comparing object-oriented and functional programming paradigms in the Web

Student Name: ADNAN HAMRBTAN
SID: 3301170001

SCHOOL OF SCIENCE & TECHNOLOGY
A thesis submitted for the degree of
Master of Science (MSc) in Information and Communication technology

2017-2018
THESSALONIKI – GREECE
comparing object-oriented and functional programming paradigms in the Web

Student Name: ADNAN HAMRBTAN

SID: 3301170001

Supervisor: Prof. Apostolos Ampatzoglou

SCHOOL OF SCIENCE & TECHNOLOGY
A thesis submitted for the degree of
Master of Science (MSc) in Information and Communication technology systems

2017-2018
THESSALONIKI – GREECE
Abstract

This dissertation was written as a part of the MSc in ICT Systems at the International Hellenic University. In this research, the website of the Libyan Embassy in Greece was developed to study the difference between object-oriented programming and functions programming. Many techniques were used to develop this system. Examples include the use of the BDO library, which provides an effective means of developing web applications and MySQL database, Based on what is accomplished, the system provides the collected functions as well as the hypothesis study.

ADNAN HAMRBTAN
Date
Background Information

3. Libyan Academy (2016) computer science, function programming Paradigms.
Table of Contents
Abstract................................................................................................................................. 4

Background Information ........................................................................................................ 5

Chapter one ............................................................................................................................. 9
Introduction.............................................................................................................................. 9
  1.1 Overview ......................................................................................................................... 10
  1.2 case study ....................................................................................................................... 12
  1.3 problem statement .......................................................................................................... 12
  1.4 importance of the study ................................................................................................. 12

Chapter 2 ................................................................................................................................ 14
Web Development Languages and Technologies ................................................................. 14
  2.1 Overview ......................................................................................................................... 15
  2.2 Web technologies ........................................................................................................... 15
  2.3 website builder ............................................................................................................... 16
    2.3.1 Types of website builders ......................................................................................... 16
  2.4 JavaScript ....................................................................................................................... 17
  2.5 ECMAScript/ES6/ES8 .................................................................................................... 18
  2.6 Node.js ......................................................................................................................... 19
  2.7 Libraries ....................................................................................................................... 19
  2.9 Ruby ............................................................................................................................. 20
  2.10 HTML/CSS .................................................................................................................. 21
  2.11 ASP.NET ..................................................................................................................... 21
  2.12 C# ............................................................................................................................... 21
  2.13 Java ............................................................................................................................ 21
  2.14 Database Options for Web Development ..................................................................... 22
    2.14.1 SQL ....................................................................................................................... 22

Chapter 3 .............................................................................................................................. 23
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security in the Software Development</td>
<td>23</td>
</tr>
<tr>
<td>3.1 Overview</td>
<td>24</td>
</tr>
<tr>
<td>3.2 Web server security challenge and defense</td>
<td>26</td>
</tr>
<tr>
<td>3.3 Database encryption</td>
<td>26</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>29</td>
</tr>
<tr>
<td>Programming Paradigms</td>
<td>29</td>
</tr>
<tr>
<td>&amp;</td>
<td>29</td>
</tr>
<tr>
<td>Statistical analysis</td>
<td>29</td>
</tr>
<tr>
<td>4.1. Programming Paradigms</td>
<td>30</td>
</tr>
<tr>
<td>4.2 Object-oriented programming</td>
<td>31</td>
</tr>
<tr>
<td>4.2.1 OOP concepts</td>
<td>31</td>
</tr>
<tr>
<td>4.2.2 An attribute</td>
<td>32</td>
</tr>
<tr>
<td>4.2.3 A method</td>
<td>32</td>
</tr>
<tr>
<td>4.2.4 Inheritance</td>
<td>32</td>
</tr>
<tr>
<td>4.3 Functional programming</td>
<td>33</td>
</tr>
<tr>
<td>4.3.1 Immutability</td>
<td>33</td>
</tr>
<tr>
<td>4.3.2 Recursion</td>
<td>33</td>
</tr>
<tr>
<td>4.3.3 Pure Functions and Referential Transparency</td>
<td>34</td>
</tr>
<tr>
<td>4.4 Case Study Design</td>
<td>35</td>
</tr>
<tr>
<td>4.5 Case Selection and Units of Analysis</td>
<td>38</td>
</tr>
<tr>
<td>4.5.1 split the requirements</td>
<td>39</td>
</tr>
<tr>
<td>4.5.2 Define system requirements</td>
<td>39</td>
</tr>
<tr>
<td>4.6 Sequence diagram</td>
<td>51</td>
</tr>
<tr>
<td>4.6.1 Active user</td>
<td>51</td>
</tr>
<tr>
<td>4.6.2 Deactivate user</td>
<td>52</td>
</tr>
<tr>
<td>4.7 Data collection</td>
<td>53</td>
</tr>
<tr>
<td>4.7.1 Results</td>
<td>55</td>
</tr>
</tbody>
</table>
4.7.2 Paired Sample T-Test ................................................................. 59
4.7.3 Boxplot .................................................................................. 61
4.8 Agile Methodology ................................................................. 63
4.9 Results .................................................................................... 64
Conclusion .......................................................................................... 65
Discussion .......................................................................................... 66
Future work ......................................................................................... 67
Appendices ......................................................................................... 69
Tables and pictures ............................................................................. 76
Chapter one

Introduction
1.1 Overview

Nowadays websites are the most convenient way to present and disseminate information to the maximum number of people in the world. The web browsers are the means to render the information on a webpage, the basic building blocks of a website which has the basic structure (architecture) Written in web program.

In today’s information age. Almost all organization have a website with their manifesto and their product and services information, it is probably the most economic and the most convenient way to disseminate information all over the world. Web technologies is more and more becoming one of the decisive factors of business success.

A program is a set of instructions that tell the computer to do various things; sometimes the instruction it has to perform depends on what happened when it performed a previous instruction. in today’s information age, almost all organization have a website with their manifesto and there product and services information. it is probably the most convenient way to disseminate information all the world.

We will define the website as big container of relevant and related information arranged in some logical way. web design can be best viewed as client-server architecture, where a client machine requests for services and the validates the request to access service, probably from a database:

Depending on the functionalities, a complete web design can be further classified into three main components:

**Fig 1.1 Atypical client serves Architecture**

- **Web client**
  - Markup
  - Languages (HTML, DHTML, XHTML, CSS)
  - SCRIPTS Javascript, vbscript, java applet
- **Web container**
  - (Apache, tomcat, iis, coldfusion)
  - Programs (jsp, asp, PHP)
  - Dot net environment
- **Database server**
  - (Table systems)
  - SQL, ORCAL
  - MS ACCESS
  - MYSQL
**Client Side Design:**

Client side is the actual interface from the user, the application such a web browser (IE, Netscape, google chrome) on the client machine sends service request data to web serves (TOMCAT, APACHE, IIS) running on the server machine.

The web server then send either sends an existing page to the client machine or generates a new page and sends to the client machine accordingly. The client web page is typically constructed by HTML, CSS and some script (Java script).

**Server Side Design:**

Server side is the logical controlling part of the website, the web container (such as Apache, Tomcat, IIS) running under the serves machine handles the client request, validate with the server side program (written in PHP, JAVA, JSP, ASP, VB and C++) and then generates an appropriate page or locates an existing appropriate page sends that page to the client side, server pages are typically written in PHP, JSP or ASP.NET.

**Database Design:**

Database is always at the back end of the client-server architecture, the data stored in the database is gathered, organized and designed in sophisticated logical manner (such as using DFD, RDBMS, OODB, or UML) and stored in one or more tables.

The webserver can pull up data with the help of a database server (such as MYSQL, Microsoft SQL and ORCAL), fit into a web page and send it to the client machine.

Paradigm can also be termed as method to solve some problem or do some task. Programming paradigm is an approach to solve problem using some programming language or also we can say it is a method to solve a problem using tools and techniques that are available to us following some approach. There are lots for programming language that are known but all of them need to follow some strategy when they are implemented and this methodology/strategy is paradigms. [2]
1.2 case study
The case study has been chosen for study the compare the PHP web app development with Object-oriented programming vs. non-oop (function programming) technologies in order to achieve this goal we decompose the goal to two research questions.

1.3 problem statement
Many interactive web application are written by developers who have little knowledge of writing them in secure from code side, by using developer tools available under their hands like ASP.NET, JAVA, PERL, PHP, combined with fact that web development has rapid development cycle resulting in code which is vulnerable to many mistakes in the development time, Yields the fail down and destruction of the product in short time of maintenance stage?

1.4 importance of the study
- be quick coding revising with clear look.
- provide methodology for site development.
- offer un application code experience.
Background Information


4. Reusability of open source software across domains: A case study
   Maria-Eleni Paschali a, Apostolos Amapzoglou a, Stamati Bibi b, Alexander Chatzigeorgiou c, Ioannis Stamelos a (2017).
Chapter 2

Web Development Languages and Technologies
2.1 Overview

The increasing use of the web to access information and request services has led many organization irrespective of their business activity, to incorporate web development as part of their business, the web as we know today is not only used to advertise information or enable services and products to be purchased, but has grown to incorporate more and more flexibility as well as interactive functionality, some example of web application include e-commerce/e-business websites, search engines, transaction engines and informational web sites conveying news, advertisements articles and many others, expansions in communication technologies and web enabled appliances further explain the evolvement of web applications being used today in the future the use of the web is perceived to grow exponentially with a variety of added services in most business sectors.

2.2 Web technologies

Web technologies contain long lists of jargon and abbreviations that are used in documentation and coding, this list to tick off to understand the jargon:-

- **App, Application:** An application is a program that runs on a computer that is built to do specific tasks to the benefit of the user.
- **Web Application:** A web application is the same as a regular application, but runs on a server instead. The advantage of this is that your application is now accessible to anyone with a web browser, without them having to download it on their computer.
- **Frontend, Client-side:** The user interface or client side of an application is what the user interacts with, which is also referred to as “frontend” as it has a lot to do with the appearance, experience, and interface of the application.
- **Backend, Server-side:** The server side of an application is where all the business logic and authentication logic happens. “Server-side” is also called the “backend” of the app so there’s not necessarily any interface here to interact with.
- **Full-stack:** A full-stack developer is a developer who can program both an application’s frontend and its backend. This is a reasonably complicated skill to master as frontend programming is very different from backend programming and each requires a unique way of thinking.
HTTP(S), SSL, TLS: HTTP is a protocol for transferring data through the use of a web browser, SSL and TLS two popular protocols used on top of HTTP to make HTTP into HTTPS, and therefore a secure protocol for transferring data.

API: API’s are sets of functions and procedures that allow the creation of applications. An API enables access to the features or data of an operating system, application, or another service.

AJAX: With AJAX, web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behavior of the existing page.

JSON: JSON is an open-standard file format that uses human-readable text to transmit data objects consisting of key-value pairs and array data types.

YAML, yml: YAML is a human-readable data serialization language. It is commonly used for configuration files but could be used in many applications where data is being stored or transmitted. It is also a superset of JSON since YAML 1.2.

ENV: An environment variable is a dynamically named value that can affect the way running processes will behave on a computer. This is used for, e.g., storing secret access keys and passwords.

Node Modules (node_modules): This is a folder in most modern web applications where various mini-frameworks and utility tools are located. More on this later in the Node chapter.

Package.json: This is a file in the root directory of most modern web applications. It details the dependencies used in the applications and the steps to start run it [7]

2.3 website builder

Website builders are tools that allow the construction of websites without manual code editing. Wikipedia’s definition, In other words, a website builder is a program, or tool, that help you build a website.

2.3.1 Types of website builders

There are two main types of website builders: offline and online.
2.3.1.1 Offline Website Builders

Offline website builders come as software programs that you can download and install on your own computer. You will build your website then you can save the files on your computer, and when your website is completed, you just upload all the files in web host.

One advantage of the website builders is that the software is running on your computer locally so you can work on your website even if you don’t have internet. But because you will need to upload your files to a web host, you will need to have at least some technical skills or experience like ftp program or another program to upload it. You will also need to purchase a web hosting account in additional to the website builder software some of them they offer database and some of them they will charge you more for this. Example of an offline website builder is wix.[3]

2.3.1.2 Online Website Builders

Online website builders are web-based and run on the provider’s service. Unlike offline website builders, you don’t need to download or install the software on your computer – all you need is a web browser (ie. Chrome, Firefox or Internet Explorer) and an Internet connection. This allows you to work on your website from anywhere and any computer. Another advantage is that the website builder comes with web hosting services so you don’t need to purchase it and/or set it up separately. Many online website builders are designed for people with little or no coding experience. Website.com is an example of an online website builder.

Languages are the bedrock of any application, website, or software in general. This section will provide you with a brief overview of currently popular languages, how useful they are in the web development landscape, and whether they are used outside of web development.

2.4 JavaScript

In recent years, JavaScript has become the de-facto language of choice when building websites, and one of the most popular ones to build web applications. There are many reasons behind this, including it’s treatment of objects and the ability to build applications quickly. There are some
people who feel that it’s unjustly popular, due to a lot of JavaScript’s perceived shortcomings when compared with more traditional programming languages. JavaScript has some strange quirks and behavior that bothers some engineers. One thing everyone can agree on is that JavaScript is one of the most used languages today, and for all intents and purposes it seems to be staying that way. Knowing JavaScript is essential for any web developer.

To remedy JavaScript’s shortcomings, developers have created a vast amount of libraries, each trying to solve their own problem, or plenty trying to solve the same problem in marginally different ways. Being familiar these libraries has become essential, even if it is merely to be aware that you can use them to ease your development process. Some of the most widely used libraries are as follows: Lodash (utilities, calculations), Ramda (introduces functional programming to JavaScript), and TypeScript (introduces explicitly defining your variable types in JavaScript).

The maintainers of JavaScript language are of course aware of these needs and are adding these sought after features into the core JavaScript standard as time passes. There is one distinction that needs to be addressed, however: the differences between ECMAScript and JavaScript.

2.5 ECMAScript/ES6/ES8

ECMAScript is a standard that defines what a scripting language should do, and JavaScript is an implementation of ECMAScript. ActionScript is another example that has seen little use in recent years, but it follows the same ECMA standard as JavaScript. Web browsers have various engines that compile, “understand,” and run the JavaScript code, but these are usually running the ECMAScript standards.
Since ECMAScript releases a new version every year, it is sometimes difficult for these engines to keep up with the changes to the language. That is why we created the distinction now between ES6/ES2015, and any of the newer versions such as ES9/ES2018. To use the newer features of ES9, tools like Babel have been created, which transpile your code back into an older version of JavaScript (typically either ES5 or ES6) to ensure that it runs on all browsers.

### 2.6 Node.js

It is a JavaScript runtime built on Chrome’s V8 JavaScript engine and is used to execute JavaScript outside the browser. This has allowed JavaScript to also be used as a server (backend) language as well as a frontend language. In fact, Node.js has become the most utilized choice for backend developers due to its ease of use and intuitive behavior. Since you use the same language on the client and the server, it is also allows for smaller and cleaner technology stack.

Node.js is built on an event-driven, non-blocking IO paradigm. This model allows the possibility to run several different processes simultaneously which makes it lightweight and efficient. Node has also become a great tool to communicate with devices such as an Arduino or Raspberry Pi.

### 2.7 Libraries

While Vanilla JavaScript is excellent (i.e., without any libraries or dependencies), utilizing the many open source libraries written in JavaScript can make your work significantly more manageable. Libraries can serve multiple roles, such as utilities (Lodash, jQuery) or tools for making your life as a developer easier (Webpack, i18n).

A few notable open source libraries are:

- **i18n** — pronounced “internationalization” which simplifies localizing and making your app multilingual
- **jQuery** — arguably the most famous JavaScript library, but most of its features are integrated into the core of JavaScript now
- **Moment** — which simplifies working with date and time
- **ThreeJS** — allows you to create, display and animate 3D graphics in the browser
2.8 Hypertext Preprocessor

(or simply PHP) is a server-side scripting language designed for Web development, and also used as a general-purpose programming language. It was originally created by Rasmus Lerdorf in 1994; the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, with the original implementation acting as the de facto standard which other implementations aimed to follow. Since 2014 work has gone on to create a formal PHP specification. [7]

2.9 Ruby

Ruby is a dynamic, object-oriented general-purpose programming language. Ruby is a well-liked and popular choice due to its wide variety of uses and stable releases. On the web, Ruby has been popular since the advent of Ruby on Rails, which we will discuss into in the frameworks section of this article series.
2.10 HTML/CSS

These two languages are both their own, separate languages but given their dependency together, also paired together as one. HTML describes the structure of the page while CSS describes how that has to look. HTML (short for Hypertext Markup Language) is a markdown language used to describe the kind of content a document needs and brings all the interactions and data to one place. CSS (short for Cascading Style Sheets) is a style sheet language used to describe the style of a document concerning colors, typography, placement and more. These are crucial for any web developer to understand.

2.11 ASP.NET

ASP.NET is a server-side programming language designed by Microsoft in order to create dynamic pages back in 2002. .NET has evolved since then and is still used, for example on Site core.

2.12 C#

C# (pronounced C-sharp) is a C-language derivative general purpose programming language designed by Microsoft that takes the object-oriented nature of C++ further. This language can be used online for the server-side but gets commonly used for Windows-based programming, especially for games. One of the best-known uses of C# is in the game engine Unity 3D.

2.13 Java

Java is a popular object-oriented general purpose programming language that is relatively infamous in the web development community. The reason for this is twofold: primarily Java is overkill for most web applications and the time required to implement a change takes a lot longer in comparison to a lot of the previously mentioned languages. Java is still definitely used in the web development community, one famous example being Twitter, which was initially running Ruby on Rails, then switched over to Java to meet their scaling needs.
2.14 Database Options for Web Development

Data is arguably the most important piece of any application or website. Data is raw information; unprocessed and unstyled. This is how you want your data, to ensure that you as the developer have complete control over how the data is processed and styled. That means that the way to store data is crucial for development.

2.14.1 SQL

The most frequently used way to store data is through a SQL database, SQL stands for Structured Query Language, and in essence, that means your data gets stored in a structured manner but is still accessible in its raw form. SQL databases are notable for their relational structure with tables. Think of these tables as data “types” (consider Users and Purchases as two separate tables) and the relationship between them being how many Users purchased a certain item. Each row in this table is a single data point, also referred to as a “record,” and each column is a property of this data point.

Many databases support SQL infrastructures, each with their advantages and disadvantages. One of the most notable is MySQL for your typical, open-source database. MySQL is arguably the most famous SQL database currently in use, allowing a wide availability of tutorials and examples. Another similar option is Postgres, which is slightly more complicated than MySQL but offers a broader range of functionality. MariaDB is a fork (alternate version) of MySQL with a focus to remain free to use. CouchDB, on the other hand, is a specialized SQL database for ease of use and having a scalable architecture for vast amounts of data and is commonly used for “Big Data” needs. [5]
Chapter 3

Security in the Software Development
3.1 Overview

The growing dependency on the range of web application necessitates the development of secure and reliable web applications. Hence, organizations are seeking for a more comprehensive development lifecycle that will aid in reducing security breaches. For a long time, a lot of attention was only given to strengthen the security of networks, which led to attacking strategies from the network layer to the application layer. Besides the internet evolution, the lack of awareness in application vulnerabilities has caused the rise of attacks on the application layer evidently. According to SPI Dynamics, Inc and the internet security threat report from Symantec, more than 70 percent of all hacking even of today occur at the application level, for that reason, organizations are striving to incorporate sufficient measures into an application development lifecycle to make sure that eventually both the application and network are deemed secure under malicious attack attempts.

A development lifecycle entailing secure web application is similar to the general development lifecycle for system application except with the inclusion of adequate security analysis, defenses and countermeasure. There exist many lifecycle models each defining specific method of execution in applications lifecycle. Famous examples of application development lifecycle models are the iterative, agile and waterfall model. In most application development, as detailed in, it is important that an organization first understand the process it must adopt to build secure applications if the processes are not well understood it will be hard to determine its weaknesses and strengths which will consequently impede the continuous improvement of the process. Furthermore, by using a common framework, an organization can set its own standards and security goals to achieve its intended web application. [8]
For better understanding, suitable our study examples, web internet Libyan embassy application in used to show the various stages in specific stage for coding in object oriented programming vs function programming.
3.2 Web server security challenge and defense:

Web server security is a major issue in the current world. There is much exchange of confidential information between hosts, so we cannot avoid the security issue in web service.

Both end-user and web administrators need to worry about the confidentiality of the data transmitted across the web. We can categorize the key web service security requirement into following four intertwined areas:

1. Secrecy
2. Authentication
3. Non-repudiation
4. Integrity control

3.3 Database encryption:

When one talks about web application security, there is a tendency to immediately think about attackers defacing web sites, stealing credit card number, or passwords. So it comes to passwords or data encryption, there is always this big confusing algorithm behind. PHP has a fuss-free password hash and password verify function way. The usage is very straightforward, and they work in a pair.

We used this function into my classes by mixed into oop class when we creating a new user like the admin or the employs.
<?php

class User{
    private $connection;
    private $user_Id;
    private $user_Name;
    private $full_Name;
    private $email;
    private $password;
    private $address;
    private $country;
    //private $language;
    private $status;

    public function __construct($connection)
    {
        $this->connection = $connection;
    }

    public function createUser()
    {
        $query = "INSERT INTO user
                      SET Full_Name = :full_name,
                      Email = :email, Password = :password, Type = :type ";
        $statement = $this->connection->prepare($query);
        // $statement->bindParam("user_Name", $this->user_name);
        $statement->bindParam(":full_name", $this->full_name);
        $statement->bindParam(":email", $this->email);
        $statement->bindParam(":password", $this->password);
        // $statement->bindParam(":address", $this->address);
        $statement->bindParam(":type", $this->type);
        return $statement->execute();
    }

}
```php
public function setFull_Name($full_Name) {
    $this->full_Name = $full_Name;
}

public function setEmail($email) {
    $this->email = $email;
}

public function setPassword($set) {
    $this->password = password_hash($set, PASSWORD_DEFAULT);
}

public function setAddress($address) {
    $this->address = $address;
}

public function setCountry($country) {
    $this->country = $country;
}
```

Fig 3.2 Secure code for encryption

<table>
<thead>
<tr>
<th>ser_Id</th>
<th>User_Name</th>
<th>Password</th>
<th>Full_Name</th>
<th>Email</th>
<th>Address</th>
<th>Countr</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>adhan</td>
<td>$2y$10$5sQqHd4vEgrPn5bOXoEDN.a1fogNgN...</td>
<td>adhan</td>
<td><a href="mailto:adnacsca200426@gmail.com">adnacsca200426@gmail.com</a></td>
<td>janour</td>
<td>LY</td>
</tr>
<tr>
<td>NULL</td>
<td>asa</td>
<td>$2y$10$6iQyjSvDrVovxU5Q0mvPWhu...</td>
<td>asa</td>
<td>asa@asa</td>
<td>aaaa0</td>
<td>AM</td>
</tr>
<tr>
<td>NULL</td>
<td>asd</td>
<td>$2y$10$6iQyjSvDrVovxU5Q0mvPWhu...</td>
<td><a href="mailto:adnacsca200426@gmail.com">adnacsca200426@gmail.com</a></td>
<td>dacdad</td>
<td>A/W</td>
<td></td>
</tr>
<tr>
<td>NULL</td>
<td>asala</td>
<td>$2y$10$6iQyjSvDrVovxU5Q0mvPWhu...</td>
<td>asala</td>
<td><a href="mailto:asala@yahoo.com">asala@yahoo.com</a></td>
<td>Tripoli</td>
<td>AO</td>
</tr>
<tr>
<td>NULL</td>
<td>adhan</td>
<td>$2y$10$6iQyjSvDrVovxU5Q0mvPWhu...</td>
<td><a href="mailto:adnacsca200426@gmail.com">adnacsca200426@gmail.com</a></td>
<td>Tripoli</td>
<td>IN</td>
<td></td>
</tr>
</tbody>
</table>

Fig 3.3 database view
Chapter 4

Programming Paradigms

&

Statistical analysis
4.1. Programming Paradigms

A programming paradigm is a fundamental style of computer programming, associated with a software development methodology, which is a style of explaining specific software engineering problems, different methodologies are more suitable for solving certain kinds of same problems or applications domains. Same for programming languages and paradigms style

The ideas and abstractions used to signify the elements of a program (such as objects, functions, variables, constraints, etc.) the steps that compose a computation (assignment, evaluation, data flow, control flow, etc.). Some of the languages are designed to support one particular paradigm

Smalltalk supports object-oriented programming

Haskell supports functional programming

Other programming languages support multiple paradigms

Object Pascal, C++ and C#, Visual Basic, Common Lisp, Scheme, Perl, Python, Ruby, Oz and F#.

The design area of multi-paradigm languages is to allow programmers to use the best tool for a job, admitting that no one paradigm solves all defaults problems in the easiest or most efficient way.

A programming paradigm can be understood as an concept of a computer system, for example the von Neumann model used in traditional sequential computers. For parallel computing, there are many probable models typically reflecting different ways processors can be interconnected to communicate and share the information. In object-oriented programming, programmers can think of a program as a group of interacting objects, while in functional programming a program can be thought of as a sequence of stateless function evaluations. In process-oriented programming, programmers think about applications as sets of concurrent processes in way that acting upon shared data structures. [6]
4.2 Object-oriented programming

Object-oriented programming (OOP) is a programming paradigm that uses objects, data structures encapsulating data fields and procedures together with their interactions and class also attributes to design applications and computer programs.

It was invented with the creation of the Simula not usually used in mainstream language in 1965, and further developed in Smalltalk in the 1970s, it was software application development until the early of 1990s, many modern programming languages now support object oriented programming (oop).

4.2.1 OOP concepts

A class defines the abstract characteristics of a thing (object) including that thing's characteristics like (attributes, fields or properties) and the thing's behaviors (the operations it can do, or methods, operations or functionalities).

Classes provide modularity and structure in an object-oriented computer program, should typically be identifiable to a non-programmer familiar with the problem domain, meaning that the characteristics of the class should make sense in framework. Also, the code for a class should be relatively self-contained (generally using encapsulation), the properties and methods defined by a class are called its members.

An object is an individual of a class created at run-time through object instantiation from a class. And also the set of values of the attributes of a particular object forms its state. The object consists of the state and the behavior that's defined in the object's class. The object is instantiated by implicitly calling its constructor, which is one of its member functions responsible for the creation of instances of that class.
4.2.2 An attribute

called data member or member variable, is the data encapsulated within a class or object. Attributes are an object’s variables that, upon being given values at instantiation (using a constructor) and further execution, will represent the state of the object. In pure object-oriented programming, the attributes of an object are local and cannot be seen from the outside. In many object-oriented programming languages, however, the attributes may be accessible, though it is generally considered bad design to make data members of a class as externally visible.

4.2.3 A method

is a subroutine that is exclusively associated either with a class (in which case it is called a class method or a static method) or with an object (in which case it is an instance method). Like a subroutine in procedural programming languages, a method usually consists of a sequence of programming statements to perform an action, a set of input parameters to customize those actions, and possibly an output value (called the return value).

4.2.4 Inheritance

Inheritance is a way to compartmentalize and reuse code by creating collections of attributes and behaviors (classes) which can be based on previously created classes. The new classes, known as subclasses (or derived classes), inherit attributes and behavior of the pre-existing classes, which are referred to as superclasses (or ancestor classes). The inheritance relationships of classes gives rise to a hierarchy. Multiple inheritance can be defined whereas a class can inherit from more than one superclass. This leads to a much more complicated definition and implementation, as a single class can then inherit from two classes that have members bearing the same names, but yet have different meanings. Abstract inheritance can be defined whereas abstract classes can declare member functions that have no definitions and are expected to be defined in all of its subclasses.
4.3 Functional Programming:

Functional programming has its origins in the lambda calculus, a formal system developed in the 1930s to study function definition, function application, and recursion. Many functional programming languages can be watched as elaborations on the lambda calculus. LISP was the first of operational functional programming language. In the 1970s the ML programming language was 8979 produced by Robin Milner at the University of Edinburgh, and David Turner developed primarily the language SASL at the University of St. Andrews and after the language Miranda at the University of Kent. Functional programming also finds use in business through domain-specific programming languages like R (statistics), Mathematica (symbolic math), J and K (financial analysis), F# in Microsoft .NET and XSLT (XML).

When we say functions are first-class, it means we can use functions just as we use values in imperative programming. They can be passed as an argument to a function, defined inside another function, and can even be returned as a result. In other words, “functions are values”.[2]

4.3.1 Immutability

Immutability is the behavior that a value of a variable cannot be altered once it is defined. Different languages have different ways of reaching this; in PHP, for instance, the only way to make a variable immutable is to define it as a constant .

4.3.2 Recursion

Recursion is also very prominent in functional programming. In imperative programming, we can use looping constructs like for and for each when we need to manipulate collections or arrays, walking through each element and keeping a temporary variable to hold the current value. But because of immutability, this approach is not possible in functional programming. Recursion is the answer because such book keeping is done implicitly with the call stack.
### 4.3.3 Pure Functions and Referential Transparency

A function is said to be free from side effects if it does not modify the value of an object outside itself, such as a global or static variable, and does not have any input or output effects like writing into file, database, and so on. Such functions are otherwise called pure functions.

The output of a pure function will always be similar for a given set of arguments, which leads to another property called referential transparency. When a function is referentially transparent, we can switch that function with its value without affecting the behavior of the program. All mathematical functions are pure functions, while date functions, rand(), etc. are impure. What exactly do we mean by pure?

- There is a complete separation between the data of a program, and the behaviors of a program
- All objects created in functional programming are immutable (once something is created, it can’t be changed)
- Shared state is avoided (objects do not share scope with other objects).
- Adherence to pure functions (explained below).
4.4 Case Study Design

The goal is to compare the PHP web app development with Object-oriented programming vs. non-oop (function programming) technologies in order to achieve this goal we decompose the goal to two research questions

**RQ1: Are there differences in the development productivity when developing in OOP way, compared to non-OO PHP?**

Object-oriented Programming is the ease of development and ability for other programmer to understand the program after development. Well commented objects and classes types can tell a developer the process that the developer of the program was trying to follow. It can also make additions to the program much easier for the new developer to achieved the goal. From the organizational aspect of the code, object programming is better than non-object programming, with the possibility of working more than one person on the same project for example the connection string in object orient programming in our website we call just one time for all the classes, OOP produces natural software. Natural programs are more understandable. Instead of programming in terms of an array or region of memory, you can program using the terminology of your particular problem. You don’t have to get bogged down in the details of the computer while designing your program. Instead of force-fitting your programs into the language of the computer world, OOP frees you to express your program in the terms of your problem, allows you to model a problem at a programming level, not at the implementation level. You do not need to know how some piece of software works in order to use it: You simply concentrate on what it does. The life cycle of the modern software project is often measured in weeks. OOP aids these quick development cycles. OOP trims time off of the development cycle by providing reliable, reusable, and easily extendable software. Natural software simplifies the design of complex systems. While you cannot ignore careful design, natural software can streamline design cycles because you can concentrate on the problem that you are trying to solve. When you break down a program into a number of objects, the development of each piece can go on in parallel. Multiple developers can work on classes independently. Such parallel development leads to quicker development times.
In NON-OOP we have to declare the connection string for every function that mean more time and also is more complicated. Example in news we content the connection string inside the HTML.

```php
class Database{
    private $host = "localhost";
    private $database = "embassy";
    private $username = "root";
    private $password = "";
    private $connection;

    public function connect(){
        try{
            $this->connection = new PDO("mysql:host=" . $this->host . ";dbname=" . $this->database . ";charset=utf8", $this->username, $this->password);
        }catch(PDOException $exception){
            die("Database error: 
            
            
            " . $exception->getMessage());
        }
        session_start();

    }

    public function getConnection(){
        return $this->connection;
    }
}

$database = new Database();
$database->connect();
```
RQ2: Are there differences in the maintenance productivity when changing an OOp system, compared to non-OOp system?

use Object oriented Programming is because it makes it easy to maintain and modify existing code as new objects are using inheriting characteristics from existing ones. This cuts down the development time considerably and makes adjusting the program much simpler and this technic working with almost 75% from the developing language like .net includes c#.

non-oop The issue that is obvious in function Programming is that if an edit is needed to the program, the developer must modify every line of code that corresponds to the original change in the code. An example would be if at the starting with a program a variable was set to equal the value of 1. If other sub-procedures of the program rely on that variable equaling 1 to function properly they will also need to be changed. As more and more changes may be needed to the code, it becomes increasingly difficult to locate and edit all related elements in the program. A program’s life cycle does not end when you ship it out the door. Instead, you must maintain your code base. In fact, between 60% and 80% of the time spent working on a program is maintenance. Development is only 20% of the equation! Well-designed object-oriented code is maintainable. In order to fix a bug, you simply correct the problem in one place. Since a change to the implementation is transparent, all other objects will automatically benefit from the enhancement. The natural language of the code should enable other developers to understand it as well.
we create a website for Libyan Embassy in Greece because there is no actual website for the Embassy, to study the difference between Object-oriented Programming and function Programming and split the requirements as equally to get the comparison.

The main requirement what we received from the IT department at the Libyan embassy is

1- Registration
2- Log in
3- Issuing passport
4- View all the users
5- Issuing travel document
6- View my profile
7- View travel document
8- Edit travel document
9- Active and deactivate users
10- View passport
11- Sample birth
12- Add news
13- Renew passport
14- Issuing visa
15- View visa
16- View sample birth
17- View renew passport
18- Edit visa
19- View news
20- Edit sample birth
### 4.5.1 spitted the requirements

<table>
<thead>
<tr>
<th>Object oriented</th>
<th>Non object oriented (function Programming)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registration</td>
<td>Sample birth</td>
</tr>
<tr>
<td>Log in</td>
<td>Add news</td>
</tr>
<tr>
<td>Issuing passport</td>
<td>Renew passport</td>
</tr>
<tr>
<td>View all the users</td>
<td>Issuing visa</td>
</tr>
<tr>
<td>Issuing travel document</td>
<td>View visa</td>
</tr>
<tr>
<td>View my profile</td>
<td>View sample birth</td>
</tr>
<tr>
<td>View travel document</td>
<td>View renew passport</td>
</tr>
<tr>
<td>Edit travel document</td>
<td>Edit visa</td>
</tr>
<tr>
<td>Active and deactivate users</td>
<td>Edit sample birth</td>
</tr>
<tr>
<td>View passport</td>
<td>View news</td>
</tr>
</tbody>
</table>

### 4.5.2 Define system requirements :-

The main use case of the web application System functions are fully identified by system analysis, and specify the types of system users.
### Object oriented requirements use case

#### Registration

<table>
<thead>
<tr>
<th>Use case name</th>
<th>Registration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>User</td>
</tr>
<tr>
<td>Relationship status of use</td>
<td>Validate the entered data</td>
</tr>
<tr>
<td>Use case overview</td>
<td>Registration for the new user to access to the profile</td>
</tr>
<tr>
<td>Preconditions</td>
<td>Validate the entered data</td>
</tr>
</tbody>
</table>
| Use case description   | 1. Press sing in button  
                          | 2. Filling the user data  
                          | 3. Press the Registration button |
| Output summary         | Get a username and password |
| Exceptional flow       | You receive an error message if some fields are not filled. |
Login

Use case name: Login
Actors: User
Relationship status of use: Validate the entered data
Use case overview: Registration for the new user to access to the profile
Preconditions: Validate the entered data
Use case description:
1. Press the login button
2. Filling the user name and password
3. Press the login button
Output summary: Log in
Exceptional flow:
You receive an error message if some fields are not filled.
You receive an error massage if the user deactivate
**Use case name** | Issuing passport  
---|---
**Actors** | Administrator, user  
**Relationship status of use** | Validate the entered data  
**Use case overview** | Allows the admin and user to Issuing passport  
**Preconditions** | Login to the system  
**Use case description** | 1. Press services  
2. Press add new passport  
3. Fill the content  
4. Press send  
**Output summary** | Issuing a new passport  
**Exceptional flow** | You receive an error message if some fields are not filled.
<table>
<thead>
<tr>
<th>Use case name</th>
<th>view all the users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>administrator</td>
</tr>
<tr>
<td>Relationship status of use</td>
<td>Validate the entered data</td>
</tr>
<tr>
<td>Use case overview</td>
<td>Allows the admin to obtain to view all the users</td>
</tr>
<tr>
<td>Preconditions</td>
<td>Login to the system with administrator password</td>
</tr>
<tr>
<td>Use case description</td>
<td>.1.Press users</td>
</tr>
<tr>
<td>Output summary</td>
<td>view all the users</td>
</tr>
<tr>
<td>Exceptional flow</td>
<td>You receive an error message if some fields are not filled.</td>
</tr>
</tbody>
</table>
**Issuing travel document**

<table>
<thead>
<tr>
<th>Use case name</th>
<th>issuing travel document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>User</td>
</tr>
<tr>
<td>Relationship status of use</td>
<td>Validate the entered data</td>
</tr>
<tr>
<td>Use case overview</td>
<td>Allows the user to obtain a travel document</td>
</tr>
<tr>
<td>Preconditions</td>
<td>Validate the entered data</td>
</tr>
</tbody>
</table>
| Use case description  | 1. Press the travel document button  
                          2. Filling the user information  
                          3. Press the send button        |
| Output summary        | Extract travel document          |
| Exceptional flow      | You receive an error message if some fields are not filled. |
View my profile

<table>
<thead>
<tr>
<th>Use case name</th>
<th>View my profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>User, admin</td>
</tr>
<tr>
<td>Relationship status of use</td>
<td>Validate the entered data</td>
</tr>
<tr>
<td>Use case overview</td>
<td>Allows the user to obtain to view profile users</td>
</tr>
<tr>
<td>Preconditions</td>
<td>Validate the entered data</td>
</tr>
<tr>
<td>Use case description</td>
<td>1. Press my profile</td>
</tr>
<tr>
<td>Output summary</td>
<td>View my profile</td>
</tr>
<tr>
<td>Exceptional flow</td>
<td>Non</td>
</tr>
</tbody>
</table>
View travel documents

Use case name: view travel documents

Actors: administrator

Relationship status of use: Validate the entered data

Use case overview: Allows the admin to obtain to view the request travel documents

Preconditions: Login to the system with administrator password

Use case description:
1. Press requests button
2. Press travel documents

Output summary: View the request travel documents

Exceptional flow: Non
Edit travel document

<table>
<thead>
<tr>
<th>Use case name</th>
<th>Edit travel document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>User</td>
</tr>
<tr>
<td>Relationship status of use</td>
<td>Validate the entered data</td>
</tr>
<tr>
<td>Use case overview</td>
<td>Allows the user to obtain to edit travel document</td>
</tr>
<tr>
<td>Preconditions</td>
<td>Validate the entered data</td>
</tr>
<tr>
<td>Use case description</td>
<td>1. select the column</td>
</tr>
<tr>
<td></td>
<td>2. press edit</td>
</tr>
<tr>
<td></td>
<td>3. edit the information</td>
</tr>
<tr>
<td></td>
<td>4. Press send</td>
</tr>
<tr>
<td>Output summary</td>
<td>Edit travel document</td>
</tr>
<tr>
<td>Exceptional flow</td>
<td>You receive an error message if some fields are not filled.</td>
</tr>
</tbody>
</table>
4.5.2.2 Non object oriented (function Programming) requirements use case

- Sample birth

<table>
<thead>
<tr>
<th>Use case name</th>
<th>Sample birth notification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>User</td>
</tr>
<tr>
<td>Relationship status of use</td>
<td>Validate the entered data</td>
</tr>
<tr>
<td>Use case overview</td>
<td>Allows the user to obtain to get sample birth notification</td>
</tr>
<tr>
<td>Preconditions</td>
<td>Validate the entered data</td>
</tr>
</tbody>
</table>
| Use case description  | 1. Press the sample birth button
                         2. Filling the information
                         3. Press the send button |
| Output summary        | Sample birth notification |
| Exceptional flow      | You receive an error message if some fields are not filled. |
### Use Case: Add News

<table>
<thead>
<tr>
<th>Use case name</th>
<th>Add news</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>administrator</td>
</tr>
<tr>
<td>Relationship status of use</td>
<td>Validate the entered data</td>
</tr>
<tr>
<td>Use case overview</td>
<td>Allows the admin to obtain to add news</td>
</tr>
<tr>
<td>Preconditions</td>
<td>Validate the entered data</td>
</tr>
<tr>
<td>Use case description</td>
<td></td>
</tr>
<tr>
<td>1. Press news button</td>
<td></td>
</tr>
<tr>
<td>2. Press add news</td>
<td></td>
</tr>
<tr>
<td>3. Filling the information</td>
<td></td>
</tr>
<tr>
<td>4. Press the save button</td>
<td></td>
</tr>
<tr>
<td>Output summary</td>
<td>Add news</td>
</tr>
<tr>
<td>Exceptional flow</td>
<td>You receive an error message if some fields are not filled.</td>
</tr>
</tbody>
</table>
Renew passport

Use case name: Renew passport

A.ctors: User

Relationship status of use: Validate the entered data

Use case overview: Allows the user to obtain to renew passport

Preconditions: Validate the entered data

Use case description:
1. press services
2. press renew passport
3. fill the information
4. Press send

Output summary: Edit travel document

Exceptional flow: You receive an error message if some fields are not filled.
4.6 Sequence diagram

A sequence diagram, in the context of UML, represents object collaboration and is used to define event sequences between objects for a certain outcome. A sequence diagram is an essential component used in processes related to analysis, design and documentation.

4.6.1 Active user :-

![Sequence Diagram]

- User
- <<interface>> users
- <<control>> activate_user
- <<entity>> User
- <<database>> User
- Message: Error.
- Message: Success.
4.6.2 Deactivate user :-

![Diagram of user interface for deactivating a user](image)

- **User** interacts with the interface by clicking the "deactivate_user" button.
- The system triggers the `deactivateUser()` method.
- A message "Success." is displayed if the operation is successful.
- If an error occurs, a message "Error." is displayed.

**Message:** Error.

**Method:** deactivateUser()
4.7 Data collection

The data was collected using experiment. The experiment involved 15 individuals are programming engineers in Libyan embassy in Greece. The 15 persons there were chosen randomly from the PE's at the It department in the embassy. Each involved person was given the designed questionnaire to fill in its question with free and no intervene by the researcher. The questionnaire comprised of 6 questions. Each two questions related one category of the research topic: Development time(dt1,dt2) Time to fix bugs(TFB1,TFB2) Security (s1,s2)

**Development time :-**

What time does it take to write normal code for function in object oriented programming?

1 minute 2 minutes 3 minutes 4 minutes 5 minutes

What time does it take to write normal code for function in function programming?

1 minute 2 minutes 3 minutes 4 minutes 5 minutes

**Time to fix bugs :-**

What time does it take to correct errors when using object oriented programming?

1 minute 2 minutes 3 minutes 4 minutes 5 minutes

What time does it take to correct errors when using function programming?

1 minute 2 minutes 3 minutes 4 minutes 5 minutes
security :-

The degree of security in object oriented programming

1. Very high
2. high
3. medium
4. Poor
5. Very poor

The degree of security in function programming

1. Very high
2. high
3. medium
4. Poor
5. Very poor
4.7.1 Results:

<table>
<thead>
<tr>
<th>subject</th>
<th>DT1</th>
<th>DT2</th>
<th>TFB1</th>
<th>TFB2</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

For statistical ANALYSIS WE used R package (2018) which well-known program language in statistical literature.

- The outlook from R package

```r
quest=read.table("F:\QUEST\QUEST_R.txt",head=T)DTdif=quest[,3]-quest[,2];
TFBdif=quest[,5]-quest[,4];
Sdif=quest[,7]-quest[,6]

cor(quest[,2:7])

Ttest_DT=t.test(quest[,2],quest[,3],paired=T)
Ttest_TFB=t.test(quest[,4],quest[,5],paired=T)
Ttest_TS=t.test(quest[,6],quest[,7],paired=T)
```
Ttest_DT Paired t-test
data: quest[,2] and quest[,3]
t = -7.5965, df = 14, p-value = 2.488e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-2.222718 -1.243948
sample estimates:
  mean of the differences  -1.733333

Ttest_TFB Paired t-test
data: quest[, 4] and quest[, 5]
t = 3.5369, df = 14, p-value = 0.003285
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
0.4985636 2.0347698
sample estimates:
  mean of the differences  1.266667

Ttest_TS Paired t-test
data: quest[, 6] and quest[, 7]
t = 5.5513, df = 14, p-value = 7.138e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
0.7772816 1.7560518
sample estimates:
mean of the differences

1.266667

Ttest_TS

###########################################################################CORRELATION########################################################################### round(cor(quest[,2:7]),2)

    _DT1  DT2  TFB1  TFB2  S1   S2
DT1  1.00  0.53 -0.09  0.00 -0.12  0.00
DT2  0.53  1.00  0.39 -0.04  0.02 -0.19
TFB1 -0.09  0.39  1.00 -0.29  0.59  0.07
TFB2  0.00 -0.04 -0.29  1.00  0.03 -0.26
S1   -0.12  0.02  0.59  0.03  1.00  0.23
S2   0.00 -0.19  0.07 -0.26  0.23  1.00

Means=cbind(mean(quest[,2]),mean(quest[,3]),mean(quest[,4]),mean(quest[,5]),mean(quest[,6]),
mean(quest[,7]))

SDs=cbind(sd(quest[,2]),sd(quest[,3]),sd(quest[,4]),sd(quest[,5]),sd(quest[,6]),sd(quest[,7]))
Names=c('DT1', 'DT2', 'TFB1', 'TFB2', 'S1','S2')

out=rbind(Names,round(Means,2),round(SDs,2))

Names  "DT1"  "DT2"  "TFB1"  "TFB2"  "S1"  "S2"
     "2"  "3.73"  "3.67"  "2.4"  "3.8"  "2.53"
     "0.85"  "0.96"  "0.9"  "0.83"  "0.68"  "0.74"

CV=c(round(SDs/Means*100,2))
DT1 and DT2 42.26, 25.75,  
TFB1 and TFB2 24.54, 34.50,  
S1 and S2 17.79, 29.34

> Summaries = cbind(summary(quest[,2]), summary(quest[,3]), summary(quest[,4]), summary(quest[,5]), summary(quest[,6]), summary(quest[,7]))

> Summaries

Min.    1.000   2.000   2.000   1.000   3.000   1.000
1st Qu. 1.500   3.000   3.000  2.000  3.000  2.000
Median  2.000   4.000   4.000  2.000  4.000  3.000
Mean    2.000   3.733   3.667  2.400  3.800  2.533
3rd Qu. 2.000   4.000   4.000  3.000  4.000  3.000
Max.    4.000   5.000   5.000  4.000  5.000  4.000

boxplot(quest[,2:7], horizontal = F)
legend(5, 5, c(Names), fill = c("yellow", "orange", "blue", "red", "green", "black"))

For supporting our findings research we have discuss some statistical techniques including: descriptive statistics, paired t test for dependent sample, simple correlation and finally using the boxplot for visualization of the difference between paired of the three methods.
4.7.2 Paired Sample T-Test

The paired sample t-test, sometimes called the dependent sample t-test, is a statistical procedure used to determine whether the mean difference between two sets of observations is zero. In a paired sample t-test, each subject or entity is measured twice, resulting in pairs of observations. Common applications of the paired sample t-test include case-control studies or repeated-measures designs.

In Table 1 the descriptive statistics of the considered variables are shown. The descriptive statistics are including the following measures: the mean, standard deviation, coefficient of variation and the largest and smallest values.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean+/ sd</th>
<th>Coefficient of variation</th>
<th>Min and max values</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT1</td>
<td>2.00 +/- 0.85</td>
<td>42.26%</td>
<td>(1,4)</td>
</tr>
<tr>
<td>DT2</td>
<td>3.73 +/- 0.96</td>
<td>25.75%</td>
<td>(2,5)</td>
</tr>
<tr>
<td>TFB1</td>
<td>3.67 +/- 0.90</td>
<td>24.54%</td>
<td>(2,5)</td>
</tr>
<tr>
<td>TFB2</td>
<td>2.40 +/- 0.83</td>
<td>34.50%</td>
<td>(1,4)</td>
</tr>
<tr>
<td>S1</td>
<td>3.80 +/- 0.68</td>
<td>17.79%</td>
<td>(3,5)</td>
</tr>
<tr>
<td>S2</td>
<td>2.53 +/- 0.74</td>
<td>29.34%</td>
<td>(1,4)</td>
</tr>
</tbody>
</table>
Table 2 shows the correlations between each two variables.

Table 2: the observed simple correlations

<table>
<thead>
<tr>
<th></th>
<th>DT1</th>
<th>DT2</th>
<th>TFB1</th>
<th>TFB2</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT1</td>
<td>1.00</td>
<td></td>
<td>-0.09</td>
<td>0.00</td>
<td>-0.12</td>
<td>0.00</td>
</tr>
<tr>
<td>DT2</td>
<td></td>
<td>1.00</td>
<td>0.39</td>
<td>-0.04</td>
<td>0.02</td>
<td>-0.19</td>
</tr>
<tr>
<td>TFB1</td>
<td></td>
<td></td>
<td>1.00</td>
<td>-0.29</td>
<td>0.59</td>
<td>0.07</td>
</tr>
<tr>
<td>TFB2</td>
<td></td>
<td></td>
<td></td>
<td>1.00</td>
<td>0.03</td>
<td>-0.26</td>
</tr>
<tr>
<td>S1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.00</td>
<td>0.23</td>
</tr>
<tr>
<td>S2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.00</td>
</tr>
</tbody>
</table>

As we can see from Table 2 the largest correlation is between DT1 and DT2 which is also positive. While the correlation between TFB1 and TFB2 is negative. On the other hand although the correlation between S1 and S2 is also positive but is less than half way of the correlation between DT1 and DT2.
4.7.3 Boxplot

A method for graphically depicting groups of numerical data through their quartiles. Box plots may also have lines extending vertically from the boxes (whiskers) indicating variability outside the upper and lower quartiles, hence the terms box-and-whisker plot and box-and-whisker diagram

Figure 1 shows the distributions of the six methods. The figure gives a good view to compare the methods. One can observe that the position of DT1 box (left side) is very low with less height compared to DT2, which indicates that DT1 is more skewness to the left i.e. it tends to the small values. In terms of TFB we can observe that the location TFB1 goes up while the TFB2 location goes down. The case is same for S where S2 is identical to the TFB2. While S1 is similar to TFB1 with little difference about the mean.

Paired t test

We have used the paired t test for testing the following hypotheses:

1. $H_0$: There is no difference in true means between DT1 and DT2 versus $H_1$: There is a difference between them.
2. $H_0$: there is no difference in true means between TFB1 and TFB2 versus $H_1$: There is a difference between them.
3. \( H_0 \): there is no difference in true means between S1 and S2 \textit{versus} \( H_1 \): There is a difference between them.

The results summary of the carried testes are shown in Table 3. As p-value is less than 0.001, we can see that there a highly significant difference between each pair of the considered three methods.

Note that we take the value of significance level to be 0.05. So for any of the three carried tests if the p-value less than 0.05, the \( H_0 \) will not be accepted which means the difference is significant. Moreover, if the observed p-value less is than 0.01 than we conclude that there is a highly significant between the two methods.

<table>
<thead>
<tr>
<th>The difference between</th>
<th>T calculated value</th>
<th>p-value</th>
<th>Mean difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT1 and DT2</td>
<td>2.488e-06 (**)</td>
<td></td>
<td>-1.73</td>
</tr>
<tr>
<td>TFB1 and TFB2</td>
<td>3.53, with 14</td>
<td>0.003285 (**)</td>
<td>1.27</td>
</tr>
<tr>
<td></td>
<td>degrees of freedom</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1 and S2</td>
<td>5.55, with 14</td>
<td>7.138e-05 (**)</td>
<td>1.27</td>
</tr>
<tr>
<td></td>
<td>degrees of freedom</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The symbol (**) is used to indicate the p-value is less than 0.001 i.e. there is a highly significance difference between the two methods. Here is case for all the three tests.
4.8 Agile Methodology

Is one of the latest and best ways of successful software development and management unlike conventional and classical methods in project management. Agile methodology governs a set of laws and practices that enable developers and entrepreneurs to obtain the final product as a web site or application at the lowest cost in record time. (Hayek, 2015)

What distinguishes Agile methodology is that it hits most of the prevailing concepts in project management and allows for adaptability to changes that occur to the product or project. Agile always places greater emphasis on the end-user product, where the end-user is the judge and the measure of the success or failure of the product.

Scrum Definition
A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest value.
4.9 Results

- **development time:**
  
  the time we spent using object programming is much less than function programming like I used classes so when we use this class again just call the name, but for the function programming in specially in php you have to write all the code in the same page so its takes time for the coding. Functional programming provides the advantages like efficiency, lazy evaluation, nested functions priority, bug free code, parallel programming. In simple language

- **time to fix bugs:**
  
  bugs detection is faster in function programing but slower in the maintenance process - but in object oriented programming faster detection errors and faster maintenance process. so the time to fix this bugs is depend how is the code is look in my website I used classes for every requirement even the HTML design like master page to have the same style in whole pages. use Object-oriented Programming is because it makes it easy to maintain and modify existing code as new objects are created inheriting characteristics from existing ones. It can also makes additions to the program much easier for the new developer. From the organizational aspect of the code
Conclusion

Finally in this theses we compared the object oriented programming with function programming, on website was created to the Libyan embassy in Greece. that started with Web Development Languages and Technologies description for the technologies in web development and Programming Paradigms, split the requirements to analysis to use case and sequence diagram, we decompose the goal to two research questions and the result for this question that object oriented programming is better than function programming in php from the development time and also time fixing bugs, for that reason OOP offers encapsulation, inheritance, polymorphism and composition, it helps build much more extendable, reusable, maintainable, and organized application. The advantages are more evident in larger programs, which are pretty much impossible to work on with function style, especially when teamwork is required.
Discussion

The primary purpose of this study was examine the differences between oop and non-oop ,earlier research suggests that there are benefits to use oop on programming using c++ or java but not specific in php that relevant to web programing.

Our result of the idea behind OOP was to make programs easier to write and maintain by allowing problems to be modeled in terms of objects which represent the entities or concepts that the application deals with , benefits of object oriented design , Readability , Ease of changes - Make changes in once place rather than many Stability - Less chance for slightly different code being produced in multiple places OOP . Object-oriented programs are typically slower than function programing at fixing bugs , as they typically require more instructions to be executed . Quite obviously, those on team OOP argue that OOP is a better approach to creating programs, while those on team FP argue that FP is better. How so? Team OOP argues that the concept of inheritance (new objects taking on the attributes/methods of existing objects letting us reuse more code) and encapsulation (the data and methods related to a certain object being bound together, creating independent, protected entities) makes it easier to manage and manipulate data. Team FP argues that the separation of data and methods, as well as the high level of abstraction leave less room for errors. The big thing that I think makes functional programming bad is that you’re going to quickly come up against your not having a PhD in mathematics or computer science or some logic-heavy discipline. I have a pretty good understanding of functional programming (FP) for a non-functional-programmer. But there are kind-of-important buzzwords I couldn’t begin to understand which people talk about a lot when they talk about functional programming. “Functor”, “monoid”, etc are much harder to wrap your head around than objects. The other related disadvantage, is that you’re doing a lot of putting functions into functions into functions. And just as I’m not a computer, I’m also no math wiz. So all the complex operations that functional programmers do a lot are pretty abstract to me and I can feel overwhelmed. I know this was true of other paradigms for me, and it’s just a bigger learning curve, but it’s certainly nothing to shrug off quickly.
Future work

Publish the website online

After 6 to 12 months depending on the amount of data I will collect, I will try to use one of Data mining techniques in order to transfer this dissertation into a publication.

Note: I spent more than 2 months for coding the website. I used all techniques from design like master page for all the pages and also the menu in HTML pages is dynamic. I used classes for every inch in this project.
Bibliography


Appendices

Implementation in php

```php
include_once("class/database.php");
include_once("class/news.php");

(news = new News($database->getconnection()));
```

Index : home page

```html
<!DOCTYPE html>
<html>
<head>
  <?php include_once("includes/head.php"); ?>
</head>
<body>
  <!-- header-section-starts-here -->
  <div class="header">
    <?php include_once("includes/header.php"); ?>
  </div>
  <!-- header-section-ends-here -->
</body>
</html>
```
Log in (object oriented programming) :-

![Sign In Form]

Username
Password

LOGIN
Forgot Password?

HOME
CREATE AN ACCOUNT

2018 © libya. Admin.
<?php

include_once('./class/inc.php');
$user = new user($database->getConnection());

if($user->isLoggedin()){
    header("Location: index.php");
}

if(isset($_POST['password'])){  
    $loginResult = $tempUser->login($_POST['email'], $_POST['password']);
    if($loginResult == "TRUE"){
        header("Location: index.php");
    }
}

if(isset($_POST['signup'])){  
    $user->setUser_Name($_POST['user_Name']);
    $user->setFull_Name($_POST['full_Name']);
    $user->setEmail($_POST['email']);
    $user->setPassword($_POST['password']);
    $user->setAddress($_POST['address']);
    $user->setCountry($_POST['country']);

    if($_POST['country'] == 'LY')
    {
        $user->setType(2);
    }else{
        $user->setType(3);
    }
**Request birth (Function programming)**

```php
<?php
    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = '';
    $dbname = 'embassy';
    $conn = mysqli_connect($dbhost, $dbuser, $dbpass, $dbname);
```
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    while($row = mysqli_fetch_assoc($result)) {

    }?
</tr>
<tr>
    <td><?php echo $row['Name']; ?></td>
    <td><?php echo $row['Father_name']; ?></td>
    <td><?php echo $row['Mother_name']; ?></td>
    <td><?php echo $row['Place_birth']; ?></td>
    <td><?php echo $row['Date_Of_Birth']; ?></td>
    <td><?php echo $row['Hospital_name']; ?></td>
    <td><?php echo $row['Sex']; ?></td>
    
</tr>
</tr>
<?php

} else {
    //echo "0 results";
}

mysqli_close($conn);

</tbody>
User services
Convert the lists to PDF,

Passport requests from the admin control
## Tables and pictures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig 1.1</td>
<td>Atypical client serves Architecture</td>
<td>10</td>
</tr>
<tr>
<td>Fig 3.1</td>
<td>Secure application development lifecycle</td>
<td>25</td>
</tr>
<tr>
<td>Fig 3.2</td>
<td>Secure code for encryption</td>
<td>26</td>
</tr>
<tr>
<td>Fig 3.2</td>
<td>Secure code for encryption</td>
<td>27</td>
</tr>
<tr>
<td>Fig 3.3</td>
<td>database view</td>
<td>27</td>
</tr>
<tr>
<td>Fig 4.1</td>
<td>connection string in object orient programming</td>
<td>34</td>
</tr>
<tr>
<td>Fig 4.2</td>
<td>connection string in non-object orient programming</td>
<td>35</td>
</tr>
</tbody>
</table>