

The “refresh interval” element stores the associated JSON output variable

The “refresh distance” element stores the associated JSON output variable

The “full refresh” element stores the associated JSON output variable

The “biw style” element stores the associated JSON output variable

A “save” button which saves the data from the “Basic Settings” section when the validation conditions in Table 13 are satisfied

A label element which indicates whether the panel is on “edit” or “creation” mode, as well as disclosing the layer name (item 7 from the current table)

Table 13 contains the supported data types and enumerations, and the validation constraints which apply to “save” (edit screen) or “submit” (creation screen) attempts. The save/submit actions shall only be successful when all data complies with the constraints defined in Table 13.

Table 13 – Layer edit/submit screen input elements with applicable validation constraints

<table>
<thead>
<tr>
<th>ID</th>
<th>Validation constraints</th>
<th>Mandatory element, either by JSON requirements or by GUI design</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>The drop down list displays three options: 2D Layer, 2D/3D Layer or 2D/3D Layer (Vision enabled)</td>
<td>True</td>
</tr>
<tr>
<td>8</td>
<td>All characters are supported, except single quotes.</td>
<td>True</td>
</tr>
<tr>
<td>9</td>
<td>All characters are supported, except single quotes.</td>
<td>True</td>
</tr>
<tr>
<td>10</td>
<td>All characters are supported, except single quotes</td>
<td>False</td>
</tr>
<tr>
<td>11</td>
<td>Only integers are supported</td>
<td>False</td>
</tr>
<tr>
<td>12</td>
<td>Only integers are supported</td>
<td>False</td>
</tr>
<tr>
<td>13</td>
<td>Only true/false options are supported</td>
<td>True</td>
</tr>
</tbody>
</table>
The drop down list displays two options: “classic” and “collapsed”

Using Figure 40 as reference, we will introduce the other AR elements which are manageable within the layer: actions, animations and hotspots. Figure 41, Figure 42, and Figure 43 illustrate three layer management usage scenarios whose interaction details are resumed in Table 14.

Figure 41 - Layer resource managing scenarios (scenario 1- actions)

Figure 42 - Layer resource managing scenarios (scenario 2 - animations)
Chapter 4 – The ISEGi-NOVA AR Project

Figure 43 - Layer resource managing scenarios (scenario 3 - hotspots)

Table 14 - Layer resource managing scenarios description

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1        | - Scenario 1 is the result of the user pressing the “actions” link associated with the layer settings page  
- The actions panel, for the layer case, displays the list of actions associated with the layer. These actions are applied to the layer properties menu in the client Layar AR browser.  
- This actions panel allows the user to add and edit actions previously defined in the AR Database or create new actions and instantly associate them with the layer  
- The actions-related features will be explained in section 0 |
| 2        | - Scenario 2 is the result of the user pressing the “animations” link associated with the layer settings page  
- The animation panel, when embedded in the layer resources, displays the animations set at a layer level. These animation settings, when set at a layer level, will be applied to all of the hotspots associated with the layer.  
- This animation panel allows the user to add previously created animations. It does not allow animations creation per se. By design choice, to create animations the user must use the main Animation page (refer to section 4.5.6)  
- The animation-related features will be explained in section 4.5.6 |
| 3        | - Scenario 3 is the result of the user pressing the “hotspots” link associated with the layer settings page  
- The hotspots panel, when embedded in the layer resources, displays the list of hotspots associated with the selected layer  
- This panel allows the user to add and edit hotspots previously defined in the AR Database or to create new hotspots and instantly associate them with the layer  
- The hotspots-related features will be explained in section 4.5.4 |

4.5.4. Hotspots AR resources

The hotspot resources represent the “thick” of the Augmented Reality experience, as far as the end-user is concerned; they exist through the form of virtual elements added into the real world. Figure 44 exemplifies the scenarios of either managing hotspots through the main Hotspots resources page (scenario 1) or within the Layers resources page (scenario 2).
Figure 44 – Hotspot resources “list mode” managing scenarios

Figure 44 shows the Hotspot panels in “list mode”, displaying all available hotspots in the AR Database (scenario 1) and the hotspots associated with a specific layer (scenario 2). The numbered items are explained in Table 15.

Table 15 - Hotspots “list mode” elements description

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A button which opens a new panel for creating a new hotspot</td>
</tr>
<tr>
<td>2</td>
<td>The header section which displays the available options for the hotspots</td>
</tr>
<tr>
<td>3</td>
<td>A button which allows the user to edit the settings of a hotspot</td>
</tr>
<tr>
<td>4</td>
<td>A button which allows the user to delete a hotspot. This action does not delete any animations or actions associated to the hotspot because these resources can exist without the hotspots.</td>
</tr>
</tbody>
</table>
| 5  | The header section which refers to the available hotspot types:  
  - geolocation for hotspots which use GPS coordinates  
  - layarvision for hotspots which rely on the layer image recognition technology |
| 6  | The header section which refers to the name which identifies the hotspot. This name exists to help the user to better identify the hotspot |
| 7  | The header section that refers to the number of actions specifically associated with the hotspot |
| 8  | The header section that refers to the number of animations specifically associated with the hotspot |
| 9  | The header section that refers to the number of layers to which the hotspot is associated |
| 10 | A button (available in the layer scenario only) which allows the association of a previously existing... |
hotspot in the AR Database. It only allows adding hotspots which do not belong to the list of hotspots associated with the selected layer.

11 A button (available in the layer scenario only) that allows the creation of new hotspots while instantly associating them with the selected layer.

12 A button (available in the layer scenario only) that allows the user to remove the association between a hotspot and the selected layer. It does not delete the hotspot.

Figure 45, Figure 46, and Figure 47, display three “edit screen” scenarios (the “submit screen” version, available when in creation mode, replaces the “save” button for “submit”) for the Hotspots page which is the result of the user pressing the edit button (item 3 in Figure 44), when in hotspot “list mode”. The hotspot settings page allows the user to edit all content associated with the hotspot. Additionally, Figure 48 and Figure 49 display two navigation scenarios which occur when the user changes the hotspot type from “GeoLocation” (GPS coordinates based POI) to a “LayarVision” (image recognition based POI). The “LayarVision” hotspot type selection alters the available “basic settings” hiding the GPS coordinate elements, removes the “Brief Information Widget” section from the hotspot link area, and replaces the “Hotspot Indicator” with a “Vision Content” area (item 4.1 in Figure 48).
Chapter 4 - The ISEGI-NOVA AR Project

Figure 46 - Hotspot resources - Brief information widget scenario

Figure 47 - Hotspot resources - Hotspot indicator scenario
Figure 48 - Hotspot resources - Layar Vision POI basic settings

Figure 49 - Hotspot resources - Layar Vision POI Vision content settings
The description of each numbered item in the five mentioned scenarios (Figure 45, Figure 46, Figure 47, Figure 48, and Figure 49) as well as the association with the JSON response elements (refer to Table 10) is provided in Table 16.

Table 16 - Hotspot edit/submit screen (all scenarios) elements description and JSON association

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Associated JSON response element (refer to Table 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The link which opens the hotspot basic settings area</td>
<td>N/A</td>
</tr>
<tr>
<td>1.2</td>
<td>A label which indicates the name of the hotspot being edited</td>
<td>N/A</td>
</tr>
<tr>
<td>1.3</td>
<td>The “name” element is used for information purposes. It helps the user identify the hotspot</td>
<td>N/A</td>
</tr>
<tr>
<td>1.4</td>
<td>The “general description” element provides a more extensive description to aid the user to understand the logic behind the hotspot</td>
<td>N/A</td>
</tr>
<tr>
<td>1.5</td>
<td>The “hotspot type” element indicates whether the hotspots is geo-referenced or uses the layer Vision image recognition technology</td>
<td>N/A</td>
</tr>
<tr>
<td>1.6</td>
<td>The “latitude” element stores the associated JSON output variable</td>
<td>root.hotspots.anchor.geolocation.lat</td>
</tr>
<tr>
<td>1.7</td>
<td>The “longitude” element stores the associated JSON output variable</td>
<td>root.hotspots.anchor.geolocation.lon</td>
</tr>
<tr>
<td>1.8</td>
<td>The “altitude” element stores the associated JSON output variable</td>
<td>root.hotspots.anchor.geolocation.alt</td>
</tr>
<tr>
<td>1.9</td>
<td>A “save” button which saves the data from the hotspot when the validation conditions in Table 13 are satisfied</td>
<td>N/A</td>
</tr>
<tr>
<td>2.1</td>
<td>The link which opens the hotspot BIW area</td>
<td>N/A</td>
</tr>
<tr>
<td>2.2</td>
<td>The multi-language “title” element stores the associated JSON output variable</td>
<td>root.hotspots.text.title</td>
</tr>
<tr>
<td>2.3</td>
<td>The multi-language “description” element stores the associated JSON output variable</td>
<td>root.hotspots.text.description</td>
</tr>
<tr>
<td>2.4</td>
<td>The multi-language “footnote” element stores the associated JSON output variable</td>
<td>root.hotspots.text.footnote</td>
</tr>
<tr>
<td>2.5</td>
<td>Web user control which holds the “BIW image” element and stores the associated JSON output variable. This control will be explained in section 4.5.7</td>
<td>root.hotspots.imageURL</td>
</tr>
<tr>
<td>2.6</td>
<td>The “BIW style” element stores the associated JSON output variable (classic or collapsed)</td>
<td>root.hotspots.biwStyle</td>
</tr>
<tr>
<td>2.7</td>
<td>The “show small BIW” element stores the associated JSON output variable</td>
<td>root.hotspots.showSmallBiw</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>JSON Path</td>
</tr>
<tr>
<td>---</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>2.8</td>
<td>The “show BIW on click” element stores the associated JSON output variable</td>
<td>root.hotspots.showBiwOnItemClick</td>
</tr>
<tr>
<td>3.1</td>
<td>The link which opens the hotspot indicator area</td>
<td>N/A</td>
</tr>
<tr>
<td>3.2</td>
<td>The “In Focus” element stores the associated JSON output variable</td>
<td>root.Hotspots.inFocus</td>
</tr>
<tr>
<td>3.3</td>
<td>This radio button list defines the type of visual element that will</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>represent the POI/hotspot on screen. This is where the user will define</td>
<td></td>
</tr>
<tr>
<td></td>
<td>if the hotspot is represented by the default icon (set in the Layar</td>
<td></td>
</tr>
<tr>
<td></td>
<td>publishing site), by a custom icon, or by a 2D/3D real life size object</td>
<td></td>
</tr>
<tr>
<td>3.4</td>
<td>Web user control which holds the “Icon” element and stores the</td>
<td>root.hotspots.Icon</td>
</tr>
<tr>
<td></td>
<td>associated JSON output variable. This control will be explained in</td>
<td></td>
</tr>
<tr>
<td></td>
<td>section 4.5.7</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>The links in this area are changed (when the user changes the hotspot</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>to “Layar Vision” type), resulting in the removal of the BIW section and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the replacement of the “hotspot indicator” area with a “vision content”</td>
<td></td>
</tr>
<tr>
<td></td>
<td>section.</td>
<td></td>
</tr>
<tr>
<td>4.2</td>
<td>The “reference image name” element replaces the GPS coordinate</td>
<td>root.hotspots.anchor.referencelimage</td>
</tr>
<tr>
<td></td>
<td>elements.</td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>The link which opens the vision content area. This area defines the</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>2D/3D representation of the POI as well as the transformations to be applied</td>
<td></td>
</tr>
<tr>
<td>5.2</td>
<td>Filtered radio button list which shows the only available hotspot representation type for the “Layar Vision” hotspots: the custom object.</td>
<td>N/A</td>
</tr>
<tr>
<td>5.3</td>
<td>Web user control which holds the “Object” element and stores the</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>associated JSON output variable. This control will be explained in</td>
<td></td>
</tr>
<tr>
<td></td>
<td>section 4.5.8</td>
<td></td>
</tr>
<tr>
<td>5.4</td>
<td>The “scale” element stores the associated JSON output variable</td>
<td>root.hotspots.transform.scale</td>
</tr>
<tr>
<td>5.5</td>
<td>The “apply rotation” element indicates whether the user intends to use</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>a rotation transformation. Elements 5.6, 5.7, 5.8, 5.9 and 5.10 are hidden</td>
<td></td>
</tr>
<tr>
<td></td>
<td>when this setting is unchecked</td>
<td></td>
</tr>
<tr>
<td>5.6</td>
<td>The “relative” element stores the associated JSON output variable</td>
<td>root.hotspots.transform.rotate.rel</td>
</tr>
<tr>
<td>5.7</td>
<td>The “angle” element stores the associated JSON output variable</td>
<td>root.hotspots.transform.rotate.angle</td>
</tr>
<tr>
<td>5.8,</td>
<td>The axis “X”, “Y” and “Z” elements store the associated JSON output variables</td>
<td>root.hotspots.transform.rotate.*X,Y,Z</td>
</tr>
<tr>
<td>5.9,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.11</td>
<td>The “apply translation” element indicates whether the user intends to use</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>of a translation transformation. Items 5.12, 5.13, 5.14 are hidden if</td>
<td></td>
</tr>
</tbody>
</table>
Table 17 contains the supported data types and enumerations, and the validation constraints which apply to “save” (edit mode) or “submit” (creation mode) attempts. The save/submit actions shall only be successful when all data complies with the constraints defined in Table 17.

Table 17 – Hotspot edit/submit screen input elements with applicable validation constraints

<table>
<thead>
<tr>
<th>ID</th>
<th>Validation constraints</th>
<th>Mandatory element, either by JSON requirements or by GUI design</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3</td>
<td>All characters are supported, except single quotes</td>
<td>True</td>
</tr>
<tr>
<td>1.4</td>
<td>All characters are supported, except single quotes</td>
<td>True</td>
</tr>
<tr>
<td>1.5</td>
<td>The drop down list displays two options: “geolocation” and “layarvision”</td>
<td>True</td>
</tr>
<tr>
<td>1.6</td>
<td>Integers and Float values are supported.</td>
<td>True</td>
</tr>
<tr>
<td>1.7</td>
<td>Integers and Float values are supported.</td>
<td>True</td>
</tr>
<tr>
<td>1.8</td>
<td>Integers and Float values are supported.</td>
<td>False</td>
</tr>
<tr>
<td>2.2</td>
<td>All characters are supported, except single quotes. Both textboxes are mandatory</td>
<td>True</td>
</tr>
<tr>
<td>2.3</td>
<td>All characters are supported, except single quotes. Note: If one textbox has input value, both textboxes are considered mandatory</td>
<td>False</td>
</tr>
<tr>
<td>2.4</td>
<td>All characters are supported, except single quotes. Note: If one textbox has input value, both textboxes are considered mandatory</td>
<td>False</td>
</tr>
<tr>
<td>2.6</td>
<td>The drop down list displays two options: “classic” and “collapsed”</td>
<td>True</td>
</tr>
<tr>
<td>2.7</td>
<td>Only true/false options are supported</td>
<td>True</td>
</tr>
<tr>
<td>2.8</td>
<td>Only true/false options are supported</td>
<td>True</td>
</tr>
<tr>
<td>3.2</td>
<td>Only true/false options are supported</td>
<td>True</td>
</tr>
<tr>
<td>3.3</td>
<td>The radio button list contains 3 options: default, custom icon, or custom object</td>
<td>True</td>
</tr>
<tr>
<td>4.2</td>
<td>All characters are supported, except single quotes</td>
<td>True</td>
</tr>
<tr>
<td>5.4</td>
<td>Integers and Float values are supported</td>
<td>False</td>
</tr>
<tr>
<td>5.5</td>
<td>Only true/false options are supported</td>
<td>True</td>
</tr>
<tr>
<td>5.6</td>
<td>Only true/false options are supported</td>
<td>True</td>
</tr>
<tr>
<td>5.7</td>
<td>Integers and Float values are supported</td>
<td>True</td>
</tr>
<tr>
<td>5.8</td>
<td>Integers and Float values are supported</td>
<td>True</td>
</tr>
<tr>
<td>5.9</td>
<td>Integers and Float values are supported</td>
<td>True</td>
</tr>
</tbody>
</table>
5.10 Integers and Float values are supported  
5.11 Only true/false options are supported  
5.12 Integers and Float values are supported  
5.13 Integers and Float values are supported  
5.14 Integers and Float values are supported

### 4.5.5. Actions AR resources

The action resources represent additional content that may be associated to a hotspot or layer, to enrich the Augmented Reality experience. When a user interacts with a hotspot or checks the layer’s properties, the action options add the possibility of reproducing media files, navigate on websites, send SMS, make phone calls, and even open new layers. Figure 50, Figure 51, and Figure 52 show the Actions panels in “list mode”, displaying all available actions in the AR Database (scenario 1), the actions associated with a specific layer (scenario 2), and the actions associated with a specific hotspot (scenario 3), respectively. The numbered items are explained in Table 18.

![Figure 50 - Actions resources “list mode” managing scenarios (scenario 1)](image1)

![Figure 51 - Actions resources “list mode” managing scenarios (scenario 2)](image2)
Figure 52 - Actions resources “list mode” managing scenarios (scenario 3)

Table 18 - Action "list mode" elements description

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A button which opens a new panel for creating a new action</td>
</tr>
</tbody>
</table>
| 2  | A drop down list that filters the actions by type. The allowed types are:  
   - Asynchronous Call – in our AR project this type is used to navigate to a different layer  
   - Audio – open an mp3 audio file  
   - Internal App Call – produce a phone call, SMS, or email action  
   - Video – open an mp4 video file  
   - Web Page – open a web page |
| 3  | The header section that refers to options available for each list item |
| 4  | A button which allows the user to edit an action |
| 5  | A button which allows the user to delete an action |
| 6  | The header section which refers to the name which identifies the action. This name exists to help the user to better identify the action |
| 7  | This header section refers to the type of action of the list item in the panel. The available types are indicated in element 2 of the current table |
| 8  | The header section that refers to the label name (in English) which appears on the Layar AR browser interface as an action button. Given the multi-language features in our AR project, only the English language content is displayed when in “list-mode” |
| 9  | The header section that refers to the Uniform Resource Identifier or URI (in English) which points to the action content. Given the multi-language features in our AR project, only the English language content is displayed when in “list-mode” |
| 10 | The header section which refers to the number of associations between layers and/or hotspots |
| 11 | A label that provides indications about the panel’s context |
| 12 | A button (available when embedded in the layer and hotspot resources) which allows the association of a previously existing action in the AR Database. It only allows adding actions which do not belong to the list of actions associated with the selected layer/hotspot |
| 13 | A button (available when embedded in the layer and hotspot resources) that allows the creation of new actions while instantly associating them with the selected layer/hotspot |
| 14 | A button (available when embedded in the layer and hotspot resources) that allows the user to remove the association between a given action and the selected layer/hotspot |
When the displayed URI belong the “Audio”, “Video” or “Web Page” category, the content is displayed as a link so that the user can download and preview the resources.

Figure 53 displays four “edit screen” scenarios (the “submit screen” version, available when in creation mode, replaces the “save” button for “submit”) for the Actions page which is the result of the user pressing the edit button (item 4 in Figure 50), when in action “list mode”. The description of each numbered item, in Figure 53, as well as the association with the JSON response elements (refer to Table 10) is provided in Table 19.
Table 19 - Actions edit/submit screen (all scenarios) elements description and JSON association

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Associated JSON response element (refer to Table 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The “name” element is used for information purposes. It helps the user to identify the action.</td>
<td>N/A</td>
</tr>
<tr>
<td>1.2</td>
<td>The “general description” element provides a more extensive description to help the user to understand the logic behind the action.</td>
<td>N/A</td>
</tr>
<tr>
<td>1.3</td>
<td>The “content type” element stores the associated JSON output variable. It provides a drop down list selection with the following options: • Audio – generates “audio/mp3” in the JSON output • Internal App Call - generates “application/vnd.layar.internal” in the JSON output • Video - generates “video/mp4” in the JSON output • Web Page - generates “text/html” in the JSON output</td>
<td>actions.contentType</td>
</tr>
<tr>
<td>1.4</td>
<td>A drop down list with a list with a set of icons which will appear alongside the “label” item (1.5 in Figure 53). This item was built according to available types in the Layar site’s action section (Layar 2012a) and made available in Figure 54</td>
<td>actions.activityType</td>
</tr>
<tr>
<td>1.5</td>
<td>A button that displays a popup with the list of supported BIW activity types (refer to Figure 54) to facilitate the selection process in item 1.4.</td>
<td>N/A</td>
</tr>
<tr>
<td>1.6</td>
<td>The “label” element stores the associated JSON output variable. The button will display the text set in this variable.</td>
<td>actions.label</td>
</tr>
<tr>
<td>1.7</td>
<td>The “uri” element stores the associated JSON output variable. Given the current “content type” selection (item 1.3), the item displays an upload area that only allows uploads with an mp3 extension</td>
<td>actions.uri</td>
</tr>
<tr>
<td>1.8</td>
<td>The “close biw on click” element stores the associated JSON output variable</td>
<td>actions.closeBiwOnClick</td>
</tr>
<tr>
<td>1.9</td>
<td>The “auto trigger only” element stores the associated JSON output variable</td>
<td>actions.autoTriggerOnly</td>
</tr>
<tr>
<td>1.10</td>
<td>The “auto trigger” element (available for layar-vision POIs only) stores the associated JSON output variable</td>
<td>actions.autoTrigger</td>
</tr>
<tr>
<td>1.11</td>
<td>The “auto trigger range” element (available for geo-referenced POIs only) stores the associated JSON output variable</td>
<td>actions.autoTriggerRange</td>
</tr>
<tr>
<td>1.12</td>
<td>A “save” button which saves the data when the validation conditions in Table 20 are satisfied</td>
<td>N/A</td>
</tr>
<tr>
<td>2.1</td>
<td>In this scenario, where the user selected a content of type “Internal App Call”, the prefix drop down list used for the “uri” displays the following options: Phone, SMS, Email, Open Layer.</td>
<td>N/A</td>
</tr>
<tr>
<td>3.1</td>
<td>In this scenario, where the user selected a content of type “Video”, the item displays an upload area that only allows mp4 uploads</td>
<td>N/A</td>
</tr>
<tr>
<td>4.1</td>
<td>In this scenario, where the user selected a content of type “Web page”, the prefix drop down list used for the “uri” displays the “http://” prefix</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Figure 54 – Help panel that displays the images associated with each BIW activity type

Table 20 contains the supported data types, enumerations, and the validation constraints which apply to “save” (edit mode) or “submit” (creation mode) attempts. The save/submit actions shall only be successful when all data complies with the constraints defined in Table 20.

Table 20 - Actions edit/submit screen input elements with applicable validation constraints

<table>
<thead>
<tr>
<th>ID</th>
<th>Validation constraints</th>
<th>Mandatory element, either by JSON requirements or by GUI design</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>All characters are supported, except single quotes</td>
<td>True</td>
</tr>
<tr>
<td>1.2</td>
<td>All characters are supported, except single quotes</td>
<td>True</td>
</tr>
<tr>
<td>1.3</td>
<td>The drop down list options enforce the consistency</td>
<td>True</td>
</tr>
<tr>
<td>1.4</td>
<td>The drop down list options enforce the consistency</td>
<td>True</td>
</tr>
<tr>
<td>1.6</td>
<td>All characters are supported, except single quotes</td>
<td>True</td>
</tr>
<tr>
<td>1.7</td>
<td>The upload element only allows mp3 uploads</td>
<td>True</td>
</tr>
<tr>
<td>1.8</td>
<td>Only true/false options are supported</td>
<td>True</td>
</tr>
<tr>
<td>1.9</td>
<td>Only true/false options are supported</td>
<td>True</td>
</tr>
<tr>
<td>1.10</td>
<td>Only true/false options are supported (available for layar-vision POIs only)</td>
<td>True</td>
</tr>
<tr>
<td>1.10</td>
<td>Integer values are supported (available for geo-referenced POIs only)</td>
<td>False</td>
</tr>
<tr>
<td>2.1</td>
<td>All characters are supported, except single quotes</td>
<td>True</td>
</tr>
</tbody>
</table>
### 4.5.6. Animations AR resources

Animations represent the act of associating predefined movement to a POI representation in the live phone camera view. Figure 55 shows the main Animation resources panel in “list mode”, which is the only interface available for the users to create animations. The numbered items are explained in Table 21.

![Animation Manager](image)

**Figure 55 - Animation resources in “list mode”**

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A button which opens a new panel for creating a new animation</td>
</tr>
<tr>
<td>2</td>
<td>The header section that refers to options available for each list item. The animations of type “simple” are introduced by default in the AR Database and do not allow the edit/delete options.</td>
</tr>
<tr>
<td>3</td>
<td>A button which allows the user to edit an animation</td>
</tr>
<tr>
<td>4</td>
<td>A button which allows the user to delete an animation</td>
</tr>
<tr>
<td>5</td>
<td>The header section which indicates whether the animation is user-created (“custom”) or system-defined (“simple”)</td>
</tr>
<tr>
<td>6</td>
<td>The header section which refers to the name which identifies the animation. This name exists to help the user to better identify the animation</td>
</tr>
<tr>
<td>7</td>
<td>This header section signals the number of layers and/or hotspots associated with the animation item</td>
</tr>
</tbody>
</table>

After the animations are created in the main resources page, they can be associated to either hotspots or layers, like displayed in Figure 56 and described in Table 22. Scenario 1 shows a use case of adding “custom” animations to a hotspot and scenario
2 shows the use case of adding “simple” animations to a layer (both “simple” and “custom” animations are applicable to hotspots and layers).

Figure 56 - Animation “adding” scenarios

Table 22 - Animation "adding" scenarios description

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The label which states the current type of animations selected (either “simple” or “custom”)</td>
</tr>
<tr>
<td>1.2</td>
<td>A button to switch between the available animation types (mentioned in item 1.1)</td>
</tr>
</tbody>
</table>
1.3 A label which displays the type of custom animation event ("On Create", "On Update", "On Delete", "On Focus", or "On Click")

1.4 The button which opens a small panel to add more animations to the list

1.5 The header section which indicates the short name associated with the animation

1.6 A button to remove the association between the animation and the hotspot (or layer)

1.7 Arrows that change the order on which each animation will occur within the selected event (item 1.3)

1.8 Message area that warns about "lack of content" and successful animation additions

1.9 Example where the panel is in "add" mode after the user clicks item 1.4. In this mode, the panel only shows the animations not already associated with the hotspot (or layer)

1.10 Button to move back to the hotspot or layer-associated animations

2.1 A label which characterises the "simple" panel

2.2 A remove button which disassociates a "simple" animation from the layer (or hotspot). If there are no simple animations associated, an "add" button appears next to item 2.1

Figure 57 displays the animation "edit screen" (the "submit screen" version, available when in creation mode, replaces the "save" button for "submit") for the Animations page which is the result of the user pressing the edit button (item 3 in Figure 55). The description of each numbered item in Figure 57 as well as the association with the JSON response elements (refer to Table 10) is provided in Table 23. The extensive specification of the animation terminology and meaning is available online in Layar’s documentation area for animations (Layar 2012b).
### Table 23 - Animations edit/submit screen elements description and JSON association

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Associated JSON response element (refer to Table 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The “name” element is used for information purposes. It helps the user to identify the action</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>The “general description” element provides a more extensive description to help the user to understand the logic behind the action</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>The “custom animation type” element stores the associated JSON output variable. It provides a drop down list selection with the following options, which are sent to the JSON output: scale, rotate, translate</td>
<td>animations.type</td>
</tr>
<tr>
<td>4</td>
<td>The “interpolation” element stores the associated JSON output variable. It provides a drop down list selection with the following options, which are sent to the JSON output: linear, accelerateDecelerate, accelerate, decelerate, bounce, cycle, anticipateOvershoot, anticipate, overshoot.</td>
<td>animations.interpolation</td>
</tr>
<tr>
<td>5</td>
<td>The “interpolation parameter” element stores the associated JSON output variable</td>
<td>animations.interpolationParam</td>
</tr>
<tr>
<td>6</td>
<td>The “length” element stores the associated JSON output variable</td>
<td>animations.length</td>
</tr>
<tr>
<td>7</td>
<td>The “persist” element stores the associated JSON output variable</td>
<td>animations.persist</td>
</tr>
<tr>
<td>8</td>
<td>The “repeat” element stores the associated JSON output variable</td>
<td>animations.repeat</td>
</tr>
<tr>
<td>9</td>
<td>The “from” element stores the associated JSON output variable</td>
<td>animations.from</td>
</tr>
<tr>
<td>10</td>
<td>The “to” element stores the associated JSON output variable</td>
<td>animations.to</td>
</tr>
<tr>
<td>11</td>
<td>The “delay” element stores the associated JSON output variable</td>
<td>animations.delay</td>
</tr>
<tr>
<td>12</td>
<td>The “axis” element stores the associated JSON output variables for X, Y and Z</td>
<td>animations.axis.* X,Y,Z</td>
</tr>
<tr>
<td>15</td>
<td>A “save” button which saves the data from the animation settings when the validation conditions in Table 24 are satisfied</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 24 contains the supported data types and enumerations, and the validation constraints which apply to “save” (edit mode) or “submit” (creation mode) attempts. The save/submit actions shall only be successful when all data complies with the constraints defined in Table 24.

### Table 24 – Animations edit/submit screen input elements with applicable validation constraints

<table>
<thead>
<tr>
<th>ID</th>
<th>Validation constraints</th>
<th>Mandatory element, either by JSON requirements or by GUI design</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>All characters are supported, except single quotes</td>
<td>True</td>
</tr>
</tbody>
</table>
### 4.5.7. BIW Image and POI Icon AR resources

As explained in section 3.2 (The Layar AR browser), the BIW images are part of the hotspots’ brief information widget area. In the same section we explained that the POI itself can be represented by the default iconology defined in the Layar publishing site, by a 2D/3D real world object, or by an icon image. This section approaches the BIW images and the icons, as they are defined and managed within the same web user control.

Figure 58 exemplifies the scenarios of managing BIW images and Icons through the main resources page (scenario 1). The BIW images (scenario 2) and Icons (scenario 3) are also manageable through the BIW Image/Icon user control which is embedded in the hotspots’ main resources properties. The numbered items are explained in Table 25.
Chapter 4 – The ISEGi-NOVA AR Project

### Figure 58 - Icon/Biw image resources in "list-mode" scenarios

### Table 25 - Biw Image/icons "list mode" element description

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>A button (exclusive for the BIW image/icon resources page, scenario 1) which allows the creation of either a BIW image or a POI representation icon, according to the filter selection (item 1.2)</td>
</tr>
<tr>
<td>1.2</td>
<td>A drop down list filter (exclusive for the BIW Image/Icon resources page, scenario 1) which filters the content displayed in screen and changes the button logic for item 1.1 to create a BIW Image or an Icon, accordingly</td>
</tr>
<tr>
<td>1.3</td>
<td>The header section which refers to the actions applicable to the listed element</td>
</tr>
<tr>
<td>1.4</td>
<td>A button which allows the user to edit a BIW image (or icons, scenario 3)</td>
</tr>
<tr>
<td>1.5</td>
<td>A button which allows the user to delete the BIW Image/Icon and also the association between a BIW image/Icon and the hotspot</td>
</tr>
</tbody>
</table>
Chapter 4 - The ISEGI-NOVA AR Project

1.6 The header section which refers to the listed element type (Icon or BIW image)

1.7 The header section which refers to the name which identifies the BIW image (or icon in scenario 3). This name exists to help the user to better identify the BIW image by a suggestive name

1.8 This section refers to the visual preview of the BIW image (or icon, scenario 3)

1.9 The header section that refers to number of hotspots associated with the BIW image or Icon

2.1 A button which allows the association of a previously existing BIW image (or icons, scenario 3) to a hotspot. It only allows adding BIW images other than the current hotspot selection

2.2 A button that allows the creation of new BIW images (or icons, scenario 3) while instantly associating them with the selected hotspot

2.3 A button which allows the user to remove the association between a BIW image (or icons in scenario 2) and a hotspot

Figure 59 displays the BIW Image/Icons “edit screen” (the “submit screen” version, available when in creation mode, replaces the “save” button for “submit”) which is the result of the user pressing the edit button (item 1.4 in Figure 58). The description of each numbered item in Figure 59 as well as the association with the JSON response elements (refer to Table 10) is provided in Table 26.

Figure 59 - Biw Image/Icons "edit screen" scenario
Table 26 - BIW Image/Icons edit/submit screen elements description and JSON association

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Associated JSON response element (refer to Table 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A button which moves to the BIW Image/Icon list</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>The “name” element is used for information purposes. It helps the user to identify the BIW image or icon</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>The “general description” element provides a more extensive description to help the user to understand the logic behind the BIW image or icon</td>
<td>N/A</td>
</tr>
<tr>
<td>4</td>
<td>The “icon set id” element stores the associated JSON output variable. This item is only available when creating an icon.</td>
<td>root.hotspots.icon.type</td>
</tr>
<tr>
<td>5</td>
<td>The file upload element loads the content to our AR server</td>
<td>root.hotspots.icon or root.hotspots.biwImageUrl</td>
</tr>
<tr>
<td>6</td>
<td>The content preview area which shows the visual preview of the uploaded BIW image or icon</td>
<td>N/A</td>
</tr>
<tr>
<td>7</td>
<td>A “save” button which saves the data from the BIW Image/Icon settings when the validation conditions in Table 27 are satisfied</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 27 contains the supported data types, enumerations, and the validation constraints which apply to “save” (edit mode) or “submit” (creation mode) attempts. The save/submit actions shall only be successful when all data complies with the constraints defined in Table 27.

Table 27 – BIW image/Icons edit/submit screen input elements with applicable validation constraints

<table>
<thead>
<tr>
<th>ID</th>
<th>Validation constraints</th>
<th>Mandatory element, either by JSON requirements or by GUI design</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>All characters are supported, except single quotes</td>
<td>True</td>
</tr>
<tr>
<td>3</td>
<td>All characters are supported, except single quotes</td>
<td>True</td>
</tr>
<tr>
<td>4</td>
<td>Only integers are supported. This item is only applicable for icons</td>
<td>False</td>
</tr>
<tr>
<td>5</td>
<td>The “file uploader” element validates the jpg, png, and gif extensions</td>
<td>True</td>
</tr>
</tbody>
</table>
4.5.8. Object AR resources

As explained in section 3.2 (Layar AR browser) and referred in the hotspot section (4.5.4), the POI indicator, shown on the live camera view of the smartphone, can be represented using 2D/3D real world objects. This section approaches the POI indicator object resources’ web user control.

Figure 60 exemplifies two scenarios for managing the Object resources. Scenario 1 displays the management options obtained through the main Object resources page and scenario 2 displays the object resource control embedded in the hotspots’ main resources properties. The numbered items are explained in Table 28.

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>A button that opens a panel where the user can create a new object (image, video, or 3D model)</td>
</tr>
<tr>
<td>1.2</td>
<td>A filter which restricts the presented resources according to the three available types</td>
</tr>
<tr>
<td>1.3</td>
<td>The header section which refers to the options available for the listed item</td>
</tr>
<tr>
<td>1.4</td>
<td>A button which allows the user to edit an object</td>
</tr>
<tr>
<td>1.5</td>
<td>A button which allows the user to delete the object and its associations with the hotspots</td>
</tr>
<tr>
<td>1.6</td>
<td>The header section which refers to the object type (an image, a 3D model, or a video)</td>
</tr>
<tr>
<td>1.7</td>
<td>The header section which refers to the name which identifies the object. This name exists to help the user to better identify object. Clicking on the name of the object will display the associated description in a popup.</td>
</tr>
</tbody>
</table>
1.8 The header section which refers to the “url” which contains the object. When defined, the “reduced url” can be obtained by clicking the “url” of the object which will display the associated “reduced url” in a popup. The object’s contents are downloadable through a link.

1.9 The header section that refers to the number of hotspots associated with the object

2.1 A button which allows the association of a previously existing objects in the AR Database. It only allows adding objects other than the one which is currently selected for the hotspot

2.2 A button that allows the creation of new objects while instantly associating them with the selected hotspot

2.3 A button which allows the user to remove the association between an object and a hotspot

Figure 61 - Object main resources "edit screen"

Figure 61 displays the objects’ “edit screen” (the “submit screen” version, available when in creation mode, replaces the “save” button for “submit”) which is the result of the user pressing the edit button (item 1.4 in Figure 60). The description of each numbered item in Figure 61 as well as the association with the JSON response elements (refer to Table 10) is provided in Table 29.

Table 29 - Objects edit/submit screen elements description and JSON association

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Associated JSON response element (refer to Table 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A button which allows the user to move back to the object list</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>The “object type” element which defines the type of selected object (image, video or 3D model)</td>
<td>root.hotspots.object.contentType</td>
</tr>
<tr>
<td>3</td>
<td>The “name” element is used for information purposes. It helps the user</td>
<td>N/A</td>
</tr>
</tbody>
</table>
4. The “general description” element provides a more extensive description to help the user to understand the logic behind the object

5. The “file upload” element loads the “resource url” content to our AR server

6. The “file upload” element loads the “reduced url” content to our AR server

7. The “real world size” element stores the size of the object in the real world, in meters

8. The content preview area which shows the visual preview of all image objects

9. A “save” button which saves the data from the object settings when the validation conditions in Table 30 are satisfied

Table 30 contains the supported data types, enumerations, and the validation constraints which apply to “save” (edit mode) or “submit” (creation mode) attempts. The save/submit actions shall only be successful when all data complies with the constraints defined in Table 30.

Table 30 - Objects edit/submit screen input elements with applicable validation constraints

<table>
<thead>
<tr>
<th>ID</th>
<th>Validation constraints</th>
<th>Mandatory element, either by JSON requirements or by GUI design</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>The drop down list allows the following options: Image, Video, and 3D models</td>
<td>True</td>
</tr>
<tr>
<td>3</td>
<td>All characters are supported, except single quotes</td>
<td>True</td>
</tr>
<tr>
<td>4</td>
<td>All characters are supported, except single quotes</td>
<td>True</td>
</tr>
</tbody>
</table>
| 5  | The “file uploader” element validates the download extensions:  
  - For image object type – jpg, png, and gif  
  - For video object type – mp4  
  - For 3D model object type – l3d | True |
| 6  | The validation conditions are the same as item 5, but item 6 is not mandatory | False |
| 7  | Integers and floats are supported. | True |
5. THE AR DEMONSTRATION LAYERS

5.1. OUR GUIDELINES

In order to present an AR world implementation that introduces our AR system to the Universidade Nova de Lisboa, and to validate the supported AR features, we created two demonstration layers, one for geo-referenced POIs and another for the Layar Vision POIs. Even though we tested all of the JSON response elements, individually, during the development of the AR Libs component and the AR Management Site, our demonstration AR worlds serve the purpose of a more practical real world testing. Additional system testing was done during two collaborations in ISEGI-NOVA projects, which occurred during the evolution of our AR project, and are pointed out in the contributions’ section 6.1, from this report’s closing chapter.

Before starting the creation process of the AR content, one very important factor to take into account is the motivation behind the content creation itself. Therefore, it is necessary to:

- Know how to classify a POI: “Is this an element that will interest anyone? In what context could it be fitted?”
- Obtain the GPS coordinates, in case the POI is geo-referenced;
- Provide reference images for image recognition purposes, in case the POI uses the Layar Vision technology;
- Add the necessary interaction and behaviour elements:
  - “What information should we add about this element?”
  - “Does it require images, icons or 3D elements?”
  - “Would it benefit from a button to open a web site, play a song or watch a video?”
  - “Do we want to animate the POIs so that they dance around, just for fun?”

Figure 62 shows an example of a conceptual creation flow to which the AR Management Site provides means:

1. The user creates a layer and defines its basic properties
2. The user creates an action and associates it to the current layer’s properties
3. The user creates two points of interest, with some basic information, and associates them to the current layer
4. The user creates an animation and associates it to one hotspot
5. The user creates an action and associates it to another hotspot

![Creating an AR world – example](image)

In the next section, we provide a walkthrough that describes the two created AR demonstration worlds, with the proposed features description and real on-campus photos of the created AR content, as well as the required steps on the Layar server side to make the content available to the smartphones.

### 5.2. Walkthrough

#### 5.2.1. Initial steps on the Layar publishing site

Before creating the AR content in our AR Management Site, we need to have a developer account in the Layar (2011) publishing site. With a developer account, a content provider can create layers like the ones exemplified for our demonstration AR worlds in Figure 63. For each layer, the user must define, among several administrative settings, the URL address of the web service (our AR Web Service, in this case) which will provide the content for that specific layer.
Figure 63 shows numbered items from which we introduce the following:

1. This is the unique id of the layer we are going to use in our Layar Vision world. This world will contain POIs that do not use GPS coordinates, but make use of image recognition technology instead. After creating this layer in the Layar publishing site, the only further required interaction with the Layar publishing site concerns uploading the images (one time only for each image) that the Layar Vision technology shall be able to recognise (to be detailed in section 5.2.3).

2. This is the unique id of the layer we are going to use in our geo-referenced world. This world will contain POIs which use exclusively GPS coordinates. During the definition of this layer, we added two filters (indicated in Figure 64), which allow the users, when interacting with the layer in the smartphone, to change the language of the presented contents and to adjust how far, in meters, they will be able to see into the AR World. This complies with one of the generic guideline requirements created for the ISEGI-NOVA AR project (item 8 in Table 1, page 29). After creating this layer, no further interaction is required with the Layar publishing site.

3. This column shows the status of each layer in terms of being publicly available or not. As we are creating demonstration layers, we have no intention of releasing content to the public yet, so the layers are kept in a testing/development mode. This state makes our AR content available only for specifically authorised users. If the layers were made public, they would appear on the search engine of any smartphone with the Layar application.
The unique ids for each layer in the Layar publishing site are very important as they allow the relation, as referred in section 4.5.3, between the content created in the AR Management Site (subsequently stored in the AR Database) and the requests for specific layer contents coming from the Layar AR browser.

Following these initial steps, in the next two sections, we will approach the main elements which comprise our AR demonstration worlds through a high level description.

### 5.2.2. The geo-referenced layer

Starting with an aerial view of the Universidade Nova de Lisboa’s campus, we identified several POIs which are signalled by numbers in Figure 65 and reflect the following organisation:

- The gates which allow access to the campus: the northeast entry gate (item 1) and the southwest entry gate (item 2);
- The Universidade Nova de Lisboa Rectory building (item 3);
- The institutions in the Campolide campus: Nova School of Business and Economics (item 4), the Nova School of Statistics and Information Management (item 5) and the Faculty of Law (item 6);
The university’s Halls of Residence building (item 7);
- A resting spot on the stairway next to the rectory (item 8);
- A 3D model of a blimp, outside of the campus, bearing the logo of the Universidade Nova de Lisboa (item 9).

Figure 65 – The geo-referenced POIs for the demonstration layer

5.2.2.1. POIs 1 and 2: the welcoming access gates

Since both of these points of interest represent the main access gates to the Universidade Nova de Lisboa’s campus, they were created with exactly the same elements and with the purpose of being the first welcome point into the university. The following characteristics were added to the POIs:

1. One image object with the logotype of the Universidade Nova de Lisboa (as displayed in Figure 66);
2. A vertical axis rotation animation, based on the accelerate/decelerate interpolation (as referred in section 4.3.2, Table 10), to be applied to the image object;
3. One action, to be triggered upon clicking the object, consisting of a welcome audio-guide-type message, created in two languages (English and Portuguese) and combining background classical music with speech.

Figure 66 – UNL logo
The on-campus interaction with these POIs, as exemplified in Figure 67, results in:

1. the end-user noticing the POI rotating image object;
2. the end-user clicking the object and triggering the audio message to start.

5.2.2.2. **POI 3: the Universidade Nova de Lisboa Rectory**

The Universidade Nova de Lisboa Rectory building POI, being a central administrative entity, includes elements that provide access to several types of information and is oriented to both current and prospect students. The following characteristics were added to the POI:

1. One image object with the logotype of the Universidade Nova de Lisboa (as displayed in Figure 68);
2. One action labelled “Institutional Site” (“Sítio Institucional”, in the Portuguese version) pointing to the website of the Universidade Nova de Lisboa;
3. One action labelled “NOVA presentation video” (“Vídeo de apresentação da NOVA”, in the Portuguese version) which triggers the reproduction of a promotional video about the university (English only);
4. One action labelled “Search courses” (“Procura de cursos”, in the Portuguese version) which triggers the opening of a site section that allows searching the available academic courses in the Universidade Nova de Lisboa;
5. One action labelled “Call International Relations Office” (“Chamar Gab. de Relações Internacionais”, in the Portuguese version), which triggers a direct phone call to the International Relations office;

6. One action labelled “Email International Relations Office” (“Email Gab. de Relações Internacionais”, in the Portuguese version), which triggers a direct email to the International Relations Office;

7. One action labelled the “SASNOVA - Social Welfare Services” (“SASNOVA-Serviços de Apoio Social”, in the Portuguese version) which triggers the opening of the Social Welfare Services’ institutional site.

Figure 68 – UNL Rectory logo

The on-campus interaction with this item, as exemplified in Figure 69, results in:

1. the end-user noticing the Rectory POI;

2. the end-user clicking one of the available actions and:
   a. opening Universidade Nova de Lisboa’s institutional site;
   b. watching a promotional video about the University;
   c. searching for academic courses;
   d. calling the International Relations office;
   e. emailing the International Relations office;
   f. opening SASNOVA’s institutional site.
Figure 69 – Rectory POI interaction
5.2.2.3. POIs 4 to 7: the institutes/universities/services of the Campolide campus

These four points of interest are included to represent the three academic units and the university’s Halls of Residence building, all located within the Campolide campus area, and represented with the following added characteristics:

- **POI 4 - NOVA School of Business and Economics (Faculdade de Economia, FE-UNL):** One image object with the logotype (as displayed in Figure 70);
- **POI 5 - NOVA School of Statistics and Information Management (Instituto Superior de Estatística e Gestão de Informação, ISEG-IUNL):** One image object with the logotype (as displayed in Figure 70);
- **POI 6 - NOVA Faculty of Law (Faculdade de Direito, FD-UNL):** One image object with the logotype (as displayed in Figure 70);
- **POI 7 - The University Halls Of Residence:** One image object with the Universidad Nova de Lisboa’s logo and a description (as displayed in Figure 70);

POIs 4, 5, and 6 include an action which points to each associated institutional site, while POI 7 includes an action which links to the SASNOVA site area which provides information about the Halls of Residence building.

![POI Images](image-url)

Figure 70 – The object images that represent each POI

The interaction result with each of the four mentioned POIs is explained in Figure 71, Figure 72, Figure 73 and Figure 74, which are organised in three screenshot phases, that correspond to visualising the POI, clicking the POI and experiencing the associated action.
Figure 71 – Interaction with the NOVA School of Business and Economics POI

Figure 72 - Interaction with ISEGI-NOVA POI

Figure 73 – Interaction with the NOVA Faculty of Law POI
5.2.2.4. **POIs 8 and POI 9: leisure and relaxation**

These two points of interest represent an example of a resting/entertaining moment to be experienced in the stairs next to the Rectory building. POI 8 refers to the actual stairs next to the rectory while POI 9 refers to a 3D zeppelin located outside of the campus, around the Águas Livres Aqueduct area. The following characteristics were added to the POIs:

- **POI 8** – The stairs next to the Rectory building:
  - One image object with the logotype (as displayed in Figure 75);
  - One action labelled “The NOVA Choir” (“O Coro da NOVA” in the Portuguese version) which triggers a song interpreted by the Universidade Nova de Lisboa’s choir;
  - One action labelled “The NOVA Zeppelin” (“O Zeppelin da NOVA”) which plays an audio message that invites the end-user to interact with the zeppelin POI.

Figure 75 – Stairway rest image
POI 9 – A 3D zeppelin placed around the Águas Livres Aqueduct:
- One 3D model representation of a zeppelin with the logo of the Universidade Nova de Lisboa (as exemplified in Figure 76);
- Two animations triggered upon clicking the 3D model:
  - A rotation effect around the vertical axis;
  - A zoom effect.

Figure 76 – The Nova 3D zeppelin model

The on-campus interaction with item 8, as exemplified in Figure 77, results in:
1. the end-user noticing POI 8;
2. the end-user clicking one of the available actions and:
   a. being invited to observe the beauty of campus with the sound the NOVA choir as a background track;
   b. being invited to notice POI 9 (the Nova Zeppelin) and to interact with it, triggering the animations;

Figure 77 – Leisure and relaxation actions – part I

Upon interaction with POI 9, as illustrated in Figure 78, the user shall notice a rotation applied to the NOVA zeppelin and a scale increase effect.
5.2.3. The Layar Vision layer

To demonstrate the features related with the image recognition technology (Layar Vision) made available by the Layar browser, we used the ISEGI facilities as our test scenario. Five POIs were created for the ISEGI building: four “mosaic” pictures and one office name tag montage (detailed in Figure 79) which, when scanned by the smartphone, trigger the appearance of superimposed AR content.

Figure 78 – Interaction with the NOVA zeppelin POI

Figure 79 – The elements which are recognised in the Layar Vision demonstration layer
All elements in Figure 79 were uploaded to the Layar publishing site in order to be eligible for recognition. Upon uploading, each of them was stored in the publishing site with a unique string identifier which was then used in our AR Management Site to associate each uploaded image to the POIs with the following characteristics:

- **POI 1** - An ISEGI presentation video (interaction steps detailed in Figure 80) accessible and placed at the front door of the building:
  - One image that is superimposed to the recognised image;
  - One action that triggers the opening of an ISEGI promotional video;

Figure 80 – POI 1 interaction steps – ISEGI presentation video
Chapter 5 - The AR demonstration layers

- POI 2, 3 and 4 – The ground, first, and second floor plant schemas (interaction steps detailed in Figure 81, Figure 82 and Figure 83), accessible and placed at the right of each floor entry:
  - Three ISEGI floor plant schemas superimpositions, one for each of the three floors affected by our demonstration layer;

Figure 81 – POI 2 interaction steps – Ground floor plant

Figure 82 – POI 3 interaction steps - First floor plant
POI 5 – Interacting with the personal web site of Professor Miguel de Castro Neto (interaction steps detailed in Figure 84), accessible via the adviser’s working office name tag in the second floor:

- One avatar of Professor Miguel de Castro Neto to be superimposed to the recognised image;
- One action which consists of a music track that is auto-triggered upon image recognition;
- One action which opens the thesis adviser’s personal web site, upon interaction with the avatar.
5.3. OUR FEEDBACK FROM TESTING THE SMARTPHONES IN THE CAMPUS

While we were testing and producing the AR demonstration worlds for the ISEGI-NOVA AR Project, we noticed relevant performance differences between the Samsung smartphones and the iPhone 4S, related with the aspects that involve tracking our position and updating it as we walk. It should be noted that we could not get an ideally clear sky during our on-campus testing phase but still, in the Samsung smartphones, it took minutes (rarely seconds) to lock our position coordinates, and as we walked around the campus, the repositioning of the POIs according to our walking movements was done each 80-100 meters. On the other hand, the iPhone 4S responded positively to locking our position for the first time (just a few seconds) and then updating the POIs each 5-10 meters as we walked.
We believe the difference in performance happens mostly due to the iPhone 4S using both GPS and Global Navigation Satellite System (GLONASS) satellites, which translates into a quicker locking position. GLONASS (Wikipedia 2012a) is a set of satellites, with full world coverage since October 2011, which can be used as an alternative or as an aid to the existing GPS satellites. While the fact of using additional GLONASS satellites is barely noticeable if we use the smartphones’ GPS in a car trip, it is more pronounced if we wish to track our walking steps. A Wikipedia section presents a list (Wikipedia 2012b) of smartphones which support GLONASS technology and, from observing some of the latest manufacturers’ smartphones like Apple, Samsung, HTC, Nokia and Sony Ericsson, to name a few, we can infer that the GLONASS technology will progressively be included in the upcoming smartphones, and this will possibly benefit the AR experiences.

Regarding the interaction made with the created AR worlds, the experience produced the expected interaction behaviour in both the low-end model (Samsung GIO) and the high-end models (Samsung Galaxy S II and Apple iPhone 4S):

- all POI images and 3D models were correctly displayed;
- all actions, either audio, video or website, were correctly triggered;
- all POI animations were displayed;
- all Layar Vision images were recognised and superimposed with the predefined AR content;
- the “language change” option produced the expected presentation of alternative AR content when it was applicable.
6. CONCLUSIONS

Augmented Reality, despite being available for a few years in the mobile phones’ application ecosystem, is still evolving in terms of public awareness and acceptance. This technology, which can be used for entertainment, educational, advertisement and spatial orientation purposes, just to name a few applications, is slowly benefiting from the growing tendency, on behalf of the public, to replace the simple mobile phones with the more advanced and feature-rich smartphones. Given the state-of-the-art of the available Augmented Reality browsers’ technology for smartphones, and the increase in the number of Augmented Reality content providers, we believe that the end-users have, by now, all the necessary means to engage in comprehensive Augmented Reality experiences.

The purpose of the ISEGI-NOVA AR Project was to take advantage of these favourable conditions in terms of available hardware, software and user maturity, by delivering enough artefacts that would allow the Universidade Nova de Lisboa to make a substantial statement, as far as Augmented Reality communication is concerned. We recognise that the Augmented Reality medium is unexplored and, through our project, we should not only provide technical means to bring Augmented Reality content to the smartphones, but should also facilitate the life cycle of Augmented Reality content creation. Therefore, we proposed a full Augmented Reality product, which would comprise:

a) The usage of a third-party Augmented Reality technology to display Augmented Reality content in a smartphone;
b) The development of a web service (AR Web Service) with the ability to respond to Augmented Reality content requests on behalf of the smartphones;
c) The development of an Augmented Reality intelligent tier (AR Libs), with the ability to produce a message that contains Augmented Reality worlds with all of the associated interaction experiences, and that is fully understood by the third-party Augmented Reality technology installed in the smartphone;
d) The development of a database schema to store all of the Augmented Reality content;
e) The development of a web site (AR Management Site) that allows the management of complete Augmented Reality experiences, through the use of accessible graphical user interfaces.
Chapter 6 - Conclusions

The starting point for the ISEGI-NOVA AR Project was to define a set of Augmented Reality requirements and expectations, to help in the process of selecting a third-party technology, installable in a smartphone, to provide Augmented Reality content representation capabilities. Once the requirements were test-proven against a selected batch of Augmented Reality technologies, we were able to define our development environment and the full set of technologies to be used in the project, including hardware and software.

We, then, under the project’s outputs scope, moved to introduce the full system architecture, relating the to-be-produced artefacts with the third-party Augmented Reality browser. In this context, we approached all of the outputs including:

- The web service’s architecture and expected behaviour;
- The AR Libs’ architecture, with full details about how the request and response messages are handled and produced, respectively, and the options taken to optimise performance;
- The Augmented Reality database schema as well as the binary file storage structure;
- The Augmented Reality management site, with full details about the site’s architecture, the priority of building simple interfaces; and the pages’ hierarchy, features, validation methods and level of code reuse.

All of these artefacts are part of our newly-developed Augmented Reality communication platform, which is hereby delivered to the Universidade Nova de Lisboa.

6.1. Contributions

About the Augmented Reality project, described in this thesis, we would like to emphasise the following contributions:

1. We developed a full working Augmented Reality product which includes a database repository to store Augmented Reality content, a web service with the ability to handle Augmented Reality requests and to produce an adequate content response to a prepared smartphone, and a web site that allows full Augmented Reality content management;
2. We created a new communication medium for the Universidade Nova de Lisboa. The university is now equipped with Augmented Reality means that allow, not only the full management of Augmented Reality content, but also bring forward the ability to quickly make the content publicly available;
3. We created two Augmented Reality worlds, located in the campus of the Universidade Nova de Lisboa, for demonstration purposes, which serve as a proof of concept and a basis for future content creation ideas.

During the making of the thesis, we also had the chance to collaborate with two other projects, one developed in a prototype form and the other made in the context of a public event:

1. We contributed to the creation of a prototype for an augmented-reality-fuelled greenhouse (Figure 85), based on a concept idea by the thesis’ adviser, Professor Miguel Neto. Our contribution involved the creation and storage of content in the AR Database, and we used our AR Web Service as a content provider. This collaboration helped to validate both the AR Web Service and the AR Libs tier. The proposed scenario exemplified an “intelligent” AR Greenhouse with the ability to dynamically manage wireless sensors data and showing, by each greenhouse item, the level of soil water, the temperature, the crop type, etc. In practice, the user would move around the greenhouse and get the aforementioned information by pointing the smartphone to the geo-referenced crop areas which were covered by the sensors. The information sources would combine the AR Database, which contained sensor-produced information (soil water level, temperature, etc.), and web-based information (pictures, crop description, etc.) about each area.

![Figure 85 - A prototype for a Greenhouse using Augmented Reality](image-url)
Chapter 6 - Conclusions

The AR Greenhouse prototype was then presented by Professor Miguel Neto in a conference, “I NOVAGRI International Meeting” (INOVAGRI 2012), that took place in Fortaleza, Brazil, from the 29th to the 31st of May of 2012, and received extremely positive feedback from the audience. The presented work received a proposal for inclusion in a book to be published by Embrapa (2012).

A few months later, the AR Greenhouse prototype was discussed in the “Soilless cultures and new technologies” symposium, through a presentation entitled “Supporting decisions using Augmented Reality”, organised in Torres Vedras, Portugal, on the 21st of November, again with very positive feedback for the attendants.

Based on the reception the AR Greenhouse obtained, ISEGI-NOVA fully acknowledges the potential of the concept and is keen on further developing it in the future.

2. We contributed to an Augmented Reality game (Figure 86) which served as a promotion element to a television series, and resulted from a joint effort between ISEGI-NOVA, FOX, and TMN. Our contribution involved hosting the Augmented Reality content and providing the AR Web Service. This collaboration helped to validate our AR Libs tier and also produced a “stress test” scenario consisting of ten users accessing simultaneously to several POIs with 3D models and web site actions.

This joint effort was possible after ISEGI-NOVA proposed a summer challenge, to high school students, inspired in the Walking Dead series, with the motto: “What could we do if Lisbon suffered a zombie attack?”. This proposal resulted in the creation of survival strategies and maps, which indicated the best shelter areas for humans as well as the areas more vulnerable to zombie attacks.
Making use of the produced geographic information, which concerned the areas more vulnerable to a zombie attack, an Augmented Reality game was created by ISEGI-NOVA and FOX, to be used as part of FOX’s advertisement campaign efforts to promote the third season of the Walking Dead TV series. The “Zombie Hunt” game’s purpose was to catch 2D animated rotating logos and 3D zombies which, when clicked, pointed to a website that recorded scores and rankings for the participating players. This game event, which was widely publicised on the media, took place in Lisbon’s historical centre area, on the 20th of October and was considered a success by all involved participants, who also recognised the key importance of ISEGI-NOVA AR Project to the event.
6.2. FUTURE WORK

Our Augmented Reality project produced several deliverables that represent an asset for ISEGI-NOVA and the Universidade Nova de Lisboa. We conclude with a list of suggestions at a software level, at an augmented-reality-content-creation level, and at a hardware level.

At a software level we acknowledge the following areas for future improvement:

1. AR Libs scope: for the request parameters mentioned in Table 9 (page 51), a study should be conducted, aligned with the ISEGI-NOVA content-creation strategy, to determine the utility of supporting additional parameters in future software releases. The study could result in the prospect of adding more filters and security options for the Augmented Reality worlds, while enhancing usability. Moreover, the specificity of an Augmented Reality response is highly correlated with the number of parameters used;

2. AR Libs scope: for the response elements mentioned as “not supported” in Figure 23 (page 55), a study should be conducted, aligned with the ISEGI-NOVA content-creation strategy, to determine the utility of supporting additional
response elements. Supporting more response elements results in increased possibilities to represent Augmented Reality content;

4. **AR Management Site scope**: besides the expected improvement suggestions that may arise after the site enters a production environment, we recognise benefits in evaluating the following items:
   a. **AR worlds in-document preview**: whenever a content provider creates an AR world, there could be the possibility to produce a one-click report (in PDF or other word format) that would list all of the Augmented Reality content, including details about all POIs, the interaction results, associated animations, displaying preview of all visual elements;
   b. **Multimedia on-site content preview**: currently, the only supported immediate media preview in the AR Management Site exists exclusively for image files, while video, audio and 3D models are displayed as a link. A study should evaluate the loss of performance versus the gain in content management effectiveness that could result from implementing small previews, embedded in the pages, for video, audio and 3D models;

At an augmented-content-creation level we present some suggestions (some might require changes in the AR Libs logic):

1. **Methodology**: create an AR content creation methodology that implements audio guides, before-and-after pictures, a sequential POI tour, and POI typology classification like:
   a. Library, university, bar, etc…;
   b. 2D, 3D, etc.;
   c. Computers, Classrooms, Wi-Fi spots, etc.;

2. **AR worlds suggestions** (eventually, some suggestions could be included in the same AR world):
   a. Create an AR world to provide an initiation tour, about the University campus’ services and locations, to the student community, namely the Erasmus students;
   b. Create a cultural and community-oriented AR world which shows the full history of Universidade Nova de Lisboa with complete tours, including audio, video, images, 3D models of the campus’ evolution since the beginning;
   c. Create an AR world that provides transportation tips: the best way to reach the campus, the means available, the approximate cost, etc.;
d. Create an AR game in a quiz-like manner that poses questions and reveals curious facts about the university, with the purpose of reaching out to the general community;

e. Organise AR events where the student community is reunited in the campus to perform a joint Augmented Reality experience;

f. Create an AR World to promote the list of universities and the available courses catalogue under the Universidade Nova de Lisboa sphere.

3. Hardware based suggestions:

a. Currently, the tablet computers (i.e. iOS or Android based) are emerging as another medium for mobile computing. The tablets use operating systems very similar to the smartphones, possess similar sensors, 3G, Wi-Fi and cameras, and are also supported by the Layar technology. Therefore, it is important that the tablets’ specific characteristics, like the bigger screen size and the increased processing capacity, are considered for the prospect of creating specific AR content.

b. In a further future, the advent of AR-enabled types of glasses, like Project Glass (Google 2012), for instance, which is a head-mounted display, can represent a lightweight and practical way of experiencing Augmented Reality content. This type of head-mounted display devices, which carries advanced processing capabilities and fits naturally above the eyesight, may represent another alternative for AR experiences and to enrich the list of hardware eligible for representing AR content produced by ISEGI-NOVA.

Figure 88 – Google’s Project Glass
7. REFERENCES


Wi-Fi-Alliance (2011) 'Wi-Fi Alliance', [online], available: http://www.wi-fi.org/ [accessed 2011-05-14].


