ATU

Ollscoil
Teicneolaíochta
an Atlantaigh

Atlantic
Technological
University

Biometric Data Analysis in Digital Game Scenario

By
Rodrigo Almeida

November 16, 2023

Minor Dissertation

**Department of Computer Science & Applied Physics,
School of Science & Computing,
Atlantic Technological University (ATU), Galway.**

B.Sc. (Hons) in Software Development

Contents

1 Introduction **2**

2 Methodology **3**

3 Technology Review **4**

3.1 Executive Dashboard 4

3.1.1 Angular 4

3.1.2 Angular architecture 5

3.1.3 Advantages of using Angular 5

3.2 AI module - Data analysis 7

3.2.1 Hadoop 7

4 System Design **9**

4.1 Working with Images 9

5 System Evaluation **10**

5.1 Working with Tables 10

6 Conclusion **12**

List of Figures

3.1 Angular application architecture 5

List of Tables

5.1 Conversion from Hexadecimal to Binary 11

Chapter 1

Introduction

This chapter should provide a clear context for your project and set out its objectives. You can cite references from the bibliography using IEEE format, such as Claude Shannon [?] and John Von Neumann *et al* [?]. Use Google Scholar's BibTex export function to get a LaTeXformatted citation to copy and paste into the *references.bib* document.

Chapter 2

Methodology

Describe the way you went about your project. Was your approach to the problem valid? You need to discuss both your software development methodology and your research methodology.

Chapter 3

Technology Review

3.1 Executive Dashboard

When it came to developing the Executive Dashboard for data analysis it was crucial to select the tools that could transform data into easily comprehensible information. After conducting research and exploring alternatives Angular and Chart.js were chosen for specific reasons.

3.1.1 Angular

Angular is based on TypeScript and is an open-source framework, and it's a great tool for building client-side web applications. It can be used for single-page applications (SPAs) or for enterprise-level solutions.

Main Concepts: Components, Modules and Services

Angular is composed of three foundational blocks: Components, Modules and services.

Components

Components are like Lego blocks of the application. Each component holds a portion of the user interface and its behaviour. Components serve as the bridge between the application data and what the user experiences on the screen.[1]

Modules

In Angular modules serve as containers that group related components, directives, and services together that can be combined with other modules. It plays an important role in improving maintainability and re-usability, key concepts of Angular development.[2]

Services

Angular services use typescript classes with injectable decorators. The decorator tells angular that the class is a service and can be injected into other components that need that service.[3]

3.1.2 Angular architecture

Angular uses the Model-View-Controller (MVC) pattern, with a variation known as Model-View-ViewModel (MVVM). The controller is responsible for the interaction between the model and the view.

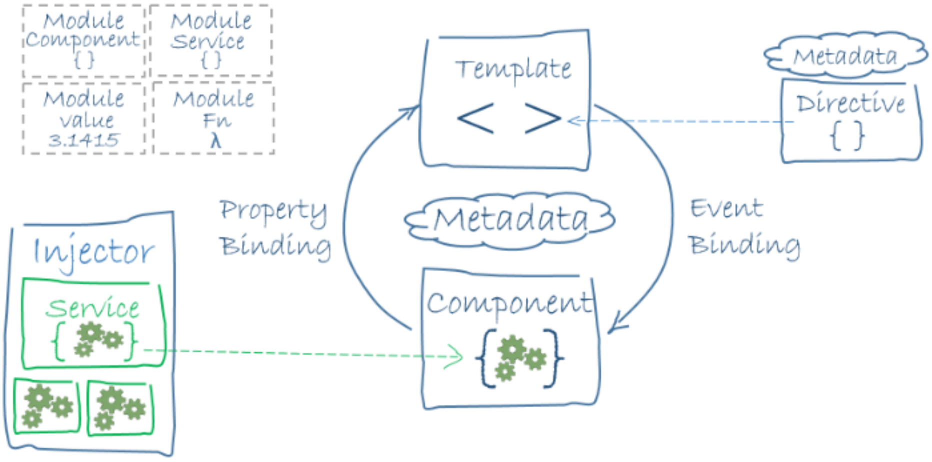

Figure 3.1: Angular application architecture

3.1.3 Advantages of using Angular

Component-Based Architecture

As mentioned earlier in discussions Angular organisess its functionalities into components. These components have the ability to communicate with each other enabling updates to sections without affecting the rest of the application.

Mobile-Friendly Approach

Angular incorporates techniques such, as lazy-loading, which means loading parts of the application (like images) only when they are needed. This ensures that users do not experience long waiting times.

Two-Way Data Binding

With Angular two way data binding data can seamlessly flow between the component and the view allowing for synchronization.

Asynchronous Programming

By utilizing programming executes code in a non-sequential manner and employs multi-threading to enhance performance. This speeds up operations and prevents system freezes, providing users with a seamless experience.

Single-Page Applications

Angular creates a dynamic single-page application which can be navigated without page reloads, improving the user experience with better user interaction and engagement.

Code Re-usability

The component-based architecture of Angular promotes the re-usability of UI components saving development time.

Dependency Injection

With dependency injection in place, Angular allows for the creation of objects that rely on other objects. This improves modularity and efficiency, within the app.

Angular Material

Angular's documentation offers a range of built user interface components and modules that adhere to Google's Material Design principles. This greatly facilitates the developer's work, simplifying the design process and enabling application development.

Angular CLI

Angular command line interface gives the developer the ability to generate Angular projects, modules, services, and components with a single command, this not only saves time but also reduces configuration errors, it gives the developer the freedom to dive into creative aspects of the project, focusing on innovation and functionality rather than getting bogged down by initial setup complexities.

Angular command line interface gives the developer the ability to generate Angular projects, modules, services, and components with a single command, this not only saves time but also reduces configuration errors, it gives the developer the freedom to dive into creative aspects of the project, focusing on innovation and functionality rather than getting bogged down by initial setup complexities.

3.2 AI module - Data analysis

Ninety per cent of the world's data was generated in just the last two years. In the early 2000s, the amount of data being generated exploded exponentially with the use of the internet, social media, and various other technologies. Organizations found themselves facing a massive volume of data that was very hard to process. To address the challenge, the concept of Big data emerged. Big data refers to extremely large and complex data sets that are difficult to process using traditional methods. [4]

3.2.1 Hadoop

As the volume of data grew rapidly across the globe, organizations needed a way to manage it to gain valuable insights. That's where Hadoop came in. In 2006, a team of engineers from Yahoo developed Hadoop inspired by Google's MapReduce. Hadoop has introduced a new way of handling data called distributed processing. Now, to process all data it wouldn't use single machines anymore, instead, Hadoop uses multiple computers, allowing large amounts of data to be processed way faster than before. Hadoop has mainly two main components: HDFS (Hadoop Distributed File System) and MapReduce. HDFS stores the into multiple computers, and MapReduce is responsible for processing the data in parallel allowing the organizations to store and process large amounts of data. Hadoop was a great advance in big data processing, however, it had a few limitations. One of the biggest problems was that it relied on storing data on disk. This slowed down data processing because every time a job ran it had to save its data to disk, read it back, process it, and save it back to disk Another problem is that Hadoop processes data only in batches. This means a new job couldn't be submitted before the other job ended. With Hadoop, storing and processing large amounts of data became possible, even with some limitations it was an important step in big data processing.[5]

Chapter 4

System Design

Provide a detailed explanation of the overall system architecture [?], i.e. the HOW of the project. Use UML, system architecture diagrams, screenshots, code snippets and algorithms to illustrate your design.

4.1 Working with Images

You can embed an image in a LaTeXdocument using the technique shown below. System diagrams and images with a small numbers of colours (100s, not 1000s) should be stored in PNG format. Although LaTeXdoesn't care where you place your images, it is good practice to place them in a single sensible directory and apply some sort of hierarchy to them, e.g. the path images/chapter1 might contain all of the images for Chapter 1 of your dissertation.

Chapter 5

System Evaluation

Evaluate your project against the objectives set out in the introduction. This chapter should present results if applicable and discuss the strengths and weaknesses of your system. This is a clear opportunity for you to demonstrate your critical thinking in relation to the project.

5.1 Working with Tables

Table 5.1 can be referenced with the label given to the table, i.e. **\ref{table:HexToBin}**. Note that LaTeXwill place the table wherever it deems fit. Don't bother trying to change where a table or figure is placed until your document is ready for final layout.

Hexadecimal to Binary					
Hex	Binary 2	Hex	Binary	Hex	Binary
1	00000001	B	00001011	15	00010101
2	00000010	C	00001100	16	00010110
3	00000011	D	00001101	17	00010111
4	00000100	E	00001110	18	00011000
5	00000101	F	00001111	19	00011001
6	00000110	10	00010000	1A	00011010
7	00000111	11	00010001	1B	00011011
8	00001000	12	00010010	1C	00011100
9	00001001	13	00010011	1D	00011101
A	00001010	14	00010100	1E	00011110

Table 5.1: Conversion from Hexadecimal to Binary

Chapter 6

Conclusion

Briefly summarise your context and objectives. Remind the reader about the overall rationale and goals of the project. Highlight your findings from the System Evaluation chapter.

Bibliography

[1] Angular Components, https://angular.io/api/core/Component.

[2] Angular Modules, https://angular.io/guide/architecture-modules.

[3] Angular Services, https://angular.io/guide/architecture-services.

[4] News. *Significance*, 9(4):2–3, 08 2012.

[5] Ivanilton Polato, Reginaldo Ré, Alfredo Goldman, and Fabio Kon. A comprehensive view of hadoop research—a systematic literature review. *Journal of Network and Computer Applications*, 46:1–25, 2014.