**National College of Ireland**

**MSc in Web Technologies**

**2013/2014**

**ARIA**

**Project – Loyalty Card Holder**

**Entaria**

****

**Hugh Kelly 13117386**

**Adrian Mann 12110701**

**Brendan O’Brien 13122096**



Contents

[1 Motivation 3](#_Toc386731063)

[2 State Of The Art Review 4](#_Toc386731064)

[3 User Interface Design 5](#_Toc386731065)

[3.1 User Interaction Design Strategies 7](#_Toc386731066)

[3.2 User Testing 7](#_Toc386731067)

[4 Architecture 8](#_Toc386731068)

[4.1 Application Architecture: 8](#_Toc386731069)

[4.2 Security: 8](#_Toc386731070)

[4.3 Toolkits and Frameworks: 8](#_Toc386731071)

[4.4 Data Transfer Strategies: 8](#_Toc386731072)

[4.5 Evaluation and Testing: 9](#_Toc386731073)

[5 Summary 10](#_Toc386731074)

[References 11](#_Toc386731075)

# Motivation

## Project Scope

This project is a rich internet application that is intended to provide an interactive, responsive and modern front end to an enterprise application framework that is implemented for the business sector.

The scope of the project is to develop, in conjunction with the Enterprise Frameworks module, a cloud based server that will provide a loyalty card support system for small to medium sized businesses. The system shall allow businesses without IT specialism to partake in a loyalty card scheme without having to issues loyalty cards themselves.

In addition to this, and the focus for this module, we will design and implement a front end application that features many modern and upcoming frameworks that will be relevant in the year 2022.

The list of frameworks will include AngularJS which demonstrates a Model, View, Controller capability. We will also implement a number of APIs which will keep in theme with the offering of the application. We will also use the CanvasJS library to display json data to the user. There is also Google Map API features on the site. We have embedded a twitter feed to display relevant tweets. We also have created a form that posts data to a back end server.

## Area of Contribution

The team has identified a gap in the market for a Loyalty Card System that removes the need for multiple paper based cards and will satisfy many retailers across multiple store types. We aim to use an existing resource, in Iarnrod Eireann’s Leap Card, and this will benefit users and retailers alike.

We are interested in creating a front end application that users can interact with and one that will compel users to return to the application via interactivity, responsiveness and interesting features.

# State Of The Art Review

The rate of change of technologies used in the internet seems to be ever increasing. For example in 2007 MicroSoft launched Silverlight as a framework to support RIAs but after quick acceptance in development it is now no longer seen as a practical framework for development of new projects, MicroSoft have not made a major upgrade to it since May 2012 and Netflix, possibly the largest users of Silvelight, have announced that they will be moving to HTML5. AT this point in time, May 2014, the most important enablers for RIAs are HTML5, JavaScript and CSS3.

**HTML5**

“This specification defines the 5th major revision of the core language of the World Wide Web: the Hypertext Markup Language (HTML). In this version, new features are introduced to help Web application authors, new elements are introduced based on research into prevailing authoring practices, and special attention has been given to defining clear conformance criteria for user agents in an effort to improve interoperability.” (W3C, 2014)

From this definition of HTML5 provided by the W3C working group we can see that more capability for rich internet applications has been included in HTML’s repertoire. HTML5 is the fifth generation of HTML, the markup language that has been used to describe the layout of web pages since the early 1990s. It is an elaboration of the previous generation of HTML designed to replace and extend HTML4, XHTML and DHTML. One of the main improvements of HTML5 over previous versions is its support for multimedia e.g. the introduction of support for <audio> and <video> tags (implying that HTML5 browsers will support the playback of media referred to via these tags). It also includes an API that can be called by JavaScript, this API allowing support for more advanced web applications than was possible heretofore (W3Schools, 2014).

Some of the capabilities supported by HTML5’s Web API include:

Web Storage: a simple key-value pair storage system that works in a similar way to cookies with a larger storage volume.

Geolocation: an API that can provide geographical position information (i.e. latitude and longitude co-ordinates) using various sources of location such as WiFi (including other identifiable WiFi hubs in range that are not necessarily being connected to), IP address, GPS and mobile phone cell ID depending on what is available to the browser from the client device

Web Sockets: a communication protocol that provides bi-directional (full duplex) traffic over one TCP/IP connection directly for within the browser.

Canvas: an API that allows for 2D drawing directly to the browser document.

Web Worker: an independently executed JavaScript thread that can run in the background suitable for use in browsers hosted on modern multicore systems.

WebRTC: Web Realtime Communication allows browser-to-browser connectivity to allow voice messaging, IM chat, video conferencing, etc. directly between browsers in a similar way to Skype connectivity but without using a third party application or plugin.

The increased capabilities of HTML5 are ideal for RIAs and reduce the need for ‘thick’ third party plugin-like frameworks such as Flash, Flex or Silverlight.

**CSS3**

CSS is a simple and flexible way to style Web Pages and provide a number of feature which can be easily implemented. CSS3 is built on CSS2 and is backwards compatible. CSS3 is modular based and this allows CSS3 to be backwards compatible with CSS2 as each of these modules extend the CSS capabilities. CSS3 is the standard in effect of W3 working group.[]

CSS3 does not require any special tools to develop and can be easily integrate into web page design. CSS and HTML can be easily  integrate with each other to enhance the visuals of a web page.

CSS3 supports the following.

·         Animations

·         User Interface

·         Text Effects

·         2D/3D Transformations

·         Box Model

·         Multiple Column Layout

·         Image Values and Replaced Content.

·         Selectors

·         Rounded Corners

CSS3 allows developer to implement some nice special effect such as Transformation and we have integrated a Spinning Cube which has an image on all for sides to show this capability.

CSS3 allows designers to implement some very complex manipulation with very little programming in Javascript, which most designers shy away from. CSS3 allows programmers to create stylish pages using the CSS3 scripts. A designer can write 1 line of code which can replace 10 lines of code.

Media Queries take care some of the issues with view the web pages on different devices such as mobile, tablet or desktop. They can detect the device type based on the device attributes so the correct stylesheet can be targeted directly to the device viewing  the page.

CSS3 style sheets should be kept in separate files which allow these files to be compressed  and cached on the browser side.

CSS3 is not HTML5 and visa versa HTML5 has a lot of great features but CSS3 is what does the really spectacular effects such as Rotations, scaling and animations. CSS3 continues to be developed using modules which allow more feature to be implemented on an on going basis.

**Jquery**

It is an open Source Library licensed under M.I.T and is free to usse. It is a light weight footprint at only 32KB when minified and qzipped,  it is Cross Browser and its CSS3 compliant.  It is pack with features, very fast, concise Javascript Library.  It allows traversal of the HTML Document , AJAX and Event Handling. It has a plugin architecture , the API is fully Documented.

It is used by 500K websites and is one of the most popular Libraries among a large  developer community, with a very large number of contributors from developers. It is being used by large and small organisations alike.

It is easy to learn, and the Jquery philosophy is ***Write less, do more***. It is very friendly to use and helps avoids conflicts with other javascrip libraries.

**AngularJS**

AngularJS is an open source JavaScript framework that allows you to extend HTML vocabulary for your application. AngularJS enforces the Model View Controller pattern by dividing the application into MVC sections and then Angular connects the separate sections.

The view is constructed using HTML and we can also specify which controllers to use for each HTML element or multiple elemnts. In other words, lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. Angular's data binding and dependency injection eliminate much of the code you currently have to write. And it all happens within the browser, making it an ideal partner with any server technology (Docs.angularjs.org, 2014).

Another benfit of angular is that is reduces the disconnection between the static HTML and how we can modify and perform actions on this HTML via data binding, DOM control structures and grouping of HTML into reusable components.

Angular is a complete client side solution which includes everything needed to build a CRUD application as well as unit testing, end to end testing, seed application with directory layout and test scripts as a starting point (Docs.angularjs.org, 2014).

**CanvasJS**

CanvasJS is an easy to use HTML5 & JavaScript Charting library built on Canvas element. Graphs can render across devices including iPhone, iPad, Android, Windows Phone, Desktops, etc. This allows you to create rich dashboards that work on all devices without compromising on maintainability or functionality of your web application. Charts include several good looking themes and is over 10x faster than conventional Flash and SVG Charts – resulting in lightweight, beautiful and responsive dashboards (Urs, S. (2014))

CanvasJS offers many different chart types that we have illustrated in the application with some jQuery which allows the user to interact and choose how they would like to view the data that we have provided.

Below is how a minimal basic Column Chart would look like. Here are important things to remember

1. Instantiate a new **Chart** object by sending the ID of div element where the chart is to be rendered.
2. Pass all the Chart related **“options”** to the constructor as the second parameter.
3. Call **chart.render()** method to render the chart

Chart **“options”** mainly contains 4 important items.

1. **title** object with its **text** property set.
2. **dataPoints** – which is an array of all data items to be rendered
3. **dataSeries** – parent of dataPoints that also defines type of chart and other series wide options
4. **data** – array element which is collection of one or more dataSeries objects. Here we have only one dataSeries.

Though we are going to use Column Charts to learn various concepts, it should be easy to change them into any kind of chart that you wish! Just change the type property of dataSeries to any chart type that you need – bar, area, line, scatter, stackedColum, etc. (Urs, S. (2014))

**Google Maps API**

##### The Google Maps JavaScript API allows users to build highly customisable maps with your own

##### content and imagery and to use these creation to create rich applications and stunning visualisations

##### of your data, leveraging the comprehensiveness, accuracy, and usability of Google Maps and a

##### modern web platform that scales as you grow (Developers.google.com, (2014)).

We have found extremely efficient to implement the Google Maps API due to the ease of use and the extensive documentation that is porivided on the Mpas section of the Google Developers website. However, it should be noted, for non-technical users, that an understanding of JavaScript is recommended before getting started with this API.

We firstly had to obtain a key which allows us to monitor our application’s Map API usage. If your application's Maps API usage exceeds the [Usage Limits](https://developers.google.com/maps/documentation/javascript/usage#usage_limits), you must load the Maps API using an API key in order to purchase additional quota (Developers.google.com, (2014)).

All instances of a Google Map are centered around a few basic points as can be seen below:

· We declare the application as HTML5 using the <!DOCTYPE html> declaration.

· We include the Maps API JavaScript using a script tag.

· We create a div element named "map-canvas" to hold the Map.

· We create a JavaScript object literal to hold a number of map properties.

· We create a JavaScript map object, passing it the div element and the mapproperties.

· We use an event listener to load the map after the page has loaded.

With the aid of numerous tutorials and our team’s experience with JavaScript and Google Maps we implemented this feature to the site which will display a map that is centered on your current location and display coffee sotres nearby.

# User Interface Design

The User Interface design for the ARIA continuous assessment was undertaken in conjunction with the Enterprise Frameworks project. In the formative stages of the development process we completed initial sketches of a possible user interface design and we refined these sketches through the wire framing process using the Balsamiq software package. An example of the sketches that were done can be found below in Figure 3.1.

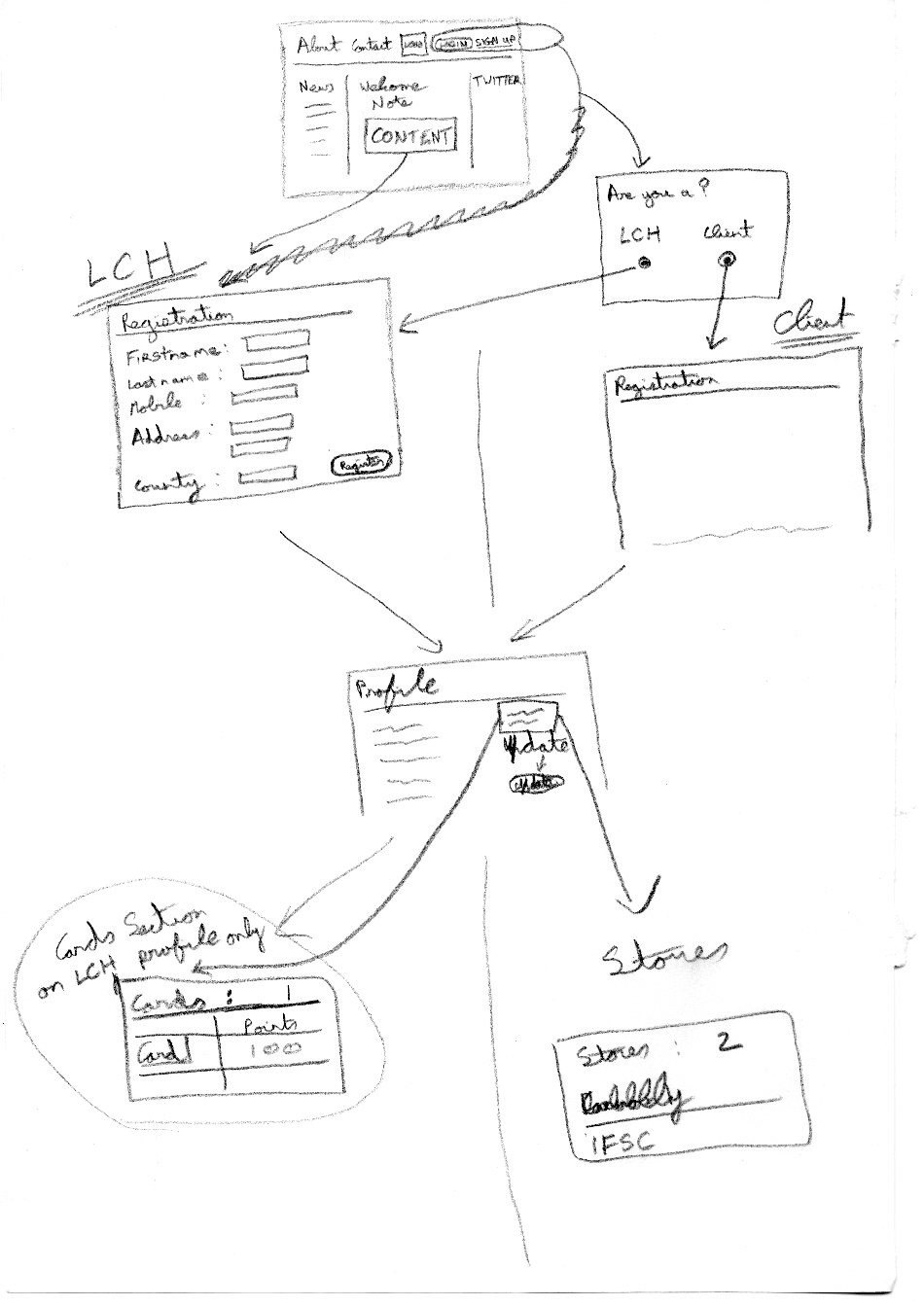


Figure 3.1 Application Sketches

As mentioned above, these sketches were further enhanced in the higher fidelity wire frames that were produced with Balsamiq and an example of the home page wire frame can be seen in figure 3.2 below.

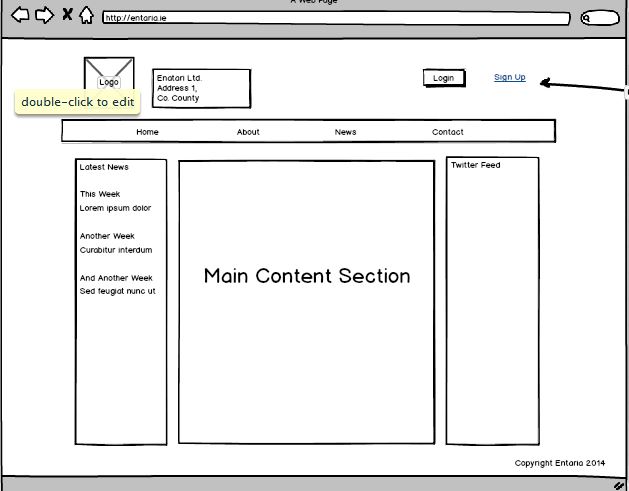


Figure 3.2 Landing Page Wire Frame

Due to the agile nature of our development process it is evident that there were some adjustments made between the first and second stages of the user interface design process. This pattern was continued as we implemented our interaction design strategy. Therefore, the result that we have deployed contains some features that were not including in the original mock ups and/or sketches. However, these have not only improved the design of the application but also show cased some of the technologies that we have encountered in tutorials and lectures. The final home page design can be seen in Figure 3.3 below.

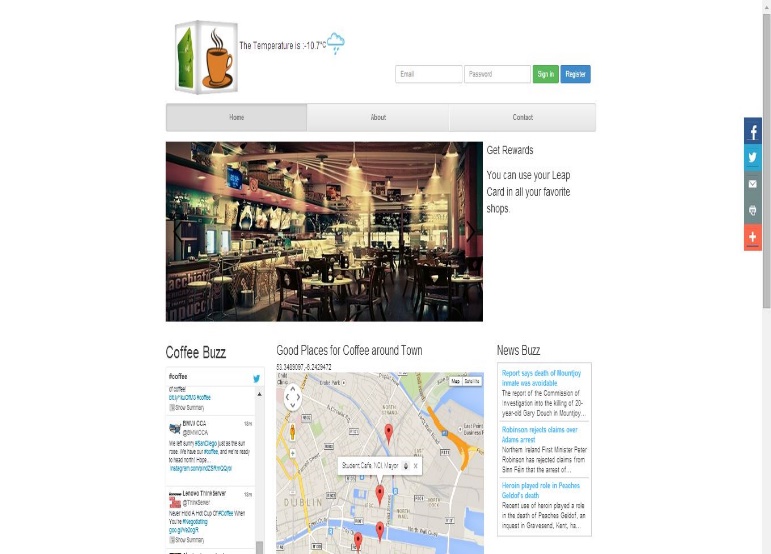
****

Figure 3.3 Landing Page – Final Design

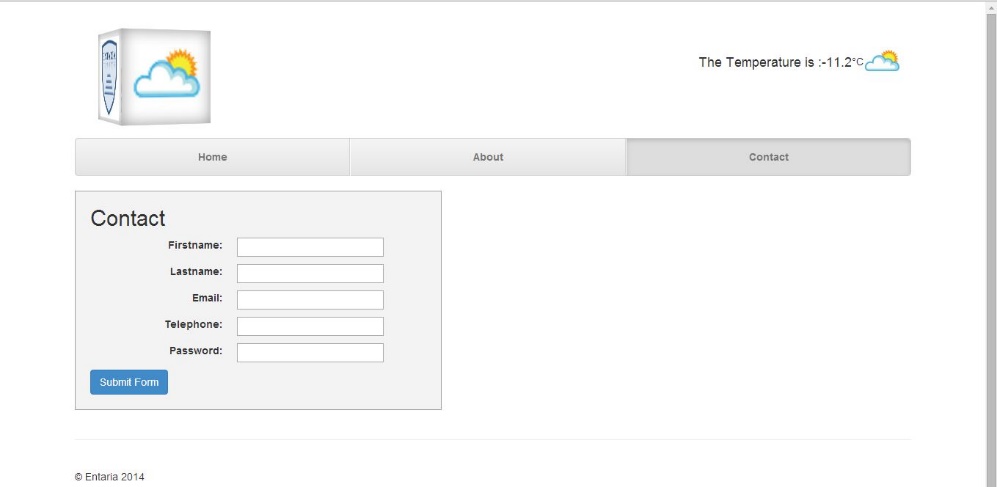


Figure 3.4 Contact Page – Final Design

The layout of the site and the responsiveness have been implemented with the Twitter Bootstrap front end framework. This framework will reduce the work load in terms of designing a layout that is interactive and provided a large amount of sleek styling.

The final layout incorporates many features of a modern website in a style and format that is familiar to users of web applications yet one that we anticipate will be suitable in the year 2022. As is evident in the image of the homepage above, the basic layout shows a header with navigation, a slider and basic three column layout.

The theme of the site and the nature of the application that we are providing was heavily focused on and is evident throughout the style of the application. The theme of coffee is also noticeable in the news, twitter feed, weather, google map coffee shop locations and data display on the homepage.

## User Interaction Design Strategies

As previously stated, we have focused on creating an interactive design that is available to as many users as possible on as many devices as possible. A major feature of the application is that it is a SPA or Single Page Application which was implemented using JavaScript. This improves load time and increases the enjoyment of using the application.

The Google Map is situated purposefully in the centre of the application to gain the users attention and to drive traffic to stores which is the goal of the system for both of our participants. In relation to this we have focused on the overall flow of the site. The lack of a page refresh, and the columnar layout of the site does not disrupt the user flow. The page refresh creates an artificial break in the action – or a break in the user’s flow (Scott, B & Niel, T, 2009).

We encourage users to share our site as much as possible with social media, email and print icons that are located in a menu that is always evident on the screen to encourage use.

## User Testing

The user testing focused on a think aloud process. I have documented the scenarios below that our users had to undertake.

**Scenario 1**

Navigate to the Coffee Sales section of the website. View the data that is displayed on the doughnut chart that is displayed. Identify the buttons above the chart and click on one of your choice to display the same information in a different style of graph.

**Scenario 2**

Navigate to the Twitter section of the website. View the Tweets that are displayed and navigate with through the list of coffee related tweets.

**Scenario 3**

Navigate to the header section of the website and identify the menu navigation. Click on the “Contact” tab. When you are taken to the “Contact” page, identify the Contact form. Enter your details into this form and hit submit.

Each user who completed the Think Aloud testing was recorded using the Cam Studio software.

**Satisfaction**

Each user was also asked each participant in the study to complete as systems usability test. The test is documented below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Strongly Disagree 1 | 2 | 3 | 4 | Strongly Agree 5 |
| 1. I found the system unnecessarily complex. |  |  |  |  |  |
| 1. I thought the system was easy to use. |  |  |  |  |  |
| 1. I think that I would need the support of a technical person to be able to use this system. |  |  |  |  |  |
| 1. I found the various functions in this system were well integrated. |  |  |  |  |  |
| 1. I thought there was too much inconsistency in this system. |  |  |  |  |  |
| 1. I think that I would like to use this system frequently. |  |  |  |  |  |
| 1. I would imagine that most people would learn to use this system very quickly. |  |  |  |  |  |
| 1. I found the system very cumbersome to use. |  |  |  |  |  |
| 1. I felt very confident using the system. |  |  |  |  |  |
| 1. I needed to learn a lot of things before I could get going with this system. |  |  |  |  |  |

# Architecture

## Application Architecture:

Outline the solution architecture for your application. You should justify the approach you took, and how it supports the project scope. This means mapping back to the state of the art review when describing the features and functionality of your system.

## Security:

Evaluate industry standard error handling, and outline how you integrated these approaches within your application.

## Toolkits and Frameworks:

Discuss the toolkits and frameworks used and the justification for using them.

## Data Transfer Strategies:

Our data transfer strategy  is to use JSON  (JavaScript Object Notation)., throughout  the application, we are consuming a number of API’s using Angular.JS HTTP and JQuery.get.  All data retrieval  willbe performed using AJAX and JSON.  JSON  is a lightweight data-interchange format. It is human readable,  it is completely language independent. Unlike xml,   JSON can support Objects, arrays, strings and numbers.  We are also sending back data to the server using JSON which is then inserted into the CardHolderDetails table on the Azure Website.

JSON allows  data to be sent straight into the application with little or no parsing which reduced the load on the Host Processor this is what make it ideal for Mobile devices with limited horsepower.

JSON is much more suitable for use with Javascript over XML as it can support data types and arrays objects.

## Evaluation and Testing:

You must include a short description of how your application was evaluated for its audience.

Critically analyse the testing methodology employed, as well as any debugging

# Summary

More of the same

# References

W3C (2014). *HTML5* [online] Available at http://www.w3.org/TR/html5/ [accessed 2nd May 2014]

W3Schools (2014). *HTML5 Introduction* [online] Available at http://www.w3schools.com/html/html5\_intro.asp

Gamma, E. et al., 1995. *Design Patterns: Elements of Reuseable Object-Oriented Software,* Addison Wesley, New York, p.293-303

Google Inc., 2014. *What is Angular?* [online] Available at http://docs.angularjs.org/guide/introduction [accessed 6th March 2014]

Docs.angularjs.org, (2014). *AngularJS*. [online] Available at: https://docs.angularjs.org/guide/introduction [Accessed 2 May. 2014].

Urs, S. (2014). *Beautiful HTML5 & JavaScript Charts | CanvasJS*. [online] CanvasJS. Available at: http://canvasjs.com/ [Accessed 2 May. 2014].

Developers.google.com, (2014). *Google Maps Web APIs — Google Developers*. [online] Available at: https://developers.google.com/maps/web/ [Accessed 3 May. 2014].