

# Prototyping of a Browser-Based Social N-Screen Platform

[Building Up to Boost User Experience]

Begoña Álvarez de la Cruz  
Computer Science MSc Thesis  
Web & Media Group  
Vrije Universiteit, Amsterdam  
begona.alvarezd@gmail.com

## ABSTRACT

A second screen is a hand-device which is susceptible to provide added value to the TV content consumption. Notube, with their web browser-based second screen application, moved further through this concept, creating an association between the second screen, Web and TV content. Nevertheless, the implementation still lacks completion in order to achieve a full service and its users' satisfaction.

This project shows the development of a social N-Screen prototype based on previous researches and implementations carried out by Notube. This platform is intended to be used by small groups to explore on-demand content. The main goal of the project is consisted on searching and implementing features to the platform in order to offer to the final user an improved user experience. This improvement is led by the features that allow a completion in the user interaction flow with the platform, such as the implementation of a registration and login, the provision of persistence to the user based content and the addition of new functionalities as personal lists and likes/dislikes tracking.

## 1. INTRODUCTION

### 1.1 Background

#### 1.1.1 The Multi-Screen World

The human being has become throughout the last years into a multi-screener<sup>1</sup> nation. From the appearance of television in our living rooms, until the incorporation of lighter and portable new devices such as smartphones or tablets, users have been including all these devices in their routines until turning them into everyday objects. Consequently, tablets, smartphones, televisions and computers have become the main group of devices with which an average user consumes most of their media content[17].

Despite each archetype provides a particular motivation and practise to users, an important fact is that screen devices as

<sup>1</sup>Multi-screening: use of more than one screen at a time.

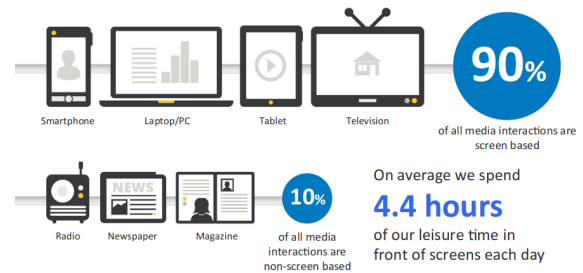


Figure 1: Study of daily media interaction, obtained from [17]

mentioned before are no longer used in isolation but collaboratively. Regarding this device collaboration, two different models of multi-screen[17] behaviour are distinguished: sequential and simultaneous. *Sequential usage* refers to moving through more than one device in order to achieve a task. *Simultaneous* concerns the usage of multiple devices at the same time for either related or unrelated activities. Both consumption forms are increasingly becoming the default mode and is surely influencing the way users engage.

Following this scenario comes the necessity to understand how users interact with these screen devices in combination[6]. The opportunity to decide which device to use, where and how makes possible for users to control their own interaction and content flow.

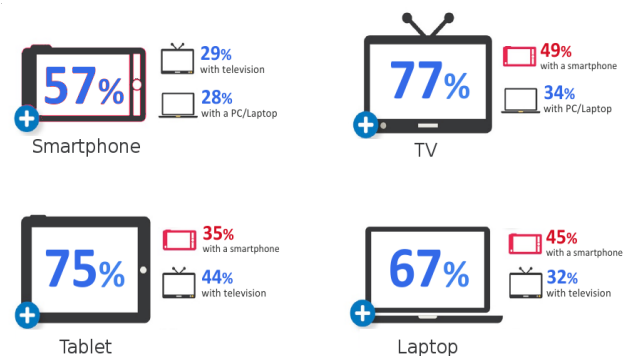


Figure 2: Study of companion devices during simultaneous usage, obtained from [17]

A Google study developed in 2012 [17] exposed how an average consumer makes use of companion devices - smartphone, TV, tablet and laptop - during simultaneous<sup>2</sup> usage - See Figure 2. There are three main multi-screen combinations:

- Smartphone + TV - 81%
- Smartphone + Laptop/PC - 66%
- Laptop/PC + TV - 66%

Additionally, a research developed by Microsoft in 2013 [9] illustrated how is the user behaviour while multi-screening in simultaneous usage.

- 68% of consumers interact with multiple devices at the same time to access unrelated content; e.g. they may be texting a friend while watching TV.
- 57% of consumers make use of more than one device simultaneously in order to achieve a related activity.

From now on, we will focus our attention in simultaneous usage for *related* activities.

### 1.1.2 Second Screens

One of our everyday routines that has been altered by this new screen multitasking<sup>3</sup> behaviour is that moment while a user watches TV. Viewers no longer focus their entire attention to the TV screen but share it with portable devices.

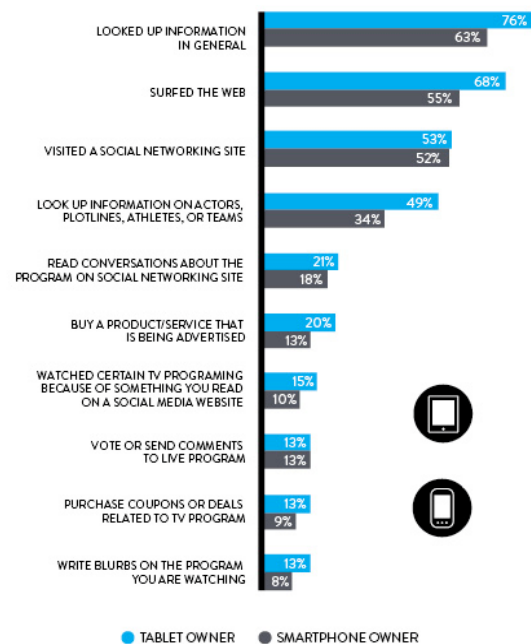
Figure 3 illustrates how consumers make use of their tablets and smartphones while watching TV. We can observe how indeed users not only surf on the web, but they interact with the device with activities directly related to the program or advertisement that they are watching at that moment. This demonstrates the fact that consumers are not merely interacting with their hand devices as a simple distraction, but sometimes in order to improve their TV content consumption.

This new practice has led to the creation of the new concept *second screen*. Second screen is a hand-device which is susceptible to provide added value to the TV content consumption. These devices such as tablets or smartphones play a role as companion screens that 'connect' viewers to complementary interaction opportunities while they watch TV via applications, additional show-oriented content or in-synch functionalities [5].

This new activity has become such important that a survey developed by Nielsen Holdings N.V. [11] reported 'Using a tablet or smartphone while watching TV is more common than not'. Nearly half of tablet owners - 43%- and smartphone owners - 46% - declared that while watching TV they are making use of their devices as second screen every day.

<sup>2</sup>Usage for either related or unrelated activities.

<sup>3</sup>*Human multitasking* is the apparent performance by an individual of handling more than one task at the same time.



**Figure 3: Tablet or smartphone activities while watching TV, obtained from [11]**

As a consequence of this fact, there are emerging new apps<sup>4</sup> which take advantage offering second screen experiences that can improve even more this interactivity while watching TV.

Second screen apps [4] are intended to enable viewers to interact before, during and after the broadcast of a programme using a laptop, smartphone or tablet. The most competent apps, instead of distract, have the potential to increase the viewers' attention and enjoyment on the watching programme. According to [4], eight types are distinguished in order to categorize these apps based on their functionalities:

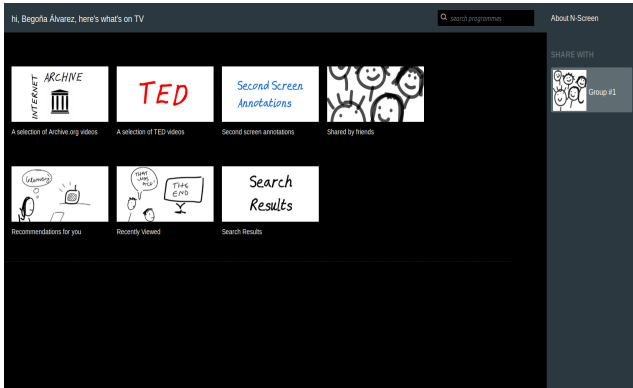
- Socializing
- Loyalty
- Recommendation
- Transaction
- Information
- Program guides
- Participation
- Creation

On that account, second screen apps arised as a manner to unlock new research and business models due to the wide range of potential possibilities they offer.

<sup>4</sup>*App* is an abbreviation for application. An app is a piece of software. It can run on the Internet, on a computer, on a phone or other electronic device.

### 1.1.3 Use case

This project is based on an already existing platform developed by **Notube**[2]. Notube, is a project funded on 2009 that was specialized in second screens and Web merging. Their motivation was getting the Web and TV closer together via shared data models and content across multiple devices. They developed their own N-Screen<sup>5</sup> prototype, a Web browser-based<sup>6</sup> second screen application<sup>7</sup> for small group exploration of on-demand content, both in the same room and remotely, with each individual having their own second screen device. It integrates different combinations of recommendation strategies and allows to decide within a closed group to watch a selected program sharing a 'virtual television', that in their case is the hand device itself.



**Figure 4: Notube N-Screen screenshot, source code obtained from the public repository <https://github.com/notube/n-screen>**

It is developed following a *user-centric approach*<sup>8</sup> in order to explore main aspects of users' content customisation demands, interaction requirements and entertainment preferences. Their main goal was to investigate if making decisions in collaboration might guide people to find something interesting to watch.

## 1.2 Goals

The main goal of the project is to re-design, implement and improve the functionality, interactivity and user experience of the browser-based second screen recommender platform<sup>9</sup> carried out by Notube, in order to provide an attractive functional platform that can be used to graphically test different recommendation strategies.

<sup>5</sup> N-Screen because it might be the primary screen, or one of a bunch of equals.

<sup>6</sup> Browser-based refers to computer tools and applications which run on a web browser via the Internet without accessing the operating system of any individual computer. These applications are accessed through web pages and can be used by people who are prevented from downloading software applications by firewalls.

<sup>7</sup> All the source files used to generate this application can be accessed at the public repository: <https://github.com/notube/n-screen>

<sup>8</sup> Placing user's goals at the heart of development.

<sup>9</sup> A platform is a group of technologies that are used as a base upon which other applications, processes or technologies are developed.

Most of the second screen apps in the market are not successfully developed because they are focused on enhance activities that do not fulfil the real consumer needs[4]. Consequently, the audience engagement fails. This fact is crucial and it must be taken into account in order to make the platform valuable.

In addition, the platform must be design taking into account a possible future recommendation system integration. Furthermore, the project must accomplish certain requirements for a potential release to final users. It needs to be developed offering user-based content, due to its recommendation feature. It is required to provide engaging content and interaction activities, as well as offering an intuitive appearance. Moreover, the platform must be flexible and scalable, so it will not need to be entirely redone with every particular change. Hence, it is needed to reach a complete solution covering all these features, overcoming the gaps and deficiencies that other platforms show.

Here, the main goal is fragmented into more specific sub-goals:

- *Analysis of previous conclusions and results.* Study of already tested aspects. Definition of new tests to provide new data of interest. These data will help in making decisions about the implementation of interaction activities.
- *Requirements definition.* Study of needs and constraints.
- *Software design.* Global design of the new involved software. Debugging and optimization.
- *Platform development.* Implementation and integration in order to achieve the development of the final demo.
- *General purpose tests.* Tests oriented to prove the proper operation and verify the achievement of the requirements and constraints compliance.
- *Results analysis and conclusions.* Achievement evaluation. Study of weaknesses or possible improvements. Definition of further studies.

## 1.3 Project Organization

The organization of this project is described as follows:

- *Analysis of previous conclusions and results.*

At this first stage, N-Screen-related information and specific knowledge was acquired. An approach to the development tools was also outlined. Goals:

- Second screen state-of-the-art review. Review of the current commercial solutions.
- Adaptation to the development of the platform as well as required tools (repository, client and server side programming languages research and learning, etc...).
- Analysis of related projects results, either completed or under development.

- *Software design.*

This phase covered from the first software definitions and specifications, to the platform implementation until reaching a final demo. Goals:

- Software design and implementation for required features.
- Content dataset migration.
- *Alpha*-version deployment. Source code debugging and improvement.
- *Beta*-version deployment. Source code debugging and improvement.

- *Tests and evaluation.*

At last, the final demo was evaluated and the results were analyzed. Goals:

- Fully integrated back-end and front-end test.
- Interaction test.
- Graphical user interface test.
- User experience test.
- Results interpretation and conclusions review. Statement of further studies and future development lines. Found problems evaluation.

- *Documentation generation.*

Dissertation and other required documentation writing.

## 1.4 Outline

Typically, the body of a paper is organized into a hierarchical structure, with numbered or unnumbered headings for sections, subsections, sub-subsections, and even smaller sections. The command `\section` that precedes this paragraph is part of such a hierarchy.<sup>10</sup> L<sup>A</sup>T<sub>E</sub>X handles the numbering and placement of these headings for you, when you use the appropriate heading commands around the titles of the headings. If you want a sub-subsection or smaller part to be unnumbered in your output, simply append an asterisk to the command name. Examples of both numbered and unnumbered headings will appear throughout the balance of this sample document.

## 2. STATE-OF-ART

Typically, the body of a paper is organized into a hierarchical structure, with numbered or unnumbered headings for sections, subsections, sub-subsections, and even smaller sections. The command `\section` that precedes this paragraph is part of such a hierarchy.<sup>11</sup> L<sup>A</sup>T<sub>E</sub>X handles the numbering and placement of these headings for you, when you use the appropriate heading commands around the titles of the headings. If you want a sub-subsection or smaller part to be

<sup>10</sup>This is the second footnote. It starts a series of three footnotes that add nothing informational, but just give an idea of how footnotes work and look. It is a wordy one, just so you see how a longish one plays out.

<sup>11</sup>This is the second footnote. It starts a series of three footnotes that add nothing informational, but just give an idea of how footnotes work and look. It is a wordy one, just so you see how a longish one plays out.

unnumbered in your output, simply append an asterisk to the command name. Examples of both numbered and unnumbered headings will appear throughout the balance of this sample document.

## 3. REVIEW STUDY

A model for the platform is defined in this chapter. Initially, the first version of the N-Screen developed by Notube[2] is evaluated to obtain some guidelines and potential improvements facing our own. The final software implementation requirements and constraints are defined. Finally, a discussion and evaluation over different features is carried out, exposing different options and some final conclusions in order to face a potential deployment to final users.

### 3.1 Browser-Based Social N-Screen Platform Model

Essentially, the BBSNSP<sup>12</sup> responds to a second screen application based on a recommendation system. However, a further review shows some general differences:

- *Browser-based*: instead of being developed as a mobile or computer app, the BBSNS is a web app. A *web app* refers to software that runs on a web browser. This way, not only the app can be updated and maintained without disturbing potential users by requiring them to re-download. Additionally, it provides implicit support for cross-platform<sup>13</sup> compatibility.
- *Group decision recommendation system -Social-*. The recommendation feature is focused on 'how collaborating together might help people find something interesting to watch'<sup>14</sup>. It is designed to be used within a small group of friends in a collaboratively way to reach together a successful programme to watch.
- *Apart<sup>15</sup> group oriented*: this BBSNS is mainly designed to watch together with other friends but being remotely located.
- *N-Screen*: It is not only oriented to be used as a secondary. It is considered that it might be used as a primary screen, secondary screen, or one of a screen devices collection.

### 3.2 First Browser-Based Social N-Screen Platform Review

The first BBSNSP was a project of Notube developed in 2011. It supposed a big step forward to get the Web and TV closer together using shared data models and content across multiple devices[14]. It is designed to help deciding and enabling to interact using drag and drop over screen devices. It allowed to investigate how helpful is group collaboration in order to find an interesting programme to watch, for limited

<sup>12</sup>Browser-Based Social N-Screen Platform

<sup>13</sup>*Cross-platform* regards the capability of a software to run identically on different platforms.

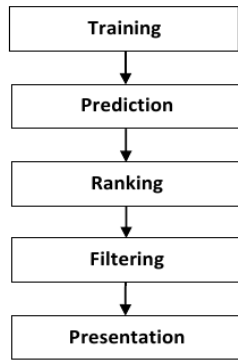
<sup>14</sup><https://notube3.wordpress.com/2011/10/10/n-screen-a-second-screen-application-for-small-group-exploration-of-on-demand-content/>

<sup>15</sup>Different physical locations.

group of users. It gave the strengths and weaknesses to establish the fundamentals for future developments.

While constituting a successful platform meeting most of its features, the Notube BBSNSP introduced some issues found after its evaluation. This issues are following listed.

I) As a recommender system, the platform is intended to be user based content oriented. User preferences and explicit interaction should provide to the recommendation strategies an on-real-time update to re-rank the displayed personal suggestions[10]. In Notube's platform, user interaction is treated as *volatile* data, each time that the session is closed, the information is lost. This implementation impedes to exploit all this relevant information. Figure 5 shows the 5 tasks that are intended to be accomplished by the recommendation engines in the client-side. If the platform lacks tracking the user interaction and preferences, it would not be possible to realize properly the first *training* step due to a shortcoming information.



**Figure 5: Recommender system tasks, obtained from [10]**

II) Their own preliminary findings after the evaluation<sup>16</sup> showed us a weak point. Notube results obtained after testing N-Screen with real users showed how some of them received a successful recommendation and indeed were wishing to watch later that specific selected programme. They received feedback such as "Not for watching something instantly, only for making suggestions for things that could choose to watch later if you wanted to" or "I do not think it is on to recommend things for others to watch instantly". This results revealed an interesting *sequential usage* behaviour. In this case, instead of passing from one device to another, the user is making use of the same device time later to accomplish a task. Therefore, not only the volatile activity of the user, but also a lack of option to save wished programmes for a future moment, makes unlikely the activity to watch *that programme later*.

III) It is very important to offer an attractive set of possible activities for the user to interact with the [16]. The only implicit offered interactions are the browsing and the suggestion part. For this reason, we found that Notube N-Screen slightly incomplete in terms of interactivity actions.

<sup>16</sup><https://notube3.wordpress.com/2011/12/12/preliminary-findings-of-n-screen-user-testing/>

IV) The platform was designed to be not only a recommendation system, but also a place to watch a selected programme. There exist two different ways to realize this action: locally and remotely. The first one is performed when a user decides to watch a programme individually. By contrast, the second one is conducted when a group decides to watch together a selected programme and the members are located in remote locations. This last visualization mode involves the execution of a browser-based 'virtual TV' in which every person within the group can watch simultaneously the same. Unfortunately, this last feature was not working because the programme visualization part was missing. Making the virtual TV feature working is indeed necessary for a complete success of the platform.

V) As mentioned before in this section, issue I), the platform is designed to provide -future- recommendation strategies. For this purpose, its structure design needs to be flexible and scalable to allow future recommendation scripts compatible in terms of data, in this case, a wide set of programmes. Following a deep study of the source code, we found an important bottleneck. The implementation was not flexible to face data content migrations. The programmes displayed, and all the logic behind it, were designed to correctly work just locally and with a limited flexibility range. It is not possible to afford heavy data as a wide set of programmes running locally due to memory limitations.

VI) The inherit second screen behaviour of the platform makes strongly important to take into account that higher complex interaction in this type of environments is a determinative factor [3], since involves a multi-tasking activity. We found that the welcome page was not simple and informative enough in order to not distract the user from the main screen attention. The welcome page lacked first sight relevant information, therefore it required more attention than actually needed.

## 4. IMPLEMENTATION STUDY

Once the first BBSNSP has been reviewed, it is moment to study how to settle every found issue and bottleneck exposed in Section 3.2 and start the implementation. The decisions regarding the implementation of new features are exposed.

### 4.0.1 Localhost Deployment

Before being able to start programming the platform, it was required preparing the environment.

On the one hand, it was needed to set up a local server to be able to run and test our web application. We used XAMPP on a Linux distribution for this purpose, due to its multiple advantages and intuitiveness. XAMPP is an open source independent server platform which consists mainly of a MySQL database, Apache Web server and required interpreters for scripting languages as PHP and Perl.

On the other hand, the set-up required an Ejabberd installation to manage instant messages in order to enable automatic non-typed communication between remote users and allow them to collaborate -suggest-. Ejabberd is a Jab-

ber/XMPP<sup>17</sup> server [7] which is realized in Erlang<sup>18</sup> language. Its main functionality is that it works as an instant messaging server that allows more than one people to communicate and participate in real time, based on typed text or not. Additionally, considering that is intended to be used by a web application as part of a communication tool, it was necessary to enable BOSH in our Ejabberd server configuration. HTTP is synchronous providing simple call and response methods, discordant to XMPP's asynchronous event based protocol. Therefore, BOSH<sup>19</sup> is a protocol that provided us a method to use XMPP in our platform by making http requests with a long time-out.

After this preparation, the environment was therefore ready to start the implementation.

## 4.1 Registration & Login

The platform is intended to be used to test researches developed about recommendation strategies, therefore, it follows a user content based model. As explained in Section 3.2 issue I), user preferences and explicit interaction should provide to the recommendation strategies an on-real-time update to re-rank the displayed personal suggestions. In order to deal with the volatile data it was decided to implement a Registration & Login system. This way, it is possible to tack user preferences and provide relevant data to the recommendation side.

An important step before implementing the registration system was to study different types of registration which we could offer. Due to the imminent growth of the social networks popularity such as Facebook<sup>20</sup> [15], websites are currently offering two different options to accomplish a registration:

- *Good registration*: known as the one where a user fills in directly the registration form with their personal data.
- *OpenID*: OpenID Registration is an extension to the OpenID<sup>21</sup> Authentication protocol that allows for very light-weight profile (mostly social networks profile, as Facebook or Twitter<sup>22</sup>) exchange . It is designed to pass eight commonly requested pieces of information when an end user goes to register a new account with a web service.

The OpenID provides an even faster registration than before. The '1-click' activity can definitely be very attractive

<sup>17</sup> *Extensible Messaging and Presence Protocol*, better known as XMPP (formerly Jabber), is an open and extensible XML-based protocol originally designed for instant messaging.

<sup>18</sup> Therefore, it was required also the Erlang packages installation for compilation

<sup>19</sup> Bidirectional-streams Over Synchronous HTTP

<sup>20</sup> <https://www.facebook.com/>

<sup>21</sup> OpenID is an interoperable authentication protocol designed to be safe, faster and easier way to log in to web sites

<sup>22</sup> <https://www.twitter.com/>

to implement as unique registration for a second screen platform[1], where users distraction has a crucial importance. But despite its significance through the last years, not every user is willing to share their social networks' personal profile due to a lack of confidence in such websites[13]. For this reason, we decided to include both in our platform, selecting the OpenID registration provided by Facebook, since this social network can offer valuable information for the recommendation system.

The implementation for registration & login has been carried out making use of:

- *MySQL database*
- *Database handler scripts*:
  - HTML5 submit forms & PHP for the good registration
  - Javascript Facebook SDK & PHP for the OpenID registration

### 4.1.1 Database

The structure of our database has been changing as the platform evolved. At the end, intending to match every needed requirement, we reached the following schema based on two tables:

Table **members**. Column fields:

- **member\_id**: Primary key<sup>23</sup>
- **firstname**: First Name
- **lastname**: Last Name
- **login**: Unique user-name
- **passwd**: MD5 encrypted<sup>24</sup>
- **facebook\_id**: unique ID extracted from Facebook OpenID.

Table **content**. Column fields:

- **member\_id**: Primary key<sup>25</sup>
- **recommendations**: Personal recommendations list, JSON<sup>26</sup> format.
- **recently\_viewed**: Recently viewed or watched programmes personal list, JSON format.

<sup>23</sup> The *primary key* of a relational table uniquely identifies each record in the table.

<sup>24</sup> The *MD5* message-digest algorithm is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, typically expressed in text format as a 32 digit hexadecimal number

<sup>25</sup> The *primary key* of a relational table uniquely identifies each record in the table.

<sup>26</sup> *JSON* (JavaScript Object Notation) is a lightweight format that is used for data interchanging

- **watch\_later:** Watch Later personal list, JSON format.
- **like\_dislike:** List containing personal likes and dislikes of the user, JSON format.
- **shared\_by\_friends:** List containing suggestions to the user made by members within his/her same group.

For further details in terms of database fields definition, the following code (mysql.sql) shows in SQL code shows the tables initialization:

```
1 CREATE TABLE IF NOT EXISTS 'members' (
2   'member_id' int(11)
3     unsigned NOT NULL AUTO_INCREMENT,
4   'firstname' varchar(100) DEFAULT NULL,
5   'lastname' varchar(100) DEFAULT NULL,
6   'login' varchar(100) NOT NULL DEFAULT '',
7   'passwd' varchar(32) NOT NULL DEFAULT '',
8   'facebook_id' bigint(11) DEFAULT NULL,
9   PRIMARY KEY ('member_id')
10 ) ENGINE=MyISAM
11   DEFAULT CHARSET=latin1 AUTO_INCREMENT=
12   1 ;
13 CREATE TABLE IF NOT EXISTS 'content' (
14   'member_id' int(11)
15     NOT NULL AUTO_INCREMENT,
16   'recommendations' longtext NOT NULL,
17   'recently_viewed' longtext NOT NULL,
18   'watch_later' longtext NOT NULL,
19   'like_dislike' longtext NOT NULL,
20   'shared_by_friends' longtext NOT NULL,
21   PRIMARY KEY ('member_id')
22 ) ENGINE=InnoDB
23   DEFAULT CHARSET=latin1 AUTO_INCREMENT=
24   1 ;
```

Picture whatever shows how the content is initialize for each user the very first moment during the registration.

\*\*\*\*INCLUDE AN IMAGE OF THE DATABASE STRUCTURE BEING POSTED BY REGISTRATION\*\*\*\*\*

#### 4.1.2 Login Security

An important implementation to mention is that, in order to access to the personal webpage after the login action, a security checking has been implemented. The access to the welcome page containing personal user-based content is only possible if the login has been correctly done, case in which the authentication SESSION VARIABLES<sup>27</sup> are set and authenticated.

The following script (auth.php) shows this implementation:<sup>28</sup>

```
1 <?php
2 //Start session
```

<sup>27</sup>A Session Variable is an associative array containing session variables available to the current script.

<sup>28</sup>SESS\_MEMBER\_ID is only initialized when the login authentication had been successful.

```
3 session_start();
4
5 //Check whether the session variable
6 //SESS_MEMBER_ID is present or not
7 if(!isset($_SESSION['SESS_MEMBER_ID'])
8    || (trim($_SESSION['SESS_MEMBER_ID'])
9        == '')){
10    header("location:index.html");
11    exit();
12 }
13 ?>
```

In addition, for localhost testing purposes, is been included a management of the browser cookies<sup>29</sup> to allow more than one sessions running simultaneously in the same browser.

## 4.2 Interaction Activities

### 4.2.1 Watch Later

It is severely important to understand how is the user behaviour when interacting with a web service. Developers and designers work together to deploy a final product, but sometimes users can present unpredictable conduct and wishes[12]. It is only after realizing tests on final users when the success or failure can be declared. Notube N-Screen, for example, was designed to be a second screen recommendation platform to watch immediately a selected programme. But after a evaluation with real users they found out how some users where wishing to watch that successful recommended programme not immediately, but in another later moment. This result revealed a very interesting *sequential usage* behaviour in which users utilize the same platform -and sometimes the same device- but in separated moments to accomplish a task, in this case, watch the desired programme. The platform therefore showed a general successful feedback, but was lacking an important user conduct that was not taking into account at the beginning.

Our simplest and effective solution to this situation[8] has been the implementation of the possibility to add programmes to a personal 'watch later' list. This list is directly populated with the implicit user interaction of clicking a 'watch later' button. The list can be edited by the addition or removal of a programme. It is important to mention that each time the user edits this personal list, it is directly updated in the database making use of the PHP script `set_channel.php`, which source code is as follows:

```
1 <?php
2 //Start session
3 session_start();
4 $data = mysql_escape_string($_POST['data',
5   ]);
6 $channel = $_POST['channel'];
7 ini_set( 'default_charset', 'UTF-8' );
8
9 //Include database connection details
10 require_once('config.php');
11
12 //Connect to mysql server
```

<sup>29</sup>A cookie is a small piece of data sent from a website and stored in a user's web browser while the user is browsing that website.



```

12 $link = mysql_connect(DB_HOST, DB_USER,
13     DB_PASSWORD);
14 if(!$link) {
15     die('Failed to connect to server:'.
16         mysql_error());
17 }
18 //Selecting database for the user
19 $db = mysql_select_db(DB_DATABASE);
20 if(!$db) {
21     die('Unable to select database');
22 }
23 $member_id = $_SESSION['SESS_MEMBER_ID'];
24 mysql_query('UPDATE content SET $channel
25     =
26     '$data' WHERE member_id = '$member_id
27     ');
28 ?>

```

\*\*\*\*INCLUDE SCREENSHOT WATCH LATER \*\*\*\*

### 4.2.2 Like & Dislike

As mentioned in Section 3.2 issue III), we analysed Notube N-Screen and we found that the platform was missing interaction activities. The implementation of the Watch Later list explained in Section 4.2.1 is actually an addition that improves this issue. Nevertheless, after a deep study we came to the result that just adding that feature was not enough. Based on the recommendation system that is going to be included afterwards, we converged to an idea to include the possibility to elicit user preferences by ranking a programme with a simple Like or Dislike action.

The idea came from the project developed by Vista-TV Sibyl<sup>30</sup>. Sibyl is a TV and radio programme recommender system designed for tablets and personal computers which uses a novel drag-and-drop system to extract user preferences. A user is able to express preferences by dragging individual programmes into 'like' and 'dislike' boxes. These preferences are immediately used by the client-side recommender to re-rank the programmes and refresh the recommendation list. Therefore, the implementation in N-Screen not only improves the user interaction, but also can be used to improve the recommendation strategy using directly relevant information provided by the user.

As well as with the case of the Watch Later list, likes and dislikes are directly populated with the implicit user interaction of clicking a button; simultaneously updating the database with PHP script `set_channel.php`.

\*\*\*\*PONER IMAGEN DE LIKES AND DISLIKES\*\*\*\*

### 4.2.3 Hyperlink Metadata

In order to keep improving the platform without distracting the user from the main purpose[1] that is watch a programme, we decided to implement another more additional

feature: Hyperlink<sup>31</sup> Metadata<sup>32</sup>. Inside each selected programme description, we included the possibility to click hyperlink metadata to browse further information of this selected concept. The first idea was to add the possibility to include clickable actions to actor, director, genre and general metadata of a TV programme. Unfortunately, it has been only possible to implement the hyperlink with 'tags' due to content dataset restrictions - data explanation in Section 4.4-. For that reason it has been possible only to display only a small part of the goal, but the implementation is developed to work successfully with other datasets that provide wider content extraction.

\*\*\*\*PONER IMAGEN DE POSIBILIDAD DE CLICKAR METADATA\*\*\*\*

## 4.3 Remote TV

Social activities through the Web are becoming exponentially popular since the web encourages users to participate socially active without even moving from their rooms[14]. On the other hand, TV remains a largely passive experience that usually requires physical presence to become social in terms of simultaneous experience sharing, i.e. if you want to watch the same programme at the same time with a friend, you usually meet together in order to do it. For this reason, one of the most innovative and attractive features included in Notube N-Screen was the possibility to share a cross-platform 'virtual TV' in which every person within a group can watch simultaneously the same. Watching TV with faraway friends through a virtual living room.

Unfortunately, as explained in 3.2 issue IV), this last feature was not working because the programme visualization part was missing. We decided that it was necessary to resuscitate the virtual television in order to achieve a successful platform. After long time studying the source code `player.html`, the bug was found. The problem was mainly residing in data compatibility and how `player.html` handled this data to enable a visualization. Further details concerning data structure are explained in section 4.4. The following `player.html` section of code shows how we handled the issue, replacing the local '*manifest*'<sup>33</sup> extraction for a `http-request` in order to retrieve the manifest, and therefore, fixing the problem.

```

1 $(document).bind('tv_changed', function (e
2     ,item) {
3     console.log(item);
4     var programme = item.nowp;
5     var id = item.nowp.id;
6     me.nowp = item.nowp;
7     $("#title").html(programme["title"]);
8     var action = "Play";
9     if(programme && programme["action"]){
10         action = programme["action"];

```

<sup>31</sup>Hyperlink is a reference to data that the reader can directly follow either by clicking or by hovering or that is followed automatically.

<sup>32</sup>Metadata: data about data. It is descriptive information about a particular data set, object, or resource.

<sup>33</sup>The *manifest* provides relevant metadata for a specific programme such as video-url and video-format

<sup>30</sup><http://sibyl.prototyping.bbc.co.uk/>



```

11 }
12 show_message(action+'ing'+programme["
    title"]);
13
14 if(action=="Play"){
15
16 $.ajax({
17     url: "get_tedtalks_by_id.php",
18     type: "POST",
19     async: false,
20     data: {id: id},
21     dataType: "json",
22     success: function (data) {
23         item = changeData(data);
24         //JSON with suggestions format
25         var manifest = item.suggestions[0].
            manifest;
26         process_manifest(manifest,programme)
            ;
27     }
28 });
29 // ---- PREVIOUS CODE ----
30 //pretty much everything should have a
    manifest
31 // var manifest = programme["manifest"
    ];
32 // var manifest = item.manifest;
33 // if(manifest){
34 //     console.log("manifest is "+
        manifest);
35
36 //     $.ajax({
37 //         url: manifest,
38 //         dataType: "json",
39 //         success: function(data){
40 //             process_manifest(data,
                programme);
41 //         }
42 //     });
43 }else{
44     alert('no manifest');
45 }
46 });

```

\*\*\*\*\*PONER IMAGENES QUE DEMUESTRAN EL FUNCIONAMIENTO DE LA REMOTE\*\*\*\*

## 4.4 Content Data

This project presents a functional platform that can be used, between others, to test future recommendation strategies. One of the possibilities that are currently being researched in our Web & Media Department <sup>34</sup> is a recommendation system based on BBC<sup>35</sup> programmes through the webpage <http://www.bbc.co.uk/programmes/>. As mentioned in section 3.2 issue V), the Notube N-Screen structure was designed to work with locally stored data. This approach was inefficient since facing future data migrations or including heavier data would follow not only memory but also

<sup>34</sup><http://wm.cs.vu.nl/>

<sup>35</sup>The *British Broadcasting Corporation* (BBC) is a UK-based international public-service broadcaster headquartered at Broadcasting House in London.

functional issues. Consequently, we decided to change how the platform deals with this. We converted it to a platform structure designed to deal with remotely stored data. In order to make this happen, a set of scripts to handle `http-requests`<sup>36</sup> has been implemented. This way, it is not needed to locally store the testing dataset but to provide to the scripts a suitable URL to extract information. As a result, the functionality speed is maintained at the same time the platform total weight is severely reduced since it does not locally store any dataset.

In addition, it has been taking into account the possible data structure that the recommendation system may use as input. For this purpose, a deep study of data structure compatibility has been conducted to facilitate as much a possible future implementations carried out by recommendation engines researchers. Based on various data extraction APIs<sup>37</sup> such as the *BBC Developers API*<sup>38</sup> or the *TED Talks Lab API*<sup>39</sup> we decided to set up the following structure for every object contained in our N-Screen, as programmes or videos<sup>40</sup>:

```

1 {
2   "pid":1000,
3   "title":"Gero Miesenboeck: Re-engineering
    the brain",
4   "description":"In the quest to map the
    brain, many scientists have attempted
    the incredibly daunting task of
    recording the activity of each neuron.
    Gero Miesenboeck works backward --
    manipulating specific neurons to
    figure out exactly what they do,
    through a series of stunning
    experiments that reengineer the way
    fruit flies perceive light.",
5   "date_time":"2010-11-03 22:44:00",
6   "url":"http://download.ted.com/talks/
    GeroMiesenboeck_2010G-950k.mp4",
7   "video":"http://download.ted.com/talks/
    GeroMiesenboeck_2010G-950k.mp4",
8   "speaker":[
9     {
10      "speaker":{
11        "id":741,
12        "title":"",
13        "firstname":"Gero",
14        "middleinitial":"",
15        "lastname":"Miesenboeck",
16        "description":"Optogeneticist",
17        "whotheyare":"Using light and a little
            genetic engineering -- optogenetics
            -- Gero Miesenboeck has developed
            a way to control how living nerve
            cells work, and advanced

```

<sup>36</sup>*HTTP request/response* protocol, which means a client-side application sends a request for some file, and the web server sends back a response.

<sup>37</sup>An *Application-Programming Interface* (API) is a set of programming instructions and standards for accessing a Web-based software application or Web tool.

<sup>38</sup><https://developer.bbc.co.uk/>

<sup>39</sup><http://developer.ted.com/APIDocs>

<sup>40</sup>Notice that it is a random example using a extracted TED Talks Video.

```

understanding of how the brain
controls behavior.",
18 "whylisten":"<p>Gero Miesenboeck is
pioneering the field of
optogenetics: genetically modifying
nerve cells to respond to light.
By flashing light at a modified
neuron in a living nervous system,
Miesenboeck and his collaborators
can mimic a brain impulse -- and
then study what happens next.
Optogenetics will allow ever more
precise experiments on living
brains, allowing us to gather
better evidence on how electrical
impulses on tissue translate into
actual behavior and thoughts...</p
>",
19 "slug":"gero_miesenboeck",
20 "published_at":"2010-06-09 08:14:00",
21 "updated_at":"2010-11-04 15:11:51"
22 }
23 }
24 ],
25 "image":"http://images.ted.com/images/ted/
51f652b9ff6854867d1d7abb2683caf1d8dd22
fb_240x180.jpg",
26 "manifest":{
27 "pid":1000,
28 "id":1000,
29 "title":"Gero Miesenboeck: Re-engineering
the brain",
30 "image":"http://images.ted.com/images/ted/
51f652b9ff6854867d1d7abb2683caf1d8dd22
fb_240x180.jpg",
31 "provider":"ted",
32 "duration":1750,
33 "media":{
34 "mp4":{
35 "uri":"http://download.ted.com/talks/
GeroMiesenboeck_2010G-950k.mp4",
36 "is_live":"false"
37 }
38 },
39 "type":"video/mp4"
40 },
41 "tags":{
42 "biology":"biology",
43 "brain":"brain",
44 "neurology":"neurology",
45 "science":"science"
46 }
47 },

```

It is important to mention that in order to present an attractive platform to interact with, a complete data migration has been implemented. Due to its

## 4.5 Web Design

### 4.5.1 Welcome Page

### 4.5.2 Selected Programme

### 4.5.3 Call-to-action buttons

## 5. EVALUATION AND RESULTS

Typically, the body of a paper is organized into a hierarchical structure, with numbered or unnumbered headings for sections, subsections, sub-subsections, and even smaller sections. The command `\section` that precedes this paragraph is part of such a hierarchy.<sup>41</sup>  $\text{\LaTeX}$  handles the numbering and placement of these headings for you, when you use the appropriate heading commands around the titles of the headings. If you want a sub-subsection or smaller part to be unnumbered in your output, simply append an asterisk to the command name. Examples of both numbered and unnumbered headings will appear throughout the balance of this sample document.

## 6. CONCLUSIONS

This paragraph will end the body of this sample document. Remember that you might still have Acknowledgments or Appendices; brief samples of these follow. There is still the Bibliography to deal with; and we will make a disclaimer about that here: with the exception of the reference to the  $\text{\LaTeX}$  book, the citations in this paper are to articles which have nothing to do with the present subject and are used as examples only.

## 7. ACKNOWLEDGMENTS

This section is optional; it is a location for you to acknowledge grants, funding, editing assistance and what have you. In the present case, for example, the authors would like to thank Gerald Murray of ACM for his help in codifying this *Author's Guide* and the `.cls` and `.tex` files that it describes.

## 8. REFERENCES

- [1] J. Allen and J. Chudley. *Smashing UX design: Foundations for designing online user experiences*, volume 34. John Wiley & Sons, 2012.
- [2] L. Aroyo, A. Conconi, S. Dietze, A. Kaptein, L. Nixon, C. Nufer, D. Palmisano, L. Vignaroli, and M. Yankova. Notube-making tv a medium for personalized interaction. 2009.
- [3] L. Cruickshank, E. Tseklevs, R. Whitham, A. Hill, and K. Kondo. Making interactive tv easier to use: Interface design for a second screen approach. *The Design Journal*, 10(3):41–53, 2007.
- [4] Evolumedia Group & Canadian Media Fund. The second screen and television. white paper series number 1. overview and growth perspectives. October 2012.
- [5] Evolumedia Group & Canadian Media Fund. The second screen and television. white paper series number 2. benefits & impacts. March 2013.
- [6] N. Hritzuk and K. Jones. *Multiscreen Marketing: The Seven Things You Need to Know to Reach Your Customers Across TVs, Computers, Tablets, and Mobile Phones*. John Wiley & Sons, 2014.
- [7] G. Z.-h. L. Jia-xi and W. Jin. Xmpp-based ejabberd server privacy-list implementation. *Microcomputer Information*, 18:074, 2010.

<sup>41</sup>This is the second footnote. It starts a series of three footnotes that add nothing informational, but just give an idea of how footnotes work and look. It is a wordy one, just so you see how a longish one plays out.

- [8] S. Krug. Don't make me think, revisited: A common sense approach to web (and mobile) usability. *Berkley: New Riders*, 2014.
- [9] Microsoft Advertising, Flamingo & Ipsos OTX. Cross-screen engagement: Multi-screen pathways reveal new opportunities for marketers to reach and engage consumers. March 2013.
- [10] C. Newell and L. Miller. Design and Evaluation of a Client-side Recommender System. Technical report, British Broadcasting Corporation, 01 2014.
- [11] Nielsen Holdings N.V. Action figures: How second screens are transforming tv viewing. Technical report, June 2013.
- [12] D. A. Norman. *The design of everyday things*. Basic books, 2002.
- [13] P. Pu and L. Chen. Trust building with explanation interfaces. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 93–100. ACM, 2006.
- [14] B. Schopman, D. Brickly, L. Aroyo, C. van Aart, V. Buser, R. Siebes, L. Nixon, L. Miller, V. Malaise, M. Minno, et al. Notube: making the web part of personalised tv. 2010.
- [15] C. Shih. *The Facebook era: Tapping online social networks to build better products, reach new audiences, and sell more stuff*. Prentice Hall, 2009.
- [16] Y. X. Skadberg and J. R. Kimmel. Visitors's flow experience while browsing a web site: its measurement, contributing factors and consequences. *Computers in human behavior*, 20(3):403–422, 2004.
- [17] Think With Google. The New Multi-Screen World: Understanding Cross-Platform Consumer Behavior. *Think Newsletter*, August 2012.

## APPENDIX

### A. HEADINGS IN APPENDICES

The rules about hierarchical headings discussed above for the body of the article are different in the appendices. In the **appendix** environment, the command **section** is used to indicate the start of each Appendix, with alphabetic order designation (i.e. the first is A, the second B, etc.) and a title (if you include one). So, if you need hierarchical structure *within* an Appendix, start with **subsection** as the highest level. Here is an outline of the body of this document in Appendix-appropriate form:

#### A.1 Introduction

#### A.2 The Body of the Paper

##### A.2.1 Type Changes and Special Characters

##### A.2.2 Math Equations

##### *Inline (In-text) Equations*

##### *Display Equations*

##### A.2.3 Citations

##### A.2.4 Tables

##### A.2.5 Figures

##### A.2.6 Theorem-like Constructs

##### *A Caveat for the T<sub>E</sub>X Expert*

#### A.3 Conclusions

#### A.4 Acknowledgments

#### A.5 Additional Authors

This section is inserted by L<sup>A</sup>T<sub>E</sub>X; you do not insert it. You just add the names and information in the `\additionalauthors` command at the start of the document.

#### A.6 References

Generated by bibtex from your .bib file. Run latex, then bibtex, then latex twice (to resolve references) to create the .bbl file. Insert that .bbl file into the .tex source file and comment out the command `\thebibliography`.

### B. MORE HELP FOR THE HARDY

The acm\_proc\_article-sp document class file itself is chock-full of succinct and helpful comments. If you consider yourself a moderately experienced to expert user of L<sup>A</sup>T<sub>E</sub>X, you may find reading it useful but please remember not to change it.