Global Journal of Programming Languages

Volume No. 8

Issue No. 2

May - August 2023

ENRICHED PUBLICATIONS PVT. LTD

S-9, IInd FLOOR, MLU POCKET,
MANISH ABHINAV PLAZA-II, ABOVE FEDERAL BANK,
PLOT NO-5, SECTOR-5, DWARKA, NEW DELHI, INDIA-110075,
PHONE: - + (91)-(11)-47026006

Global Journal of Programming Languages

Aims and Scope

Global Journal of Programming Languages is a peer-reviewed Print + Online journal of Enriched Publications to disseminate the ideas and research findings related to all sub-areas of programming languages. It also intends to promote interdisciplinary researches and studies in computer science maintaining the standard of scientific excellence. This journal provides the platform to the scholars, researchers, and PHD Guides and Students from India and abroad to adduce and discuss current issues in the field of programming and Computer Sciences.

Global Journal of Programming Languages

Global Journal of Programming Languages

(Volume No. 8, Issue No. 2, May - August 2023)

Contents

A Methodology for Software Cost Estimation using UML Formulation

[1] Muhammad Farhan, [2] Abdul Karim, [3] Rana M. Amir Latif, [4] Tausif Anjum
[1,3]Department of Computer Science COMSATS University Islamabad, Sahiwal Campus, Pakistan
[2,4]National College of Business Administration & Economics Rahim Yar Khan, Pakistan
E-mail: [1]farhansajid@gmail.com, [2]abdullkarim@hotmail.com, [3]ranaamir10611@gmail.com,
[4]ranaomi178@gmail.com

ABSTRACT

Software cost estimation is the most critical and troublesome piece of Software Project Management. Since cost estimation is included in the whole procedure of Software Project Management, it is important to comprehend the relationship between estimation and Software Project Management. In the field of programming designing brought together Modeling Language (UML) is a long-standing approximately useful representative dialect. To make the visual models of item situated programming concentered frameworks the unified modeling language incorporates the documentation procedures to make visual models of item situated programming concentrated frameworks. In the designing of the software, the UML class outline is the static configuration graph, by demonstrating the framework"s classes these configurations depict the structure of the framework, connection among the classes, processes or methods and their characteristics. The point of this examination is to build up a device to evaluate the cost of a product utilizing UML. The achievement of a product undertaking is dictated by different components that have connected each other in the task. In this exploration, we talk about the utilization of capacity to measuring the cost evaluated of a product framework, the scholastic framework an instructive foundation that is looked at on the estimation of the model utilizing the UML based article model with the Data Flow Diagram (DFD) based organized model.

Keywords- Information Technology, Information Systems, Software Engineering, Cost Estimation, Unified Modeling Language.

I. INTRODUCTION

Arranging is one of the most vital procedures in venture administration: great arranging prompts an effective task. In any case, great arranging requires a precise estimation of the extent of the venture, the exertion of staff power, cost and calendar. The objective of a decent estimation is to give the cost and timetable to the administrator. If the real cost of a venture is out of spending plan, regardless of how to finish the item is, the undertaking is viewed as a disappointment because there is no benefit. Also, if the guaranteed deliverables are late to meet its conveying date, it is still viewed as disappointment claiming the agreement is not satisfied, and the partner may manage misfortune. Likewise, checking and controlling procedures are iterative and estimation is required between the procedures to discover the deviations from the arrangement. Hence, estimation is required all through the whole venture administration process [1, 2]. Assessments of cost and calendar in software activities depend on an expectation of the extent without bounds framework [3-5]. Sadly, the software calling is famously off

base when evaluating cost and calendar [6]. Preparatory evaluations of exertion dependably incorporate numerous components of frailty. Dependable early gauges are hard to get in view of the absence of nitty-gritty data about the future framework at an early stage [7]. Be that as it may, early gauges are required when offering for an agreement or figuring out if a task is plausible in the terms of a cost-advantage examination [8, 9]. Since procedure expectation guides basic leadership, a forecast is helpful just on the off chance that it is sensibly precise. Data parade-size estimation incorporates recognizable proof and characterization of classes, assessment of multifaceted nature level of every class utilizing 24 distinctive kinds of drivers, assessment of the Total Unadjusted Class Point (TUCP) and assessment of specialized unpredictability element estimation. After every one of these figuring, we can ascertain Final class point assessment.

The purpose of this research is to find a system to estimate cost and duration for compound implanted software projects, using The Unified Modeling Language design and a hierarchical "divide and conquer" paradigm. Programming progression has made a huge amount of advancement from general programming change, which is depicted by the composed programming perfect model introduced in the late '60s and early '70s to contemporary headway drills, which portray an item order as associating, free fragments that get the robust point of view of programming progression.

In these models, programming effort is foreseen at the gigantic grained structure level. Accordingly, these general measures do not consider the entirety and multifaceted nature of elaboration obliged, centering rather on the measure of data got to or moved, because to predict effort in Component-Based Software Development (CBSD) a fine-grained procedure is obliged therefore, UML is used for exhibiting these schemas. Its diverse estimation procedures encourage the estimation of benefits like efforts, cost and so on.

II. LITERATURE REVIEW

Examinations of cost and timetable in software exercises rely on upon a figure of the measure without limits structure. Disastrously, the software calling is broadly incorrect when assessing cost and logbook. Preliminary examinations of effort constantly fuse various parts of vulnerability. Strong early gages are difficult to secure considering the nonappearance of point by point information about the future structure at an initial stage. In any case, initial gages are required when offering for an understanding or making sense of if a wander is possible in the terms of a cost-advantage examination. Since process estimate guides fundamental initiative, a desire is significant recently in case it is sensibly correct [10]. Estimations are vital to assess the status of the wander, the thing, the strategy, and resources. By using estimation, the wander can be controlled. By choosing fitting proficiency values for the area estimation environment, known as change, it is possible to attempt desires using procedures or gadgets.

Top-down Approach

In the top-down procedure, the technique is used to breaking down the skeleton its compositional subsystems. A framework of the system is definite in this strategy, characterizing yet not indicating any first-level subsystems. Each subsystem is again refined in the paramount unobtrusive component. The top-down assessing system is in like manner named as Macro Model. Using this methodology, general cost estimation for a wander is dead set using the overall properties of the item undertaking, and after that, the assignment is divided into diverse low-level parts. Putnam model is delivered using this system [11]. Exactly when overall properties are known, this framework is most fitting for in front of calendar cost estimation. In front of the calendar time of programming change, it is uncommonly significant considering the way that there is no point-by-point information access.

Putnam Model

The Putnam model is an accurate software cost estimation model. It delineates the time and effort required to finish an item wander of specified size. It is one of the soonest model made and is among the most comprehensively used. It is to a great degree unstable to the change time: lessening the headway time can phenomenally extend the individual months needed for development. Using the Putnam model, the SLIM instrument is made. Man-month required for headway is given by the form [12]. A parameter decided on the premise of authentic information of the past ventures and reliant on the earth.

Bottom-up Approach

It underlines coding and early testing, which starts when the first module has been specified. This approach runs the hazard that modules might be coded without having a clear thought of how they join to other parts of the framework, and such joining might not be as simple as first thought. Primary benefit of the lowest part up methodology is re- convenience of code. Utilizing base up assessing strategy, the expense of every product segments is assessed and after that consolidates all the results. It goes for building the appraisal of a framework from the learning gathered from the little programming parts and their communications. COCOMO model is created utilizing this met.

COCOMO Model

The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation model. It is a relapse model which utilizes the premise relapse equation with parameters that are determined from authentic task information and present and additionally future undertaking aspects. Programming advancement eff ort is ascertained regarding system estimate by COCOMO [13]. Project size is evaluated in many source lines of code (SLOC). COCOMO accepts that the framework and programming necessities have been defined as of now and that these prerequisites are stable. The

COCOMO model has a very simple form Where K_1 and K_2 are two parameters dependent on the application and development environment:

$$\text{MAN-MONTHS} = K_1 \quad (\text{Thousands of Delivered Source Instructions})\ K_2$$

III. METHODOLOGY

An approach to assessing uncertainty for a set of estimates based on the empirical distribution of estimation accuracy. This methodology includes the following variants and extensions: Employ historical data on the accuracy of low-level tasks to produce probability distributions for aggregated effort. This can be useful to make release level PIs based on task- level accuracy data. Monte Carlo simulation is employed to produce such distributions. Allow for smoothing of empirical data sets, i.e., to fit empirical data to a parametric distribution, as the fitted data may be a better representation of the underlying uncertainty.

The approach contains four steps. The first step is to select an appropriate set of historical data, with pairs of estimates and actual effort. To produce meaningful results, we need to use the data which is most likely to have the same estimation errors as the task we are estimating. The next step is to input all task estimates for the project period to be estimated. Our approach is meant to be a complement to whatever estimation method is used. Because it uses the empirical accuracy distribution for assessing uncertainty it does not matter what kind of estimation method is used or how accurate the estimates are. If the estimates are historically very inaccurate, then the resulting effort PIs will be wider, but it should not affect the accuracy (hit rate) of the produced effort PIs. The following measure is used for estimate accuracy as shown in "Equation 1".

$$Accuracy = \frac{Actual\ effort}{Estimated\ effort} \qquad (1)$$

The third step is an automated aggregation of effort by randomly drawing task outcomes from the empirical distribution. For each task, a random task from the selected empirical data set is drawn. Accuracy factor is multiplied with the task estimate of the drawn task and sum up this into the total iteration of the simulation. By using Monte Carlo simulation with this approach, a large number of potential outcomes are generated [14]. The set of simulated outcomes represents a probabilistic distribution of actual effort. The fourth step is using the percentiles of the probability distribution to propose a PI based on the requested confidence level. Using the probabilistic distribution from the Monte Carlo simulation we can generate effort PIs. If the requested confidence is 90%, then the minimum value is set to the 5% percentile of the distribution and the maximum effort is set to the 95% percentile of the distribution.

IV. RESULTS AND DISCUSSION

In playing out the cost estimation, our device considers parameters, for example, Functional Specifications, Operational Constraints, and Organizational Maturity to handle a specific business necessity or venture. This instrument is a combination execution or a quintessence of certain software estimation and estimation systems that assist a software associated with evaluating and break down basic software measurements, for example, Effort, Time, People, and Cost, all of which are fundamental to enhancing turnaround time and achieving authoritative development. Applying regression technique over forty datasets [28] of CP1 and CP2. The effort is calculated as shown "Fig. 1" and "Fig. 2".

Function point Evaluation is Represented by Thomas Fetke Et-al at a Strategy for mapping the exact question prepare system. Authors purpose in the olution that mapping the utilization cases into the function point show utilizing an arrangement of compact decides that bolster the estimation procedure. These mapping principles depend on the IFPUG counting practices manual that is characterized by slandered FPA. In the utilization of case model, the idea of the on-screen character is more extensive then the idea of outer applications and the clients in FPA, there can't be a balanced client to outside application and mapping of performing artists. In any case, every client of the framework is characterized as an on-screen character. In a similar way, all applications which speak with the framework under thought should likewise show up as performing artists. Kerner"s utilization case point strategy could be compared. The utilization case demonstrate does not give enough data on how to number a utilize case as indicated by capacity point rules and the level of detail in the utilization case model may change. Thus, as in Karner's plan, the usage instances have to be reflected at detail remembering the ending target to own the ability to various trades as exhibited at "Fig. 3".

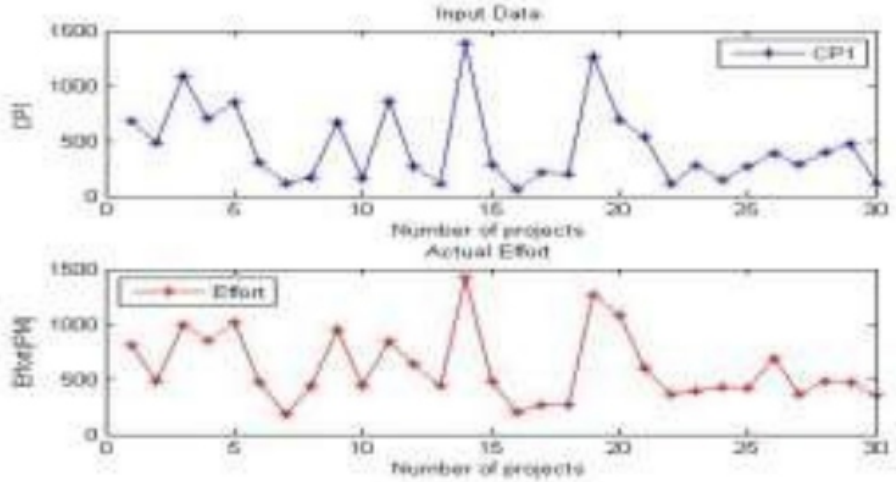


Fig.1. Eff ort Calculation Using CP1.

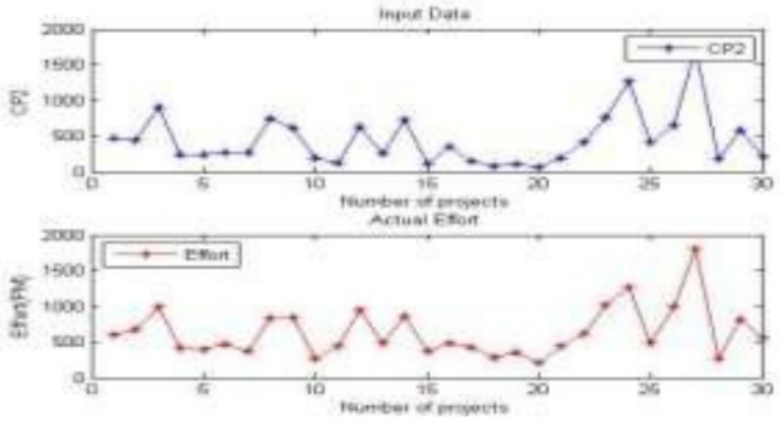


Fig. 2. Eff ort Calculation Using CP2.

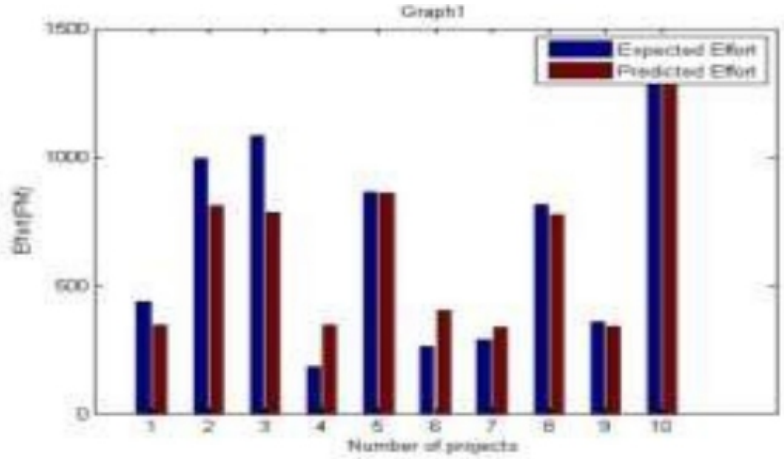


Fig. 3. Result obtained after regression.

Case Study

Even the customer's designers weren't associated in applying instance demonstrating much significantly less programming wisdom compared to no expertise anything with Rational Rose or even RUP along with those who descend out of the applications company. But merely one was connected in logical climbed the artists by the applications company needed a few instructions of query ordered displaying. The team required to embrace the exact educational RUP that is clearly one of those challenges and temptations. They're had to access fast right down seriously to new and programming into bringing fantastic steps of instruction. Programming in C++ over the UNIX point has been a wellspring of dissatisfaction with those classes, also, an expected danger, whilst the engineers were so fresh into those apparatus along with also UNIX.

Data Collection

Even the UML types for each among these subsystems were shipped from an email by the coach within the partnership earlier I spoke with her. From such versions, I accumulated that the corresponding advice: usage instance graphs and literary depictions of usage examples, category outlines, a couple of

saying relocation charts, and also group graphs. Specialized and environmental Many sided excellent elements had been put with the trainer within the length of one particular assembly. The partnership administrator supplied everybody of many evaluations after I'd enrolled the key evaluations like I desired to eliminate one-sided includes about. At this time if allotting traits into the technical factors, the coach believed that the demand of an ingredient to describe which the application form had been infected intricate. Lamentably, there aren't any methods to improve the structures of their natural or specialized parts. Once more, the variable T10 "Simultaneous" introduced an issue. Since a variable was expected to portray that the four subsystems handled information in parallel and that it was a continuous arrangement, this element could be utilized as shown in “Fig. 5”. Different processors are utilized for continuous frameworks requiring a ton of computational execution. Frequently a lot of information must move between information securing, information show, and processor sheets as appeared in “Table. 1” and “Fig. 4”.

Table. 1. Estimates made with the proposed method

Subsystem	UUCP	UC Estimate	Without TCF	Expert Estimate	Actual effort
1	163	4113	3994	3450	3723
2	161	4062	3920	2615	3665
3	145	3659	3553	3235	3835
4	123	3129	3038	3300	2710
Total	593	14965	14528	12600	13933

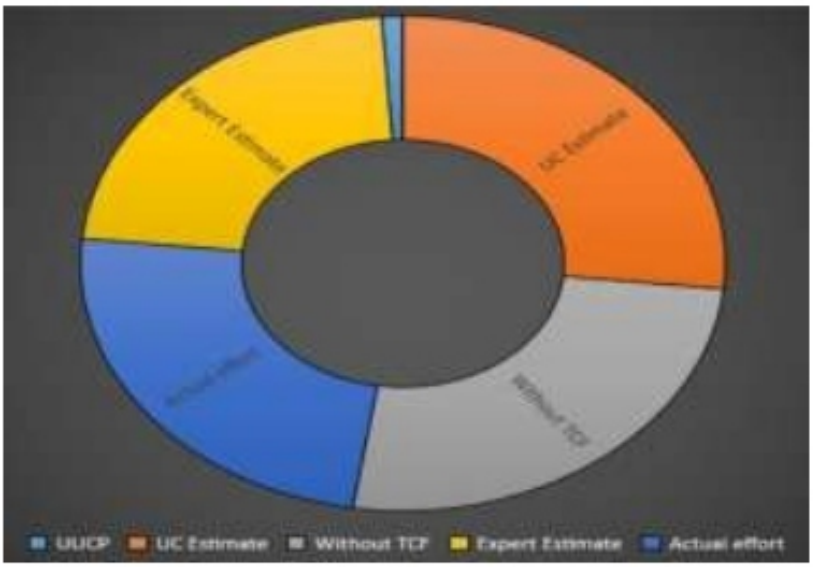


Fig. 4. Cost estimates distribution using the proposed method.

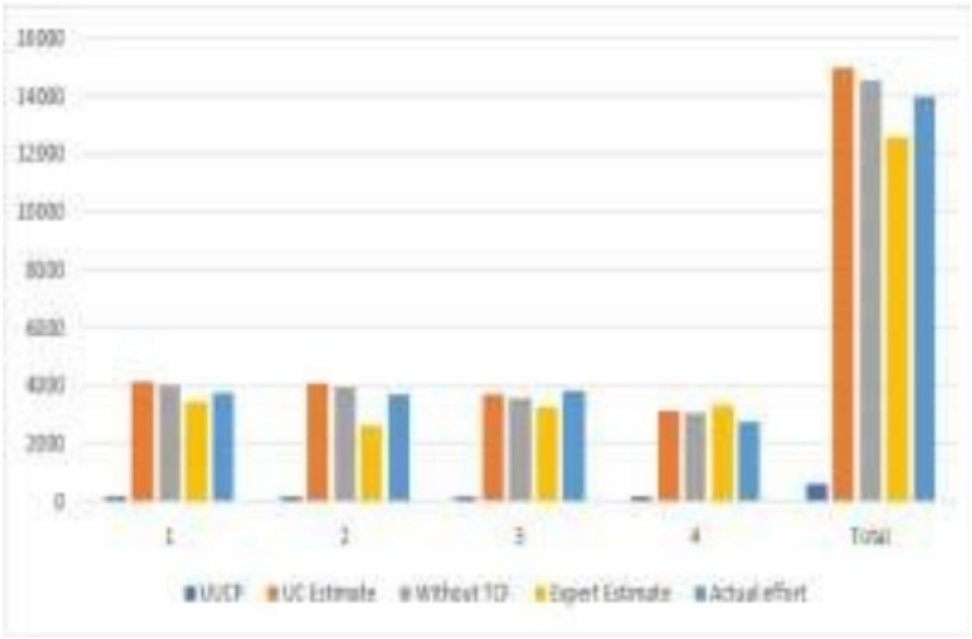


Fig. 5. Cost estimation splits.

Computing the Estimates for the case study You will find added usage cases along with also a couple collapses. Karner approved not only for such usage instances but alternatively, within this particular application, a substantial region of the intricate utility has been hauled in comprised along with amplifying usage instances. Perhaps not including these can mean critical viability wouldn't be anticipated. I like this opted to assess all of the growing and contained usage instances. Their Many-sided caliber for several of the sub-frameworks. The tests and without the technical unpredictability compute genuine effort and grasp indicators. The onscreen figures and usage instances were assessed, anticipated, and also the qualities moved to some dictionary using the natural and specialized factors. Two tests were registered, without and with the technical multi-faceted character compute. Even the UUCP is that the unadjusted use case targets obtained from such as the unadjusted onscreen character weights, UAW, as well as the unadjusted use Circumstance weights UUCW:

$$UUCP = 1*1+ 3*2+ 2*3 + 10*5 + 7*10 + 2*15 = 163$$

The well-balanced usage case targets ended up worked outside and raised by team hours a usage case stage, that with this particular situation was 28, inferable from staff knowledge and steadiness hauled from the environmental parts. This afforded an estimate of 4133 team hours using a TCF and also 3994 team hours minus TCF. I assessed that the trades out of the usage instance ways. You will find 1 2 straight-forward and 9 media usage instances. The usage instances appeared as if they'll not have ample depth. By way of instance, measure two „agreeing on swap to suitable sub-routine', can comprise a few usefulness which isn't viewed. I thusly focused the course charts to assess classes which perform such

usage instances, to affirm whether the Many-sided grade has been ideal. An easy usage instance is implemented by underneath 5 categories, and also moderate utilize instance from 5 to ten courses. The majority of the usage instances were daunted by Under-10 lessons, also lots of under 5, therefore that I decided utilization case many-sided grades has been expelled efficiently. There had been not any group graphs through that I really could concur the tallies had been very straight, nevertheless that really is only one of those vulnerabilities of this plan. The UUCP is obtained from Your recipe.

$$UUCP = 0*1+ 4*2+ 0*3 + 12*5 + 9*10 + 0*15 = 161$$

The gauge for subsystem 2 was 4062 staff hours with TCF, and 3920 staff hours without TCF as shown in "Table. 2" and "Fig. 6".

Table. 2. Estimates produced by Optimization

Subsystem	Classes	Estimate	Expert Estimate	Use cases
1	26	3092	3450	19
2	57	5095	2615	21
3	35	3245	3235	13
4	43	3529	3300	10
Total	161	14961	12600	63

"Exertion" indicates real effort, „graspest.' is grasp appraise, 'UC judge' could be your usage instance targets plan with TCF,"'UC without TCF' could be your usage case concentrates technique devoid of. The admiration obtained in 'Substantial Small Business Architect' is exactly the Exact Same regarding the usage case concentrates method together with all the TCF when effort peruse situation stage is exactly the Exact Same the usage situation targets assess with no TCF, 14528 employees‟ hours," would be the closest into this real effort of 13933 Hrs.

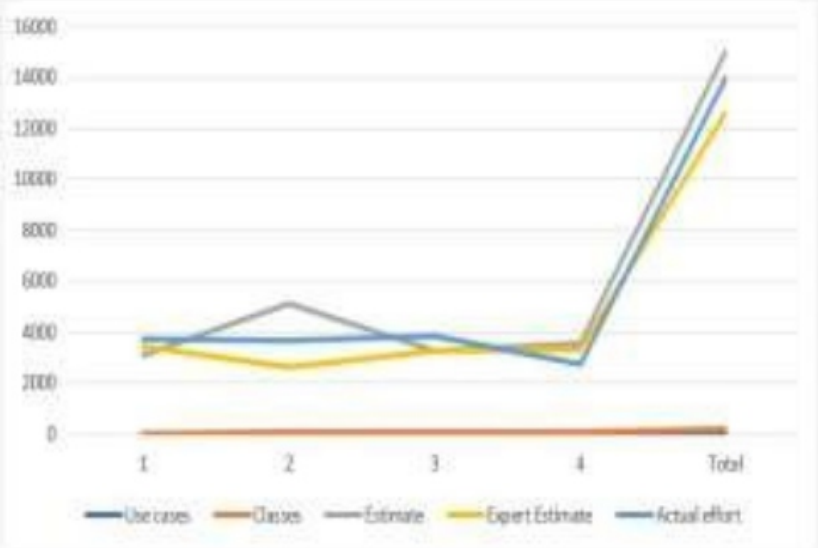


Fig. 6. Estimates produced by Optimization.

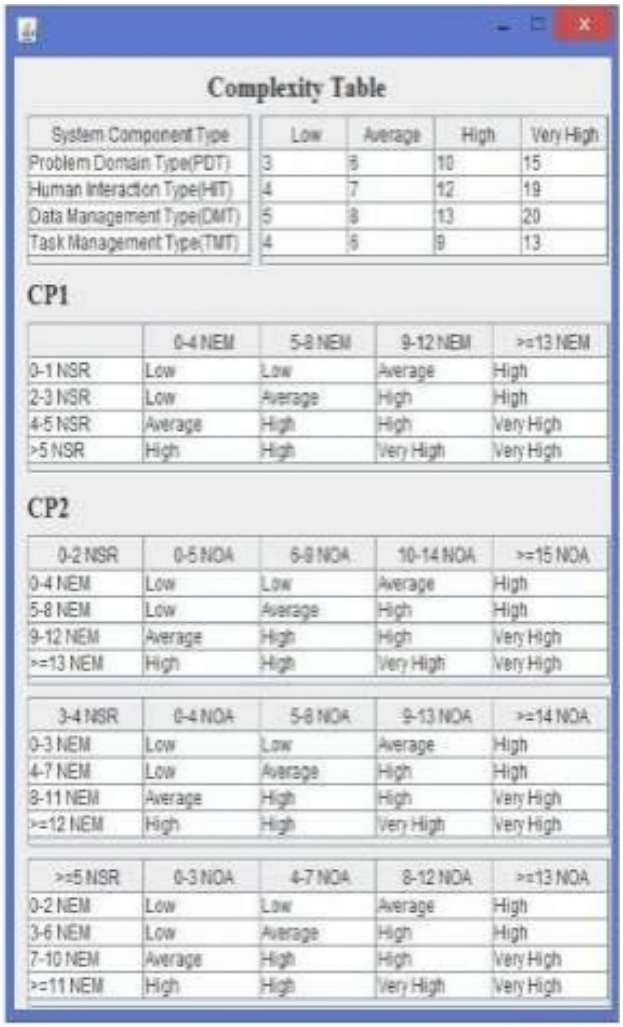
The judge provided by “upgrade" is basically exactly the very same while the judge generated with all the usage case concentrates procedure with all the TCF. The results are 14961 and also 14965 workers hrs. Whatever the scenario, often, no these kinds of ventures exist and also appointing effort per usage case stage is combined these traces in perspective of the puzzle. "Improve" requires courses because of donation and also the usage instances and can be Worthier of creating indicators after in the enterprise if more info is more available.

Threats to Validity

The utilization case focuses technique represents all venture exercises like arranging, investigation, outline, extend administration, testing and so forth in the aggregate assessments. In the tasks of both contextual investigations, extend administration was excluded from the real exertion. In Case Study, this action was essentially excluded while enrolling in exertion. In Case of Study, advancement of the four subsystems was a piece of a bigger venture. Extend administration was in this way enlisted for the venture overall and was excluded from the real advancement exertion for the four subsystems.

The subsequent evaluations were exceptionally exact in both contextual investigations. However, since venture administration was not checked, these outcomes are to some degree indeterminate. On the off chance that venture administration had been a piece of the real exertion, both technique and instruments would have undersized the frameworks. Notwithstanding, as there was no data about the exertion for venture administration, it is impractical to state how much the frameworks would have been thought little of. The outcomes appeared were along these lines utilized as a part of the further reviews.

The traits behind its natural and specialized factors weren't doled out towards the beginning of the venture. Putting these dents once a partnership can be done well on its own manner, indicates having advice which isn't obtainable towards the beginning of the opportunity. This generates the task easier, but in an ideal universe, these parts should be analyzed accurately on time all time as exhibited at “Fig. 7".



Complexity Table

System Component Type	Low	Average	High	Very High
Problem Domain Type(PDT)	3	6	10	15
Human Interaction Type(HIT)	4	7	12	19
Data Management Type(DMT)	5	8	13	20
Task Management Type(TMT)	4	5	9	13

CP1

	0-4 NEM	5-8 NEM	9-12 NEM	>=13 NEM
0-1 NSR	Low	Low	Average	High
2-3 NSR	Low	Average	High	High
4-5 NSR	Average	High	High	Very High
>5 NSR	High	High	Very High	Very High

CP2

0-2 NSR	0-5 NOA	6-9 NOA	10-14 NOA	>=15 NOA
0-4 NEM	Low	Low	Average	High
5-8 NEM	Low	Average	High	High
9-12 NEM	Average	High	High	Very High
>=13 NEM	High	High	Very High	Very High

3-4 NSR	0-4 NOA	5-8 NOA	9-13 NOA	>=14 NOA
0-3 NEM	Low	Low	Average	High
4-7 NEM	Low	Average	High	High
8-11 NEM	Average	High	High	Very High
>=12 NEM	High	High	Very High	Very High

>=5 NSR	0-3 NOA	4-7 NOA	8-12 NOA	>=13 NOA
0-2 NEM	Low	Low	Average	High
3-6 NEM	Low	Average	High	High
7-10 NEM	Average	High	High	Very High
>=11 NEM	High	High	Very High	Very High

Fig. 7. Complexity form for the cost estimation

V. CONCLUSIONS AND FUTURE WORK

The Extended Class Point Approach offers platform- based size estimation of Object-Oriented product and from pragmatic justification; it displays improved performance than the Class Point Approach. Software developed for class point calculation is simple to use. Calculating Adjusting Class Point value for many software using this tool compared the actual effort and the estimated effort using regression analysis. By concluding the effort estimation, we can achieve that, how much effort the venture has utilized and after that, we can have a more profound information of engineer group's proficient expertise level. Pioneers of the organization need this sort of information to deal with the organization and mastermind undertakings according to the developer‟s professional skill.

The jobs within this proposal shows that far more engagement with all the usage case targets plan is demanded to remember the ending target to prepare the overall helpfulness of this procedure. The procedure needs to be more linked with huge partnerships creating put in applications, constant

software, and frameworks rich in algorithmic intricacy. One among those issues of this procedure is shifting utilize case focuses on the team. The process represented by Schneider and vaccinations may create untrustworthy evaluations. More study Is Predicted to find reliable replies with this particular Problem.

REFERENCES

[1] B. Anda, H. Dreiem, D. I. Sjøberg, and M. Jørgensen, "Estimating software development effort based on use cases—experiences from industry," in ≪ UML≫ 2001—The Unified Modeling Language. Modeling Languages, Concepts, and Tools: Springer, 2001, pp. 487-502.

[2] M. Ruhe, R. Jeffery, and I. Wieczorek, "Cost estimation for web applications," in Software Engineering, 2003. Proceedings. 25th International Conference on, 2003, pp. 285-294: IEEE.

[3] G. Levesque, V. Bevo, and D. T. Cao, "Estimating software size with UML models," in Proceedings of the 2008 C 3 S 2 E conference, 2008, pp. 81-87: ACM.

[4] S. Grimstad, M. Jørgensen, and K. Moløkken-Østvold, "Software effort estimation terminology: The tower of Babel," Information and Software Technology, vol. 48, no. 4, pp. 302-310, 2006.

[5] D. C. Sharp, "Reducing avionics software cost through component-based product line development," in Digital Avionics Systems Conference, 1998. Proceedings., 17th DASC. The AIAA/IEEE/SAE, 1998, vol. 2, pp. G32/1-G32/8 vol. 2: IEEE.

[6] C. Ebert and C. Jones, "Embedded software: Facts, figures, and future," Computer, no. 4, pp. 42-52, 2009.

[7] R. S. Pressman, Software engineering: a practitioner's approach. Palgrave Macmillan, 2005.

[8] E. Mendes, N. Mosley, and S. Counsell, "The application of case-based reasoning to early web project cost estimation," in Computer Software and Applications Conference, 2002. COMPSAC 2002. Proceedings. 26th Annual International, 2002, pp. 393-398: IEEE.

[9] Y. Chen, B. W. Boehm, R. Madachy, and R. Valerdi, "An empirical study of eServices product UML sizing metrics," in Empirical Software Engineering, 2004. ISESE'04. Proceedings. 2004 International Symposium on, 2004, pp. 199-206: IEEE.

[10] L. M. Alves, S. Oliveira, P. Ribeiro, and R. J. Machado, "An empirical study on the estimation of size and complexity of software applications with function points analysis," in International Conference on Computational Science and Its Applications (ICCSA 2014), 2014, pp. 27-34: IEEE Computer Society Press.

[11] M. M. Kirmani and A. Wahid, "Revised Use Case Point (Re- UCP) Model for Software Effort Estimation," International Journal of Advanced Computer Science and Applications, vol. 6, pp. 65-71, 2015.

[12] M. M. Kirmani and A. Wahid, "Use Case Point Method of Software Effort Estimation: A Review," International Journal of Computer Applications, vol. 116, no. 15, 2015.

[13] A. Hjalmarsson, M. Korman, and R. Lagerström, "Software Migration Project Cost Estimation using COCOMO II and Enterprise Architecture Modeling," in PoEM (Short Papers), 2013, pp. 39-48: Citeseer.

[14] D. Singh, P. Mahajan, and V. Kumar, "Effective System analysis and design by using the Unified Modeling Language (UML)," International Journal of Research in Electronics and Computer Engineering, vol. 2, no. 4, pp. 2348-2281, 2014.

Comparison of NODEJS and PHP Technologies with Code Samples

[1] Mevlut Kagan Balga, [2] Fatih Basciftci
[1,2]Selcuk University Faculty of Technology Department of Computer Engineering Konya/TURKEY
E-mail: [1]kaganbalga@gmail.com, [2]basciftci@selcuk.edu.tr

ABSTRACT

The programming languages are constantly changing from the past to the present. A new programming language is available every day or a new version of existing programming languages is updated. The popularity of various programming languages is decreasing and they are no longer used. Programming languages can be categorized by many different classifications. The developer, the programming language, the platform, the compilation / interpretability, the open source status and many other criteria have a great influence on the choice of language. In this study, PHP which is still a popular and interpretable programming language in web technologies field and NodeJS technologies used in JavaScript programming language have been compared. In comparison, various code fragments and Apache Bench stress test tool were used. The results have been explained in details in this study.

Keywords - PHP, NodeJS, JavaScript, Apache Bench, Stress Test.

I. INTRODUCTION

Developments in programming languages used in computer technologies continue without stopping. It is possible to categorize programming languages with various classifications. Programming languages, which are close to the programmer's point of view, are called high-level languages, and languages that are close to the machine language are called lower- level languages. When selecting a programming language, the developer chooses this by considering both personal and programming language-specific features. The factors affecting the choice of programming language personally are specific purposes, personal differences, Integrated Development Environment (IDE) and license fees. Apart from this, the usability, intelligibility, performance, standards, compileable / interpretable difference and open source status of the programming language are taken into consideration [1].

In the compiled programming languages, high-level codes are converted into machine language with the help of the compiler. When the user runs the target program, the program converted into machine language executes. In interpretable programming languages, the interpreter interprets and reads the codes when the program is executed. Interpretable programming languages are more successful in error detection and flexibility. However, the interpreter is slower in performance because the program has to be activated each time it runs. In compiled programming languages, the compiler does the compilation once. The resulting program can be run repeatedly. In terms of performance, compiled programming languages are more successful than interpretable programming languages. Today, programming

languages are both compiled and interpretable. They use both processes at certain stages [1]. Scripting programming languages are another subcategory within programming languages. They work with the help of the interpreter, without compilation. Scripting languages are intended to connect components in various programming languages. Typically, developing programs in scripting languages is faster than traditional programming languages (C, C ++, Java, etc.), but is slower in terms of working performance. Scripting languages in a variety of different sources are defined as a language that provides coordination and interoperability between different languages. They are used in web pages and command lines [1].

With the development of the internet, Common Gateway Interface (CGI), Perl, PHP, JavaScript, etc. programming languages have emerged. CGI is an interface used to transmit requests between the user and the program running on the server. Perl is older than the programming languages used in the Internet and is used in conjunction with CGI [1, 2]. PHP, JavaScript, etc. programming languages are nowadays widely used in internet environments. TIOBE has a regular ranking of programming languages every month. This ranking is done according to the programming languages used by individuals and institutions and by the number of searches made in popular search engines. According to the TIOBE index for October 2018 programming languages, PHP is ranked 7th and JavaScript is 8th. The first 10 programming languages have been given in Table 1.

PYPL is another index that indexes programming languages according to Google Trends. As of October 2018, PHP is in 5th place and JavaScript is in 3rd place according to PYPL index. At the same time, according to the comparison between PHP and JavaScript, as of August 1, 2018, JavaScript has a usage rate of 8.4% and PHP has 7.6% worldwide [4]. Detailed list has been shared in Table 2.

Rank	Programming Language	Rating (%)
1	Java	17.801
2	C	15.376
3	C++	7.593
4	Python	7.156
5	Visual Basic .NET	5.884
6	C#	3.485
7	PHP	2.794
8	JavaScript	2.28
9	SQL	2.038
10	Swift	1.5

Table1: Programming language ranking by TIOBE index [3]

Rank	Programming Language	Share (%)
1	Python	24.72
2	Java	22.01
3	JavaScript	8.4
4	C#	7.71
5	PHP	7.42
6	C/C++	6.32
7	R	4.11
8	Objective-C	3.29
9	Swift	2.69
10	Matlab	2.08

Table2: Programming language ranking by PYPL index [4]

The Institute of Electrical and Electronics Engineers (IEEE) Spectrum is able to list the programming languages according to data from 9 different sources. These data are analyzed by 11 different metrics. According to data shared in IEEE Spectrum, as of 31 July 2018, PHP ranked 6th and JavaScript 8th. PHP ranked 4th and JavaScript 5th only when filtering according to the programming languages used in web platforms [5]. The overall order has been given in Table 3.

Rank	Programming Language	Spectrum Ranking
1	**Python**	**100**
2	**C++**	**99.7**
3	**Java**	**97.5**
4	**C**	**96.7**
5	**C#**	**89.4**
6	**PHP**	**84.9**
7	**R**	**82.9**
8	**JavaScript**	**82.6**
9	**Go**	**76.4**
10	**Assembly**	**74.1**

Table3: Programming language ranking by IEEE Spectrum [5]

On the field of comparing programming languages, it has been seen that various benchmark comparisons are made. Meyerovichet. al. (2013) analyzed how a programming language is used and how the rate of use is generated by comparing various programming languages [6]. Ogunrindeet. al. (2014) compared the PHP and ASP.NET programming languages by making benchmark and stress tests, according to the test results, PHP was more successful in both areas [7]. Lei et. al. (2014) has compared the PHP, NodeJS and Python programming languages using various tools, and NodeJS has been more successful [8]. Kruscheet. al. (2013) compared the various frameworks used in mobile platforms in the areas of performance, usability, support, cost and functionality; the most successful result was the Android SDK [9].

In general, PHP and JavaScript programming languages are popular according to the rank of different indexes. In this study, PHP and NodeJS technologies were compared by using various methods. NodeJS is a framework that runs on the JavaScript programming language. The comparison results and the methods used have been explained in the relevant sections.

II. MATERIAL AND METHODS

PHP and NodeJS technologies, benchmarking methods and benchmark tests have been explained in detail in this section.

PHP was first discovered by RasmusLerdorf in 1994 as a simple CGI set. It was not exactly a programming language during this period. With the second generation developed in 1996, PHP evolved into a complete programming language. In a study conducted by Netcraft in 1998, 60,000 domain names were used in PHP. Today, it is still being developed [10]. Static web pages can be prepared by using Hyper Text Markup Language (HTML). However, languages such as PHP can be used when it needs dynamism in a website. PHP works with HTML to add dynamism to the page. It also allows database operations to be performed by using database connections. The PHP interpreter is automatically activated when the PHP file is called. The PHP interpreter allows the dynamic output to be passed to the user by executing the operations in the PHP code [11].

In this study, PHP 7.1.3 version has been used. From the new versions of PHP, Zend Engine is used on the back side. PHP 7 uses Zend Engine 3.0 [12]. With Zend Engine, the PHP code is compiled to medium level, and the resulting output is interpreted by converting it to Opcode [13]. The schema for executing PHP code has been given in Fig. 1.

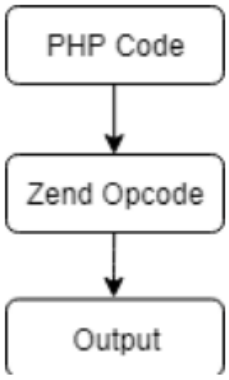


Fig.1. Execution of a PHP code

In this study, the PHP environment works on the Apache server. Apache 2.4.25 version has been used. Apache configuration settings given in Table 4 have been applied to the relevant modules so that the PHP environment can be tested at higher load.

Module	Setting	Parameter
mpm_prefork	MaxSpareServers	5000
	MaxRequestWorkers	5000
mpm_winnt	MaxConnectionsPerChild	5000
	ThreadsPerChild	5000
	MaxConnectionsPerChild	5000

Table 4: Apache configuration

NodeJS is a server-based framework used with JavaScript programming language. It was developed by Ryan Dahl. Google has a V8 engine built on it. Because it uses the V8 engine, it is developed according to ECMAScript standards. It was developed with C/C ++ programming languages for performance and low memory usage. It has a working structure depending on asynchronous I / O events. Because it is event-based, it can pass to the other process without waiting for an I / O operation to finish. Another reason for developing with the JavaScript programming language is that it supports event callback structures [14, 15]. In this study, 9.5.0 version of NodeJS has been used.

As a test environment, an Intel Core i7 7700HQ 3.8GHz processor and a 16 GB RAM memory capacity have been preferred. The technical details of the server and PHP, NodeJS environments have been given in Table 5.

Environment	Feature
Server	CPU: Intel i7 7700HQ 3.8 GHz RAM: 16 GB
	OS: Windows 10
	HDD: 1 TB – SSD: 128 GB
PHP	Apache: 2.4.25
	PHP: 7.1.3
	MySQL: 5.7.17
	PhpMyAdmin: 4.7.0
NodeJS	NodeJS: 9.5.0
	NodeJSPacket Manager (NPM): 5.6.0

Table5: Technical details of server and other environments

Five different test methods were used to compare two environments. The first of these tests is the Merge Sort algorithm. The Merge Sort algorithm is an algorithm that sorts by using the divide-and-conquer method and O(NlogN) has time complexity [16]. The second test was performed by executing the code giving the prime numbers up to a certain range in both environments. The third test was carried out by running a code that yielded a sequence of Fibonacci numbers up to a certain number. The fourth test was conducted by using the Collatz sequence. The Collatz sequence is that any number of x is reduced to 1. If the number is odd, the x value is multiplied by 3 and added by 1. If the number is even, the x value is

divided by 2. When these processes continue sequentially, 1 is reached [17]. The Collatz sequence has been shown in Eq. 1. The fifth and final test has been stress testing. Using ApacheBench, a "Hello World" application has been tested in both environments. ApacheBench is a benchmarking tool for measuring the performance of HTTP applications. It can perform a stress test by transferring a certain number of requests to the server at the same time or in succession [18].

$$T(x) = \begin{cases} \frac{x}{2} & (If\ x\ is\ even) \\ 3x + 1\ (If\ x\ is\ odd) \end{cases} \quad (1)$$

III. APPLICATION

For the comparison of PHP and NodeJS technologies, 4 different test methods were run both environments. The first test was performed at both environmetns of the Merge Sort algorithm. Merge Sort algorithm 0 – 10.000, 0 – 100.000, 0 – 1.000.000 and 0 – 2.000.000 length of arrays are ordered. The elements in the arrays are randomly selected numbers that are made at random intervals. It was repeated 10 times and the results were averaged. The average execution time of Merge Sort algorithm on NodeJS environment, has given a better execution timing than PHP environment. 0 - 2.000.000 length of an array has been made with the random numbers between 0 – 10.000.000 and Merge Sort algorithm has been performed, the NodeJS environment has been more faster execution time 152.27% than PHP. NodeJS has been more faster execution time 145.55% than PHP environment when both environments average execution time has been calculated. In the case of a two times more increase in array lengths, the execution time obtained in PHP was more than two times increase. The obtained working times have been made in Table 6.

		AverageExecution Time (Second)	
ArrayLength	**NumberRange**	**PHP**	**NodeJS**
0 – 10.000	0 – 10.000	0.030395	0.019577
0 – 100.000	0 – 100.000	0.351481	0.100496
0 – 1.000.000	0 – 1.000.000	4.211807	1.010394
0 – 2.000.000	0 – 2.000.000	8.737767	1.241534
0 – 10.000	FixedRange (0 – 10.000.000)	0.029707	0.009922
0 – 100.000		0.360691	0.061117
0 – 1.000.000		4.313795	0.600558
0 – 2.000.000		9.157918	1.240812

Table6: Execution times of Merge Sort algorithm in both environments

In both environments, a code to find prime numbers within the first 500.000, 1.000.000 and 2.000.000 numbers has been run. The same algorithm has been used in both environments in order to analyze the execution times in PHP and NodeJS environments more clearly. This algorithm has been developed with both environments syntax rules. The complexity and structure of the algorithms are exactly the same. NodeJS has been faster in this test compared to the execution times. In cases where the number range of

prime numbers increased by two times, the execution time obtained in PHP increased more than 2 times. When PHP and NodeJS environments were compared to the calculation of prime numbers in the range 0 - 2.000.000, NodeJS performed the same process 157.56% faster. Detailed working times have been shared in Table 7.

Prime Number Range	Average Execution Time (Second)	
	PHP	NodeJS
0 – 500.000	35.876737	4.00255
0 – 1.000.000	89.273908	11.486663
0 – 2.000.000	279.947178	33.221833

Table7: Execution time of Prime Number Finder program in both environments

As another test, the first 1000 Fibonacci numbers have been calculated iteratively in both environments. According to the results, PHP executed this program 187.88% faster than NodeJS environment. The calculation times have been given in Table 8.

Environment	Execution Time (Second)
PHP	0.000113
NodeJS	0.003618

Table8: Fibonacci series of the program working time in both environments

For the fourth test, the Collatz sequence was applied for each number in the range of 2 – 1.000.000, 2 – 500.000 and 2 – 100.000. This test was run a total of 10 times in the PHP and NodeJSenvironments and the results were averaged. According to the test results, PHP and NodeJS environments in the range of 2 – 100.000 gave close results in terms of execution time. In the tests conducted between 2 – 500.000 and 2 – 1.000.000, the difference in terms of execution time was large. The difference in the 2 – 500.000 number range was 3.6 seconds and the NodeJS environment has performed the same process by 145.30% faster than the PHP environment. The execution time difference between PHP and NodeJS environments in the range of 2 – 1.000.000 was 7.56 seconds, and the NodeJS environment executed the same process as 142.65% faster. When the load is low, the difference between the two environments is close, but when the load increases, the PHP environment performs the same operations more slowly. The execution times of the Collatz sequence have given in Table 9.

NumberRange	AverageExecution Time (Second)	
	PHP	NodeJS
2 – 100.000	0.073484	0.045876
2 – 500.000	4.288341	0.679218
2 – 1.000.000	9.086437	1.520734

Table 9: Collatz sequence program working time in both environments

The final test was stress testing. Two different scenarios were performed in this test by using ApacheBench. At the time of the concurrent 100 connections, 10.000 and 50.000 requests were transferred. According to the results, “Hello World” application in NodeJS accepted more requests per second. When we compared NodeJS and PHP environments with the scenario of 50.000 requests with concurrent 100 connections, NodeJS environment accepted 27.48% more request than PHP environment. The output of the comparison has been given in Fig. 2. The transfer times of the benchmark have been shared in Table 10.

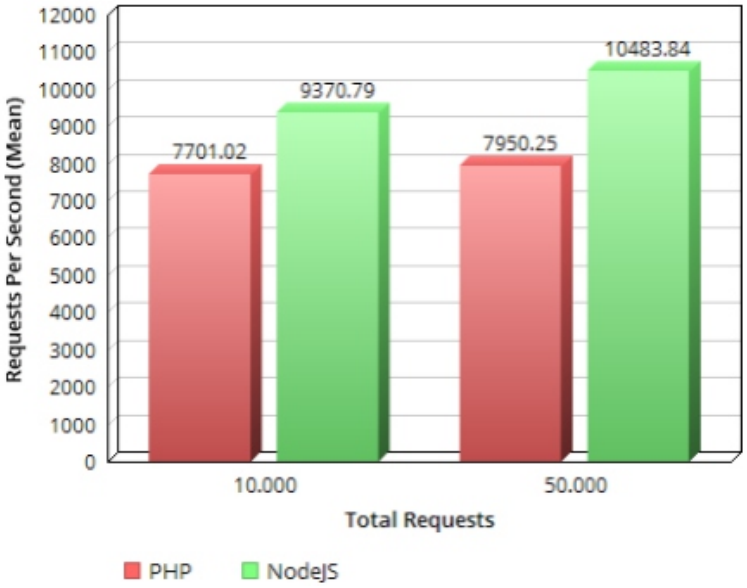


Fig.2. Total request comparison of PHP and NodeJS environments

Environment	Transferring Time
PHP	Total: 10.000 – Concurrent 100 Connections
	Total Mean Connection Time (ms): 13
	Requests Per Second (Mean):7701.02
	Time Per Request (ms – Mean): 12.985
	Transfer Rate: 1601.87Kbytes/sec
	Total: 50.000 – Concurrent 100 Connections
	Total Mean Connection Time (ms): 13
	Requests Per Second (Mean):7950.25
	Time Per Request (ms – Mean): 12.578
	Transfer Rate: 1653.71Kbytes/sec
NodeJS	Total: 10.000 – Concurrent 100 Connections
	Total Mean Connection Time (ms): 11
	Requests Per Second (Mean):9370.79
	Time Per Request (ms – Mean): 10.671
	Transfer Rate:1034.08Kbytes/sec
	Total: 50.000 – Concurrent 100 Connections
	Total Mean Connection Time (ms): 9
	Requests Per Second (Mean): 10483.84
	Time Per Request (ms – Mean): 9.538
	Transfer Rate:1156.91Kbytes/sec

Table10: ApacheBench comparison times (100 Connections)

The number of concurrent connections in the test performed with the ApacheBench tool was increased from 100 to 1000. In total, 10.000 and 50.000 requests were transmitted via 1000 connections. The results showed that the PHP server could not respond to requests, but the NodeJS could successfully respond to requests. PHP was able to respond 2597 of 10.000 requests on 1000 parallel connections and 2344 of 50.000 requests. NodeJS has more performance when the number of concurrent connections is high. The obtained statistics have been given in Fig. 3. Transfer times have been shared in Table 11.

Fig.3. NodeJS and PHP environments, concurrent comparison over 1000 connections

Environment	Transferring Time
PHP	Total: 10.000 – Concurrent 1000 Connections The target machine rejected the connection
	(Total of 2597 requests completed.)
	Total: 50.000 – Concurrent 1000 Connections
NodeJS	The target machine rejected the connection (Total of 2344 requests completed.)
	Total: 10.000 – Concurrent 1000 Connections
	Total Mean Connection Time (ms): 118
	Requests Per Second (Mean):8053.62 Time Per Request (ms – Mean): 124.168
	Transfer Rate:888.73Kbytes/sec
	Total: 50.000 – Concurrent 1000 Connections
	Total Mean Connection Time (ms):119
	Requests Per Second (Mean): 8297.54 Time Per Request (ms – Mean):120.518
	Transfer Rate:915.65Kbytes/sec

Table 11:ApacheBench comparison times (1000 connections)

According to the results of ApacheBench tests, PHP has performed well in cases where the number of concurrent connections is low. However, when there is an increase in the number of concurrent connections, PHP is unable to accept connections. NodeJS provided better performance than PHP in both scenarios.

IV. RESULTS

Nowadays, a new programming language has been born and some languages have been lost. Programming languages have been evolving day by day. With the release of new versions, new features have been added to the programming languages or some features have been removed. Programming languages can be categorized by a wide variety of classifications. When selecting a programming language for a project, the programmer is influenced by factors that depend on the platform and language. In this study, NodeJS used with JavaScript language and PHP Technologies were compared. 7.1.3 version of PHP and 9.5.0 version of NodeJS were used in the comparison. In the comparison, 5 different test methods were applied. These were the Merge Sort algorithm, the principal number program, the Fibonacci number sequence program, the Collatz hypothesis program and ApacheBench stress tests.

The Merge Sort algorithm was run in both environments, 0 – 10.000, 0 – 100.000, 0 – 1.000.000, 0 – 2.000.000 variables and 0 – 10.000.000 fixed number ranges 0 – 10.000, 0 – 100.000, 0 – 1.000.000 element arrays was sorted. In case of load, the execution speeds between PHP and NodeJS environments are large. When the average times were calculated, NodeJS performed the same process by 145.55% faster than PHP.

As the second test, prime numbers in the first 500.000, 1.000.000 and 2.000.000 numbers were calculated with the same algorithm in both environments. PHP completed this test in 135.03 seconds on average. NodeJS completed the same test in 16.23 seconds average. Based on average times, the NodeJS performed the prime number-finding test at a rate of 118.8 seconds. In addition, NodeJS has been observed to be faster when the number of calls is increased. NodeJS performed 157.56% faster than PHP according to the calculation of the prime numbers in the range 0 - 2.000.000.

As a third test, the first 1000 Fibonacci number sequence elements were found iteratively, the algorithm developed in PHP was more successful with a difference of 0.003505 seconds compared to NodeJS.

In the fourth test, the Collatz sequence was applied for each number between 2 – 1.000.000, 2 – 500.000 and 2 – 100.000. PHP performed this test on an average of 4.45 seconds and NodeJS in 0.74 seconds. The

execution time difference between PHP and NodeJS environments in the range of 2 – 1.000.000 was 7.56 seconds, and the NodeJS environment performed the same process 142.65% faster. When the load increases, there is a difference in execution speeds between PHP and NodeJS environments. NodeJS has performed operations in a much more performance manner.

In the last test using ApacheBench, two different scenarios were tested. "Hello World" was implemented in both technology environments. In the first scenario, a total of 10.000 and 50.000 requests were sent to the "Hello World" application in both PHP and NodeJS via concurrent 100 connections. The PHP programming language received an average of 7709.65 requests per second. On average, the NodeJS accepted 9927.31 requests per second. NodeJS received an average of 25.14% more requests compared to PHP in the specified number of requests and parallel connections.

In the second scenario with ApacheBench, a total of 10.000 and 50.000 requests were transmitted via 1000 connections. In this test the PHP server became unable to accept requests. The NodeJS server responded to all requests, with an average of 8175.58 requests per second. PHP was able to answer 2597 requests against 10.000 requests and concurrent 1000 connections; In case of 50.000 requests and concurrent 1000 connections, 2344 could respond to the request. Under this heavy load, the NodeJS was able to respond to all requests.

When PHP and NodeJS environments are compared, the execution speed difference between the two environments is very small if the load is low.

However, when the load increased, it was seen that the NodeJS environment gave more performance results than PHP. The success rate in the PHP environment can be increased by tuning the "Server- Pool Management" settings on the Apache web server.

REFERENCE

[1] Scott, M., L., 2009, "Programming Language Pragmatics Third Edition", Morgan Kaufmann Publishers, ISBN: 978-0- 12-374514-9.

[2] Robinson, D.,Coar, K., 2004, "The Common Gateway Interface (CGI) Version 1.1", RFC 3875.

[3] TIOBE, 2018, "TIOBE Index for October 2018", https://www.tiobe.com/tiobe-index/

[4] PYPL PopularitY of Programming Language, 2018, http://pypl.github.io/PYPL.html

[5] Cass, S., Bulusu, P., 2018, "Interactive: The Top Programming Languages 2018", https://spectrum.ieee.org/static/interactive-the-top- programming-languages-2018

[6] Meyerovich, L., A., Rabkin, A., S., 2013, "Empirical Analysis of Programming Language Adoption", ACM SIGPLAN International Conference on Object Oriented Programming Systems Languages & Applications, pp. 1 – 18.

[7] Ogunrinde, M., A., Olorisade, B., K., 2014, "Performance Comparison of dynamic Web scripting Language: A caseStudy of PHP and ASP.NET", International Journal of Scientific & Engineering Research, Vol. 5, Iss. 7,pp. 78 – 89.

[8] Lei, K.,Ma, Y., Tan, Z., 2014, "Performance Comparison and Evaluation of Web Development Technologies in PHP, Python and Node.js", IEEE 17th International Conference on Computational Science and Engineering, pp. 662 – 668.
[9] Sommer, A.,Krusche, S., 2013, "Evaluation of cross-platform frameworks for mobile applications", Software Engineering 2013, pp. 363 – 376.
[10] "History of PHP", http://php.net/manual/tr/history.php.php
[11] Bramer, M., 2015, "Web Programming with PHP and MySQL", Springer International Publishing, ISBN: 978-3- 319-22659-0.
[12] "PHP 7 makes powering the web a whole lot better", http://www.zend.com/en/resources/php-7
[13] Schlossnagle, G., 2004, "Advanced PHP Programming", Sams Publishing, ISBN: 978-0672325618.
[14] Cantelon, M.,Harter, M., Holowaychuk, T., J., Rajlich, N., 2014, "Node.js in Action", Manning Publications, ISBN: 9781617290572.
[15] Tilkov, S.,Vinoski, S., 2010, "Node.js: Using JavaScript to Build High-Performance Network Programs", IEEE Internet Computing, Vol. 14, Iss. 6, pp. 80 – 83.
[16] Qin, S., 2008, "Merge Sort Algorithm", Florida Institute of Technology.
[17] Arslan, A., N., 2018, "Methods for constructing Collatz numbers", Notes on Number Theory and Discrete Mathematics, Vol. 24, No. 2, pp. 47 – 54.
[18] "ab - Apache HTTP server benchmarking tool", https://httpd.apache.org/docs/2.4/en/programs/ab.html

Design and Implementation of Smart Traffic Light Controller using VHDL Language

Nour T. Gadawe [1*], Sahar L. Qaddoori [1]
[1] Electronic Engineering Department, Electronics Engineering College, Ninevah University, Iraq
*Corresponding author E-mail: nourgadawe@yahoo.com

ABSTRACT

The purpose of this paper is to design and implementation of smart traffic light controller system using VHDL language and FPGA. A structure of four road intersection has been selected. The intersection to be controlled is between a busy (main street), and somewhat less busy (side street), with sensor for the side street and walk request button. Also, the system contains switches to control the traffic light manually. The intersection uses four timing parameters with ability to change these parameters manually. The system has been success- fully tested with VHDL using Xilinx ISE 14.7i software environment and Chip-Scope, while, it is implemented in hardware using Xilinx Spartan 3E FPGA. It is easy to use and the cost for the same is also less as compared to the others. The designed traffic light control sys- tem is presented to work correctly as predictable.

Keywords: Traffic Light Controller (TLC); FSM; VHDL; Spartan 3E; FPGA; Xc3s500fg320-4.

1. INTRODUCTION

Traffic jamming is a critical predicament in many of the cities and towns all over the world. Traffic congestion has been causing many setbacks and challenges in the major and most occupied cities all over the globe. This traffic jam directly impacts the productivity of the workers, traders, suppliers and in all affecting the market and raising the prices of the commodities in a way light [1], [2], [3]. The problem of heavy jam is happened because of never configure the level of jam in each way and set the delay time. Another problem repre- sents when there is no jam, but the waiting still continues. The solution for these problems is to determine the level of jam and set the delay time. This problem need of evaluation of the traffic policeman, and then there is need for manual control of the traffic [4], [5], [6]. The target of this paper is to propose system provide solution for all above problems with least possible cost.

Traffic light controller (TLC) can be implemented using microcontroller, FPGA, and ASIC design. FPGA has many advantages over microcontroller, some of these advantages are; the speed, number of input/output ports and performance which are all very important in TLC design, at the same time ASIC design is more expensive than FPGA [7],[8]. Nowadays, FPGA becomes one of the most successful of today's technologies for developing the systems which require a real time operation. FPGA is a reconfigurable integrated circuit that consists of two dimensional arrays of logic blocks and flip-flops with an electrically programmable interconnection between logic blocks. The reconfiguration property enables fast prototyping and updates for hardware devices even after market launch [9], [10]. Most of in

the TLCs implemented on FPGA are simple ones that have been implemented as examples of Finite State Machine (FSM). [7]

The VHDL language has been selected for programming the FPGA to fill two important needs in the design process. Firstly, it gives full description of the structure of a design that is how it is decomposed into sub-designs, and how those sub-designs are interconnected. Sec- ondly, it allows simulating the design before starting the manufacturing. Accordingly, the designers can quickly compare alternatives and test for correctness without the delay and expense of hardware prototyping [11].

In this paper, a traffic light system is designed using VHDL and implemented by a single FPGA chip. The outline of the paper is struc- tured as follows: Section 2 covers the related works of traffic light system which are deliberated via VHDL and realized through FPGA. Also, traffic light controller system design is the subject of section 3. Furthermore, a simulation of the proposed traffic light system and the simulation results with a discussion are demonstrated in section 4. As well, section 5 includes the hardware implementation of traffic light system on Xilinx Spartan 3E FPGA and displays the testing and operation of it. Finally, the conclusions of this paper are provided in section 6.

2. RELATED WORKS

Many researches about traffic light system have been done in order to overcome some complicated traffic phenomenon but most of them sometimes fail to deal efficiently with the complex, time-varying traffic conditions and controller can't satisfy real-time character for traffic signal [2] [12]. The following literature survey elucidates some of researches in the recent few years.

In 2012, B. Dilip et al. [13] presented the FPGA implemented low cost advanced TLC system using Chip-Scope Pro and Virtual Input Output. The TLC implemented was one of the real and complex signaling lights in Kingdom of Bahrain, for pedestrian way included four roads and sensors and camera assisted motorway. The system had been implemented in hardware using Spartan-3E FPGA.

In same year, A. Mandloi et al. [14] had used Mealy based Finite state machine to design an efficient and intelligent traffic light control- ler. Mealy machines were used to implement the system because outputs signal were controlled by the inputs signals. The language used for the implementation was VHDL with mixed modeling style. The design was tested on Spartan-3 xc3s200 FPGA development kit. Total memory usage was 105456 kilobytes.

In 2014, Surabhi S. et al. [15] presented an adaptive traffic light controller (TLC) customized to have user defined number of intersection lanes and counts of signals for various intersections. The proposed system prototype was implemented on FPGA and typical TLCs were modeled using the finite state paradigm and rely heavily on software design flow. The hardware design had been deployed using the structural style of VHDL programming and thus offers more robustness.

In 2015, V. V. Dabahde el at. [16] proposed the Intelligent Traffic Light Control system to reduce waiting times of the vehicles at traffic signals. The proposed system made use of FPGA technology along with traffic sensors to control traffic according to the traffic require- ment and thus reduced the waiting time, at an intersection of two roads. The system had been successfully tested and implemented in hardware using ALTERA Cyclone II- FPGA. The system had many advantages over the exciting TLC.

In 2016, Ali K. A. [11] designed an intelligent traffic light control system using FPGA and VHDL. Moreover, multiple level of function- ality such as adding a standby control signal, the motion sensor handling part, special request implementation, and loaded traffic design had been taken into consideration. Its function was verified and simulated using ModelSim.

In 2017, P.Giri Prasad el at. [17] designed an intelligent transportation of system (ITS) using VHDL and implemented on FPGA. Also, it was determined the traffic in each road by using sensors. Using the traffic status, the signal time can be managed and in this way, the traffic on road can be handled. On each particular way junction, the IR sensors can be placed to detect the traffic density and give the current traffic status particular way in the junction. Its function was verified and simulated by using ModelSim.

3. TRAFFIC LIGHT CONTROLLER SYSTEM DESIGN

Figure 1 illustrates the structure of the selected traffic light model for four road intersections (one Main Street and three side streets). In general, Traffic Light Controller System consists of three lights (red, green and yellow) in each direction .The red light indicates to Stop, green light indicates to allow the traffic and yellow light indicates the caution that the traffic is going to be stopped in few seconds. While, turning in yellow and red lights at the same time indicates the caution that the traffic is going to be moving in few seconds. The intersection is fitted with a sensor for side street traffic and with walk request button.

This traffic light controller also has provision for walk light (which consists of two lights red and green, where, green light allows the walkers to pass the street while red light avoids the walkers from passing the street) and for the traffic sensors in each one of the side streets. A simple block diagram of the traffic

light controller system is exposed in Figure 2. The design is composed of finite state ma- chine (FSM), data storage (D_RAM), timer, divider, and various synchronizers (latch, and synchronizer).

Finite state machines (FSM)

Finite State Machines (FSM) is the heart of the traffic light controller system. This FSM controls the loading of static data storage loca- tions with timing parameters, displaying these parameters by reading RAM locations, and the control of the actual traffic lights. There are four timing parameters in this system as displayed in Table 1. They are the base interval (TBASE) for side green, an extended interval for main green and walk green light (TEXT), the time for yellow light (TYEL), and a blink interval (TBLINK). The user can specify the four timing parameters using two switches (L0, L1) manually. The FSM can execute four functions specified by two functions switches (F0, F1). These functions are listed in Table 2, where, the user can execute one of four possible functions: writing new timing parameters, reading old timing parameters, running traffic light in normal mode, and running the traffic light in blinking mode as obtained in Figure 3a. Besides, the idle state of the FSM is called the reset state, in this state, the lights are turned off and the system does not do anything. The system will stay in the reset state until the GO button is pressed.

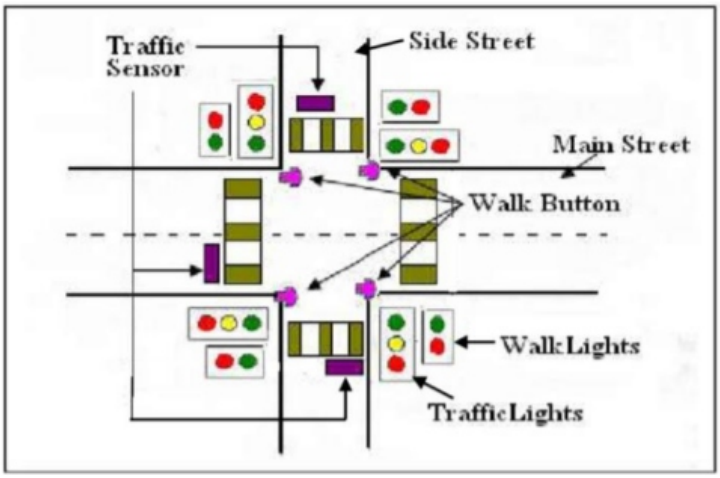


Fig. 1: Example for the Traffic Light System.

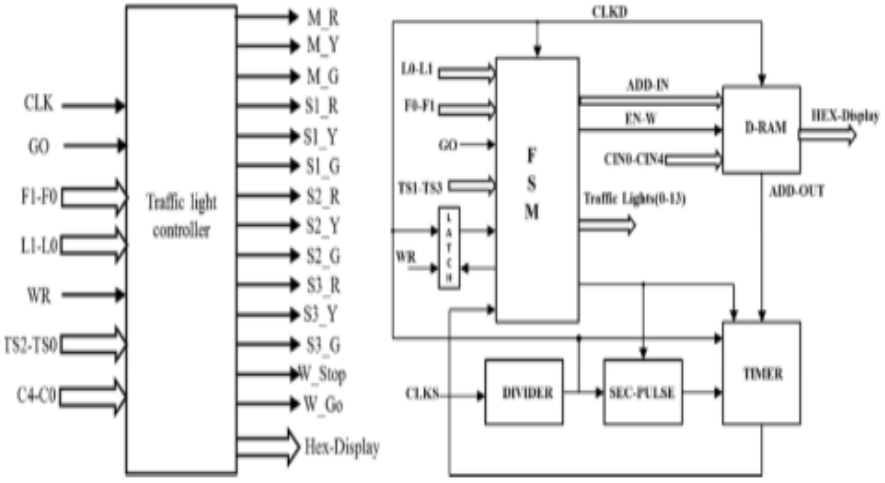


Fig. 2: The Structure Model of the Traffic Light Controller System.

Using the writing function, the user can specify the any one of the four timing parameters as shown in Table 1 using (L1, L0) switches, the value of the parameter is set using the (C4_C0) switches. For the reading operation, the user can use the same L1 and L0 switches to denote which of the four timing parameters to view on a set Hex-LEDs. In normal mode or blinking mode, the system just cycles through the various traffic light states. The regular controller has been designed with nine states as presented in Table 3 without taking the traffic sensors and walk request in the point view.

In the normal mode that is displayed in Figure 3b, the side street has a shorter green interval than the main street, but if there is traffic on the side street when the controller is about to cycle to turn that green light off, it will extend the green light by the shorter (side street) green interval. Thus the green light on the side street will stay on until traffic on the side street clears. Traffic sensor switch is used to simulate the effect waiting traffic on the side street, the system complies by keeping side street green until the traffic sensor is switch off. The walk light comes on after the main street yellow interval, and then only if the walk request button has been pushed. Late at night or when something in the system is not working, the light goes into the blinking mode this involves the lights blinking on and off, alternat- ing between main yellow side red, and main red side yellow as shown in Figure 3a.

Table 1: Time Operations for the Traffic Light Controller System

F1	F0	Mode Type
0	0	Read Mode
0	1	Write Mode
1	0	Normal Mode
1	1	Blinking Mode

Table 2: Modes Operations for the Traffic Light Controller System

L1	L0	Time Type
0	0	TBASE
0	1	TEXT
1	0	TYEL
1	1	TBLINK

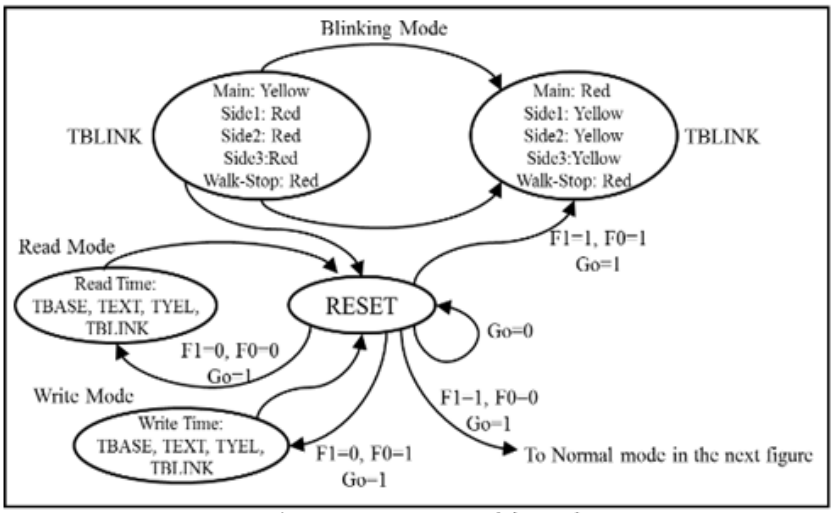


Fig. 3: A) FSM State Transition Diagram.

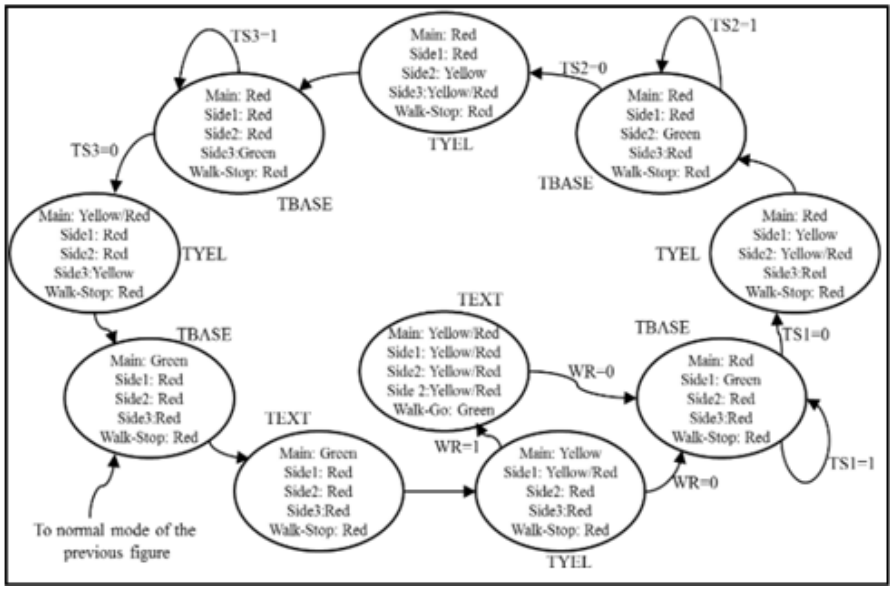


Fig. 3: B) FSM State Transition Diagram.

Table 3: Operations of Traffic Light Controller System in Normal Mode without Using the Traffic Sensors and Walk Request

Time Type	Main Street	Side 1 Street	Side 2 Street	Side 3 Street
TBASE	Green	Red	Red	Red
TEXT	Green	Red	Red	Red
TYEL	Yellow	Yellow/Red	Red	Red
TBASE	Red	Green	Red	Red
TYEL	Red	Yellow	Yellow/Red	Red
TBASE	Red	Red	Green	Red
TYEL	Red	Red	Yellow	Yellow/Red
TBASE	Red	Red	Red	Green
TYEL	Yellow/Red	Red	Red	Yellow

D_RAM

This component is used to store the four timing parameters which are declared in Table 1. Depending on the signal en_w, which select to read the contents by L0-L1 switches, or write new timing parameters by C0-C4 switches and display the contents on the HEX_LEDs

Divider, sec_pulse, and timer

The divider component is used to generate the clock (1 MHz) for overall system from 50 MHz of the chip FPGA Spartan 3E. While, the Secpulse component is used to generate one second clock, which is used in the timing of the traffic light. The timer is implemented as counter.

Latch and sensors

Walk signal is latched so that when the user pushes the walk button once the signal is queued until the FSM need it. Figure 4 expresses schematic circuit of walk latch. Furthermore, there are three traffic sensors which are synchronized by simply passing it through a flip flop.

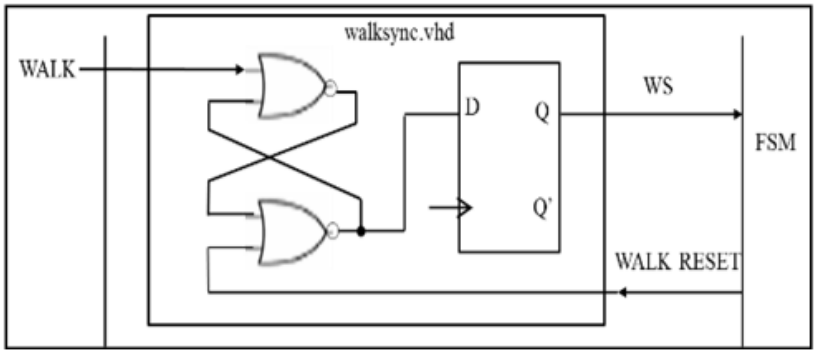
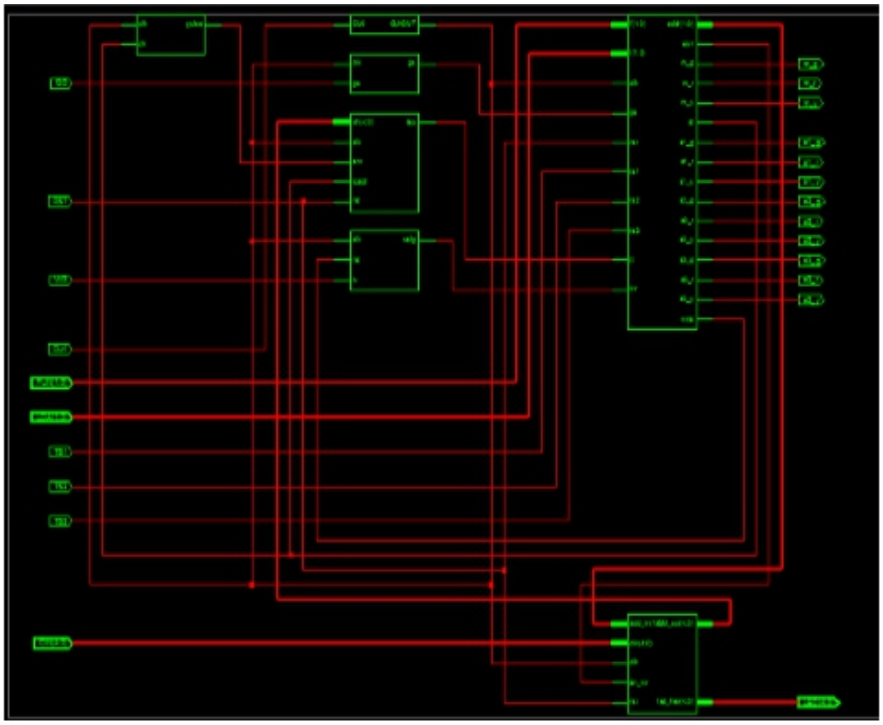


Fig. 4: Walk Latch Schematic Circuit.

The key advantage in using the VHDL in systems design is allowing the behavior of the required system to be described (modeled) and verified (simulated) before synthesis tools translate the design into real hardware (gates and wires)[11]. Figure 5 indicates the RTL and technology schematic diagram of the traffic light controller system. All component of the system are simulated using Xilinx ISE 14.7i. Viewing a schematic allows to see a technology level representation of HDL optimized for specific device architecture, that it may be assisted to discover the scheme issues early in design process. The simulation result of the traffic light controller system in reading and writing modes are exhibited in Figure 6a. In this case, all traffic light outputs are off and HEX_LEDs display the output of memory loca- tion which represents the selected time mode. Furthermore, the normal mode is displayed in Figure 6b, which denotes the operation of TLC system as appeared in Figure 3b. The synthesis process generates net list for each design element. Synthesis process checks code syntax and analyzes the hierarchy of the design to ensure that the design is optimized for the system.



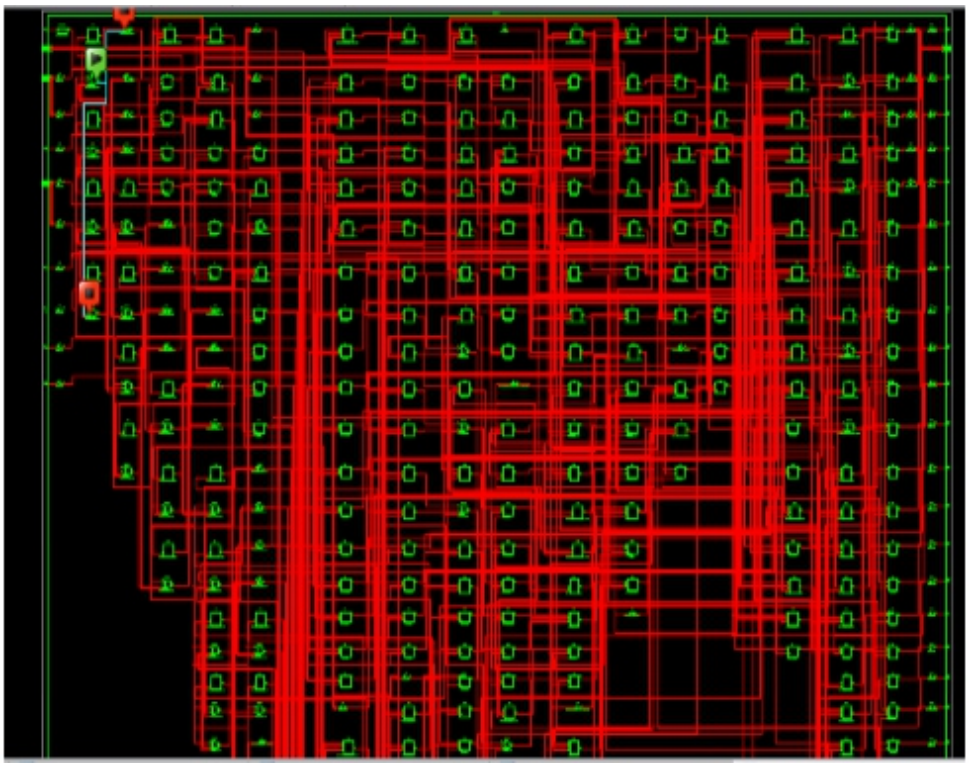


Fig. 5: RTL &Technology Schematic of the Traffic Light Controller System.

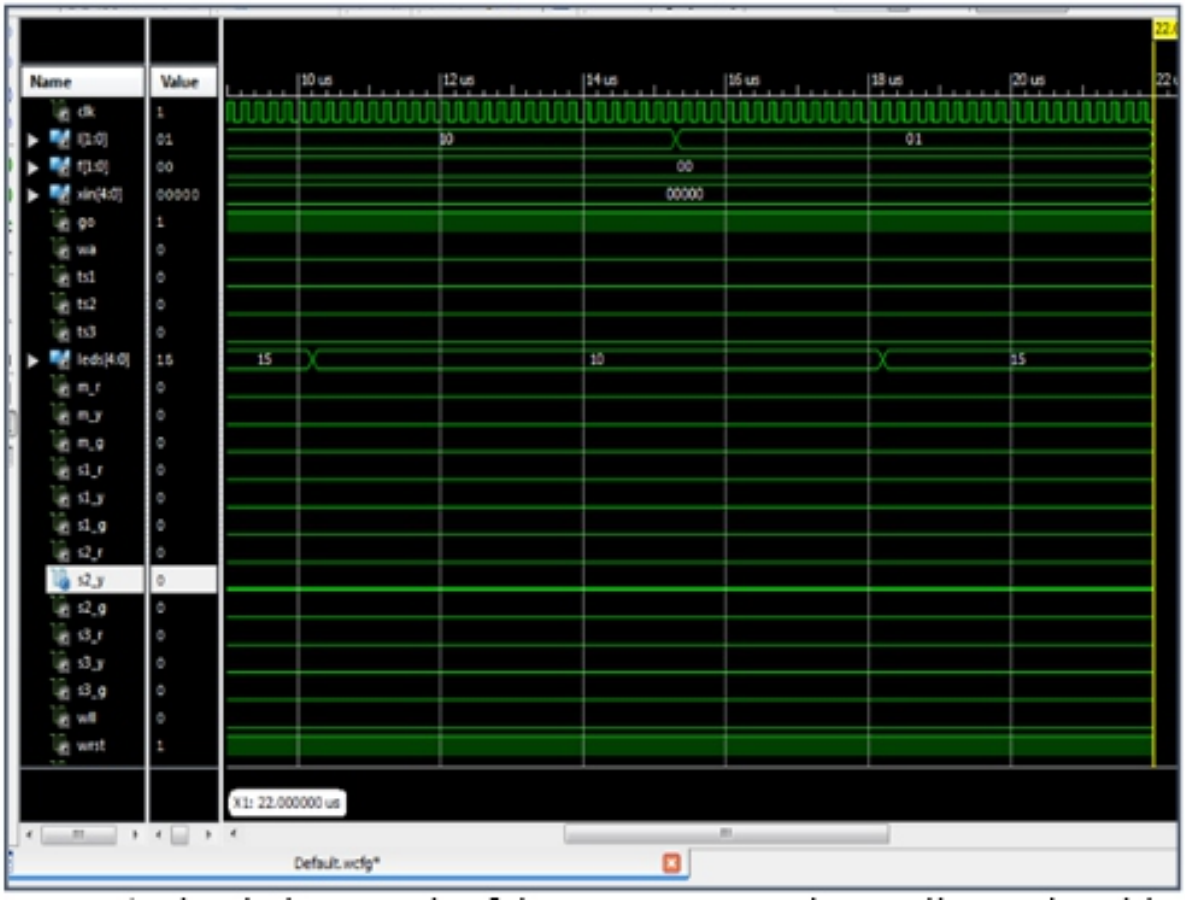


Fig. 6: A) Simulation Result of the T.L.C System in Reading and Writing.

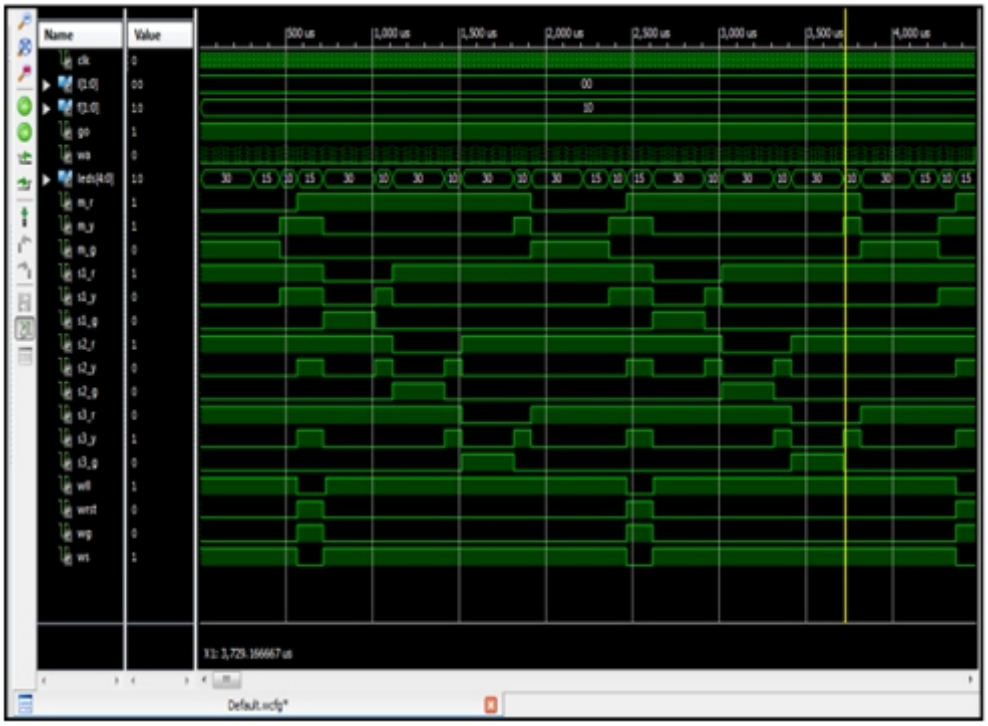


Fig. 6: B) Simulation Result of the T.L.C System in Normal Mode.

5. HARDWARE IMPLEMENTATION

The traffic light controller system design is implemented by synthesizing the VHDL structural code design, then generating bit file using Xilinx ISE 14.7 tools. This bit file is downloaded to the FPGA Spartan 3E development kit xc3s500efg320. The system's outputs are more than the LED on FPGA, then, it is used the LEDs to display one state or use the supporting chip (expansion) external pin digilent (FX2 MIB) as confirmed in Figure 7. Figures 8 and 9 demonstrate the real time implementation of TLC and Chip-Scope implementation respectively. The system design gives the realization of the hardware system as well as the software. The hardware consumptions are listed in Table 4.

Fig. 7: Supporting Chip (FX2 MIB).

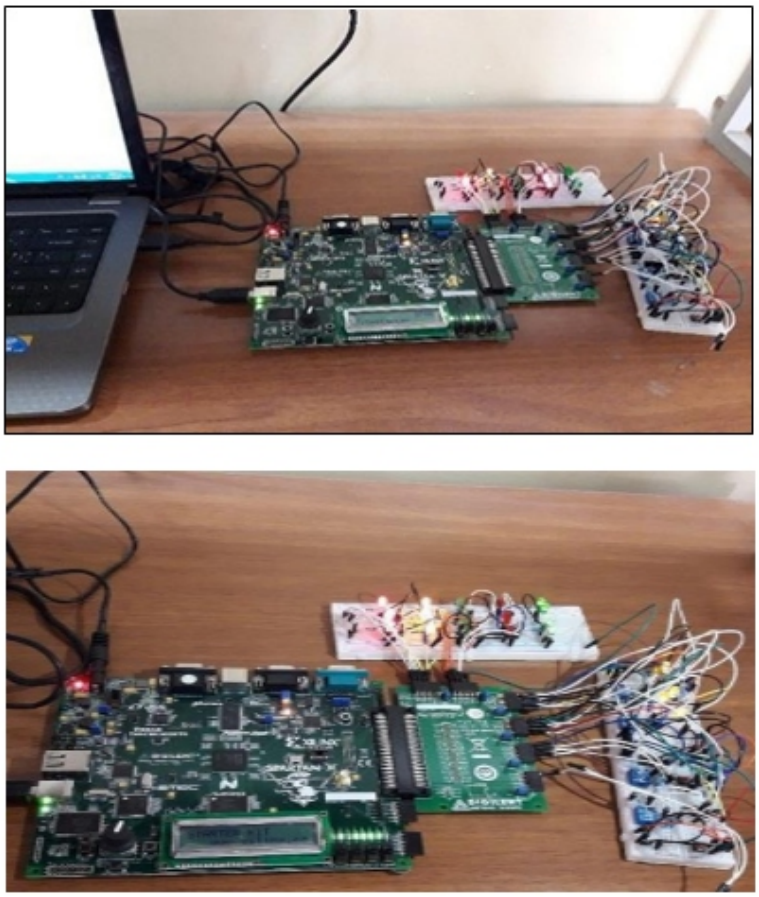


Fig. 8: FPGA Implementation of TLC System.

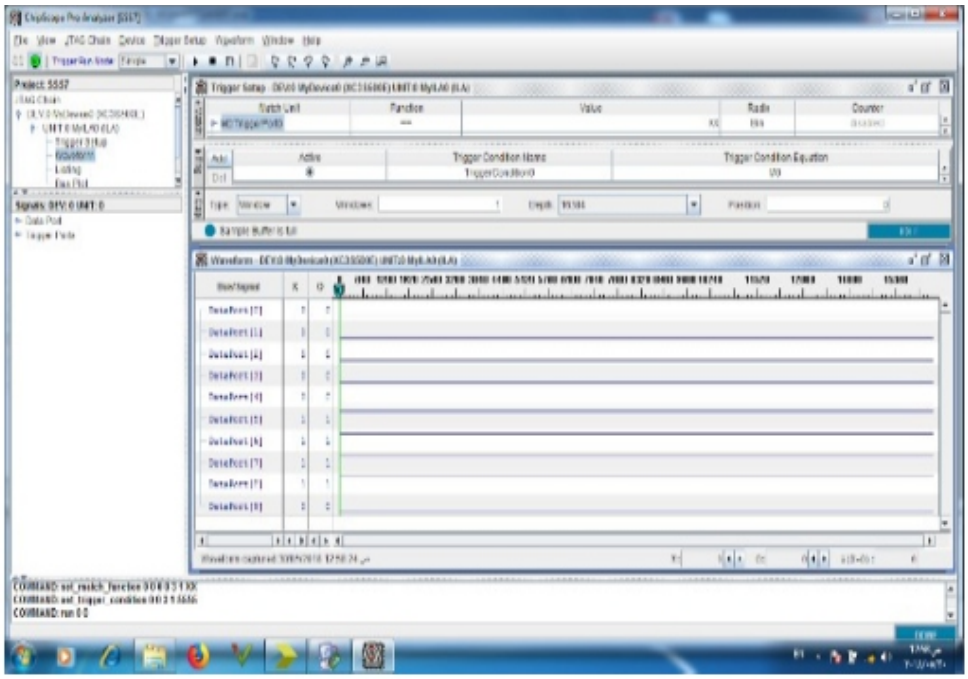


Fig. 9: TLC System Output Using Chip-Scope.

A smart Traffic Light Controller system is designed using FPGA for four roads intersection with traffic sensors and walk request signals. The system has been simulated using VHDL to realize alternating

traffic light and FSM for efficient T.L.C. with ability to change its timing parameters manually. Each subcomponent is constructed and tested thoroughly before moving onto the next one. The design is robust; all the design decisions were inspected comprehensively before employment. Synchronization component are very important in the system design where they are implemented without any hazards in the system. Overall the design and implementation of the traffic light controller is respectable to design more complex system. The system is verified on FPGA Spartan 3E xc3s500efg320-4.

Table 4: Device Utilization Summary

Selected Device : 3s500efg320-4	
Number of Slices:	83 out of 4656 1%
Number of Slice Flip Flops	115 out of 9312 1%
Number of 4 input LUTs:	141 out of 9312 1%
Number of bonded IOBs	40 out of 232 17%
Number of GCLKs	2 out of 24 8%
Timing Summary:	
Minimum period:	6.132ns
Minimum input arrival time before clock	5.355ns
Maximum output required time after clock	4.532ns

REFERENCES

[1] A. Raza, A. Kumar, E. Chaudhary, "Traffic Light Controller Using VHDL", International Journal of Modern Trends in Engineering and Research (IJMTER), Vol. 04, Issue 4, PP. 57-62, April 2017. https://doi.org/10.21884/IJMTER.2017.4118.C529F.

[2] T. Royani, J. Haddadnia, M. Alipoor, "Control of Traffic Light in Isolated Intersections Using Fuzzy Neural Network and Genetic Algorithm", In- ternational Journal of Computer and Electrical Engineering, Vol. 5, No. 1, PP. 142-146, Feb. 2013. https://doi.org/10.7763/IJCEE.2013.V5.682.

[3] M. Dev, Prerna, Sh. Sachin, S. Sharma, U. Tyagi, "Smart Traffic Control System Using PLC and SCADA", International Journal of Innovative Re- search in Science, Engineering and Technology, Vol. 1, Issue 2, PP. 169-172, Dec. 2012, Website: www.ijirset.com.

[4] M. E. Safi1, "Smart Traffic Light Controller Based on Microcontroller", Iraqi Journal of Computers, Communications, Control, and Systems Engi- neering (IJCCCE), Vol.16, No.1, PP. 38-45, 2016.

[5] J. Alam, M. K. Pandey, "Advance Traffic Light System Based on Congestion Estimation Using Fuzzy Logic", International Journal of Emerging Technology and Advanced Engineering, Vol. 4, Issue 3, PP. 870-877, Mar. 2014, Website: www.ijetae.com. https://doi.org/10.9790/0661- 16363644.

[6] A. Choudhary, "Adaptive Control of Traffic Grid Using Fuzzy Logic", International Journal of Electrical, Electronics, and Data Communication, Vol. 2, Issue 8, PP. 51-57, Aug. 2014.

[7] L. Shanigarapu, K. V. Reddy, "Design and Implementation of Intelligent Traffic Light System", International Journal of Computer Science and Mobile Computing (IJCSMC), Vol. 4, Issue 7, PP. 93 – 102, July 2015.

[8] D. Bhavana1, D. R. Tej, P. Jain, G. Mounika, R. Mohini, "Traffic Light Controller Using FPGA", Internati onal Journal of Engineering Research and Applications, Vol. 5, Issue 4, (Part-6), PP.165-168, April 2015, Website: www.ijera.com.

[9] D. Nath, V.V.Shete, Abhay Tambe, "Customization of Arm Processor in Mixed Signal FPGA", International Journal of Engineering Research & Technology (IJERT), Vol. 1, Issue 6, Aug. 2012.

[10] Beyrouthy, L. Fesquet, "An Asynchronous FPGA Block with Its Tech-Mapping Algorithm Dedicated to Security Applications", International Journal of Reconfigurable Computing, Vol. 2013, 2013. https://doi.org/10.1155/2013/517947.

[11] A. K. Abdulrazzaq, "Design an Intelligent Traffic Light Controller", Thi-Qar University Journal for Engineering Sciences, Vol.7, No. 1, pp. 47-57, 2016.

[12] S. Nath, Ch. Pal, S. Sau, S. Mukherjee, A. Roy, A. Guchhait, D. Kandar, "Design of an FPGA Based Intelligence Traffic Light Controller with VHDL", International Conference on Radar, Communication and Computing (ICRCC), PP. 92-97, 21 - 22 Dec. 2012. https://doi.org/10.1109/ICRCC.2012.6450554.

[13] B. Dilip, Y. Alekhya, P. Divya Bharathi, "FPGA Implementation of an Advanced Traffic Light Controller using Verilog HDL", International Jour- nal of Advanced Research in Computer Engineering & Technology (IJARCET), Vol. 1, Issue 7, PP. 1-6, Sep. 2012.

[14] A. Mandloi, Rajnikant, "High Level Modeling of Real Time Traffic Light Controller", International Journal of Engineering Science and Technolo- gy (IJEST), Vol. 4, No.12, PP. 4773-4779, Dec. 2012.

[15] S. S. GAOPANDE, A. D. BHARADE, "Structural Modeling of Intelligent Traffic Light Controller on FPGA", International Journal of Electrical, Electronics and Data Communication, Vol. 2, Issue 9, PP. 63-66, Sep. 2014.

[16] V. V. Dabahde, R. V. Kshirsagar, "FPGA-Based Intelligent Traffic Light Controller System Design", IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2, Issue 4, PP. 1268-1271, April 2015, Website: www.ijiset.com

[17] P.G. Prasad, S.V. Krishna, "Design and Implementation of ITLC System Using FPGA", Journal of Electronics and Communication Engineering (IOSR-JECE), Vol. 12, Issue 5, Ver. II, PP. 07-13, Sep.- Oct. 2017, Website: www.iosrjournals.org.

Development of A Real Time E- Supervisory Archival System

[1] Wahab Wajeed Bolanle, [2] Komolafe Raphael Oluwaseyi, [3] Wahab Waliyu Bolaji
[1]Bells University of Technology, OTA OGUN State, Nigeria
[2]D. S Adegbenro ICT Polytechnic, ITORI, OGUN State
[3]University of Hertfordshire, Hatfield, Uk
E-mail: [1]bola4reel2000@yahoo.com, [2]komolafe.raphael@dsadegbenro.edu.ng,
[3]bolaji_wahab@yahoo.com

ABSTRACT

In many tertiary institutions in the country, students seek a project in a given field of specialty as part of the upper level of their degree programme. Usually, a project can be filled by at most one student, though in some cases, a project is suitable for more than one student to work on simultaneously. It is a common phenomenon that final year student''s projects are often managed in paper-pen system where most of these students lay their hands on already completed projects, and present the same to their supervisors without the knowledge of the same. This had caused so many duplications of projects words for words, year in, year out. An e-supervisory archival system was developed to monitor projects that had been implemented before to enable supervisors detect repetitive projects and guide the student right to conduct original and unique projects. The system was developed using HTML, JavaScript, Cascading Style Sheet (CSS) and PHP as the front-end while MySQL Structured Query Language was used as the back end. The system developed was able to eradicate repetition of projects carried out by students.

Keywords - Archival System, E- Supervisory, Java Script, Cascading Style.

I. INTRODUCTION

Academic integrity is core to the mission of the higher education sector in any nation. A final year project is a task often engaged in by students as a means of applying the acquired knowledge to achieve set objectives. The process of managing final year projects involves three parties: students, supervisors and the Head of department. The Head of department (sometimes represented by Project coordinator) assigns a lecturer to supervise each student. Over time, it has been discovered that students are becoming very lazy and are no more original in the project carried out to qualify them for the degree awarded. From the findings carried out, it was discovered that since students are allocated to different supervisors, a project carried out by a student in a particular year with certain supervisor could be picked up by another student in another year and replicated to another supervisor within the same department without the supervisor‟s knowledge. The reason for these is often characterized with a manual process which involve a paper-pen method of keeping past projects. Keeping track of such paper-pen projects by supervisors could be tasking, repetitive and tiring, there is the need to design and develop a user-friendly, easily accessible and robust system for users (basically supervisors) to ameliorate this problem. Research misconduct, which has been widely defined as “fabrication, falsification, or plagiarism in

proposing, performing, or reviewing research, or in reporting research results" (Office of Science and Technology Policy, 2000), fundamentally undermines this mission. Of these three forms of research misbehavior, fabrication (making up data) and falsification (changing data) are often seen as most problematic, as they directly distort scientific knowledge and the decisions based on it [5]. Both have received considerable media attention in recent years [2].

II. STATEMENT OF PROBLEM

In Nigeria, every year, about 130,000 students become graduates [4] which means about the same amount of reports are submitted by the students before graduation. This leads to several problems with cases of plagiarism going unnoticed as the students attend different schools, which makes it difficult to keep track of what document belongs to who. Even for students of same school, there are no database to store their projects.

III. AIM AND OBJECTIVES

The aim of this research is to develop an E- Supervisory Archival System Using an Agile Model Software Approach. The objectives are;

- Design a model for E- Supervisory Archival System
- Design a project database for save and retrieve file system.
- Simulate and create the database

IV. LITERATURE REVIEW

A student project is a work that a student creates as part of a tertiary/high school, undergraduate, or graduate school program. Student project are long academic documents that students write after they research a particular subject in depth. Therefore, student project is usually assigned once per course, persemester, or only once as part of the requirement for an academic program. A graduate project is a paper that a student works on for a large portion of the graduate program, especially in his or her final months of the program. This project requires a huge amount of research and may even be ground-breaking for a particular industry [7]. A related study by [2] developed a final year project management system for Information Technology programmes and tried to implement an online platform that facilitates the final year projects process. The system among others was to help project supervisors track the progress of the projects in form of group projects using project management tools. According to [6], In the Design and Development of a University Portal for the Management of Final Year Undergraduate Projects" designed a portal-based system used for the automation of the processes associated with the management of final year projects in the department of Electrical and Information Engineering, Covenant University, Nigeria. The process started with the allocation of project supervisors to students

down to the final clearance of the student after the project defense. ASP.NET was used to create the web server, C-sharp language (C#), Microsoft SQL server 2005 as the back-end [6]. According to [2], [1] there is the need to inculcate research skills into students by introducing research elements in the school teaching curriculum at all levels. The study developed a prototype web-based supervision management system. The prototype of their work consists of three modules, namely user profile, project monitoring (of software development and report writing) and appointment setting. statistical analysis on „Student project performance management system for effective final year and dissertation project supervision. The study was tailored towards integrated and collaborative online supervision system for final year and dissertation projects.

A. Types of Projects

1. Projects proposed by students; and
2. Projects proposed by academic staff or other supervisors.

For projects proposed/ Identified by students, the latter are required to discuss the proposal with potential supervisors from within or outside the department. The project title has to be approved by the supervisor(s) and the department.

- **Projects proposed by students:** Students are required to submit a proposal of the project which should include: A project Title, Problem Statement, Literature review and research methodology. The proposal should be signed by the Supervisor(s) and submitted to the project Coordinator.

- **Projects proposed by academic staff or other supervisors:** The coordinator requests all academic staffs concerned to submit a list of project titles available for the programme.

B. Plagiarism

Plagiarism as a term originates from "plagiarism" Latin word, which means "to hijack/ steal a person." Literally, plagiarism means "taking or borrowing someone else's efforts and works and presenting them as the efforts of someone else. Turnitin.com, and Plagiarism.org defined plagiarism as: – Turning the work of someone else‟s as your own. Copying ideas, or contribution, or documents from someone else without giving some credit for the original owner.

C. Types of Plagiarism

There are a number of ways in which plagiarism can be classified. According to plagiarism.org, it can be classified according to the intent of security level depicted in table 1.

No	Plagiarism Type	Descriptions
1	Clone	Steal the whole work(with every small details) of another person's and use it as its own work.
2	Copy and Paste	Use large part of the text with alternation or citation
3	Find and Replace	Keep the original content while changing the main keywords
4	Remix	Use multiple documents and paraphrase them into one document
5	Recycle	Also referred to as self-plagiarism, it occurs when the author use his old work to form the new work without using citation
6	Hybrid	This type occurs when the there is an improper citation
7	Mash up	Combining the text from different resources
8	404 Error	Using original contribution with valid citation
9	Aggregator	No original contribution with valid citation
10	Re-Tweet	Using the existing structure with valid citation

Table1: Types and Description of Plagiarism[8]

A number of plagiarism detection tools have been produced in order to check the text similarity in different academic institutions in various advanced countries like the US, the UK, Australia, as well as in the Arabian countries like the Kingdom of Saudi Arabia, Jordan, and Egypt [8]. We focused on the following Plagiarism Detector System (PDS) tools in our research. Table 2 summarizes some of the well-known plagiarism detection tools.

TurnitIn	It is checks the plagiarized text by using a digital fingerprinting in order to match submitted documents against both of in-house databases, which contains previously attached papers, and internet resources.
SafeAssign	SafeAssign works by comparing submitted assignments against a set of identified sources to find areas of overlap between the existing works and submittited assignments
Plagiarism Detection.org	PlagiarismDetection.org provides a user-friendly and innovative plagiarism prevention and detection online tool that can effectively help instructors and students. It is designed to discover all chances for plagiarized text and runs against all internet resources

Table 2: Examples of Plagiarism tools [3]

V. METHODOLOGY

Approach for the Development of HMS

The Software Development Life Cycle (SDLC), Agile model has was adopted for this research. As an innovative approach, the agile software development method which is usually used for articulating a well-organized project management procedure allowing for recurrent alterations. Certainly, such type of method is a theoretical outline for undertaking several software engineering projects.

A. System Design

Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources. hardware requirements list is often accompanied by a Hardware Compatibility List (HCL), especially in case of operating system.

B. Software Requirements

Software Requirements deals with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. This System is a web-based application that runs on XAMP Server. The system utilizes MySQL as its database and backend. PHP was used alongside HTML and CSS for the development of the user interface and functionalities of the system needs[12].

C. Development Tools

S/N	Programming Languages/ Tools	Uses
1	PHP	Server side programming Language used for sending and retrieving records from the database
2	MySQL	Database Query
3	Visual Studio Code	The IDE used to write and run the codes
4	Html, Css	For implementing the structure and design of the software
5	XAMP Server	The server hosting the web Application

Table 3: Development Tools and Programming Languages Used in the Development

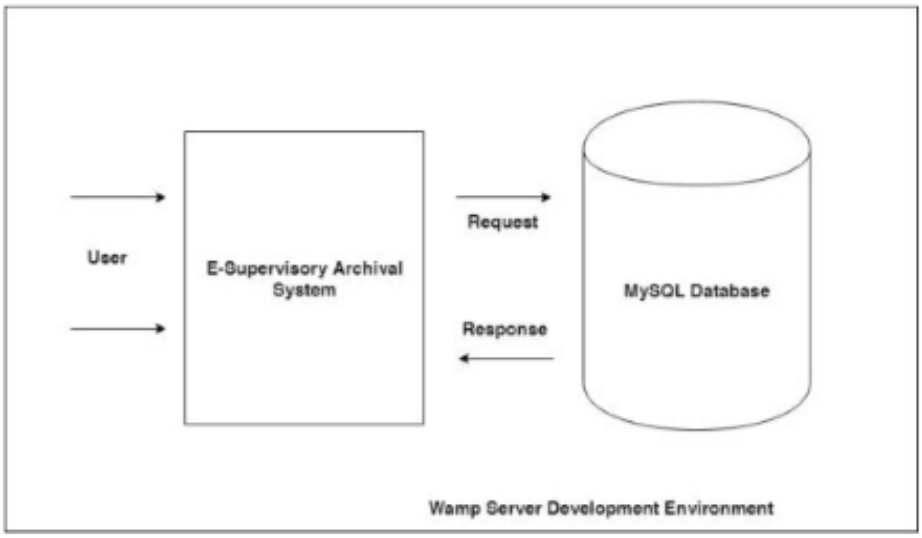


Figure 1: Description of the relationship between the User, System, Database and Sever

D. Data Flow Diagrams

Data flow diagram is a graphical representation of the flow of data through an information system, modelling its process aspects. A Data Flow Diagram (DFD) is often used as a preliminary step to create an overview of the system without going into great detail. As shown on the Data Flow Diagram of the design in figure 1

VI. SYSTEM IMPLEMENTATION AND DESIGN

1. HTML

Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as CSS and scripting languages such as JavaScript.

HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. It can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of the web pages [11].

2. Cascading Style Sheet (CSS)

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language like HTML. It enables the separation of presentation and content including layout, colors and fonts. The separation can improve the content accessibility, provide more flexibility and control in the specification of presentation characteristics. The goal of the CSS is to provide ease of

development, ease of collaboration during development and performance of the deployed style sheets in the browser [9].

3. Hypertext Preprocessor (PHP)

Hypertext Preprocessor is a general-purpose programming language that is especially suited for server-side web development, in which case PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on websites or elsewhere. PHP code may be executed with a command line interface (CLI), embedded into HTML code or used in combination with various web content management systems, web frameworks and web template systems. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common Gateway Interface (CGI) executable [9]. The web server outputs the results of the interpreted and executed PHP code, which may be any type of data, such as generated HTML code or binary image data. PHP can be used for many programming tasks outside of the web context.

4. MySQL

It is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing [9]. MySQL is an important component of an open source enterprise stack called LAMP. LAMP (Linux, Apache, MySQL and PHP) is a web development platform that uses Linux as the operating system, Apache as the web server, MySQL as the relational database management system and PHP as the object-oriented scripting language. (Sometimes Perl or Python is used instead of PHP.). Originally conceived by the Swedish company MySQL AB, MySQL was acquired by Sun Microsystems in 2008 and then by Oracle when it bought Sun in 2010. Developers can use MySQL under the GNU General Public License (GPL), but enterprises must obtain a commercial license from Oracle [10]. Today, MySQL is the RDBMS behind many of the top websites in the world and countless corporate and consumer-facing web-based applications, including Facebook, Twitter and YouTube.

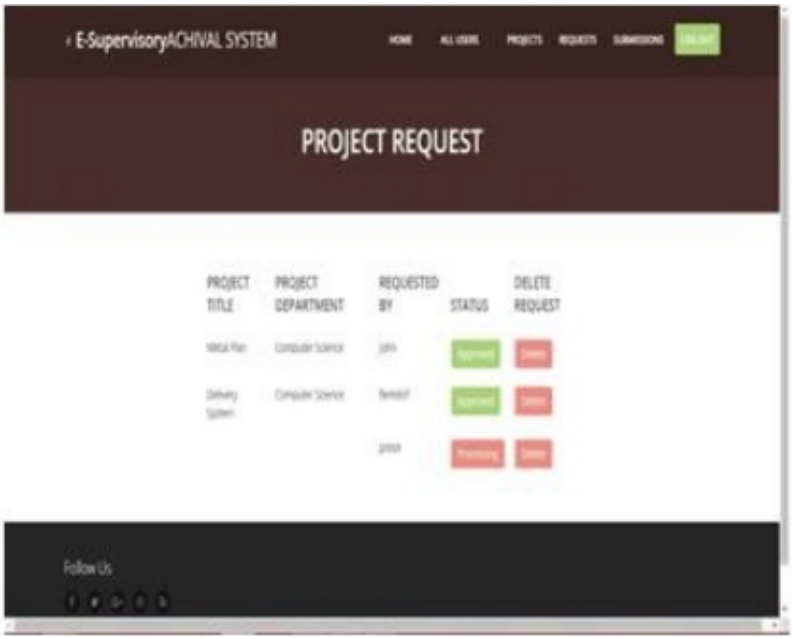


Figure 2: project requests that have been accepted by admin

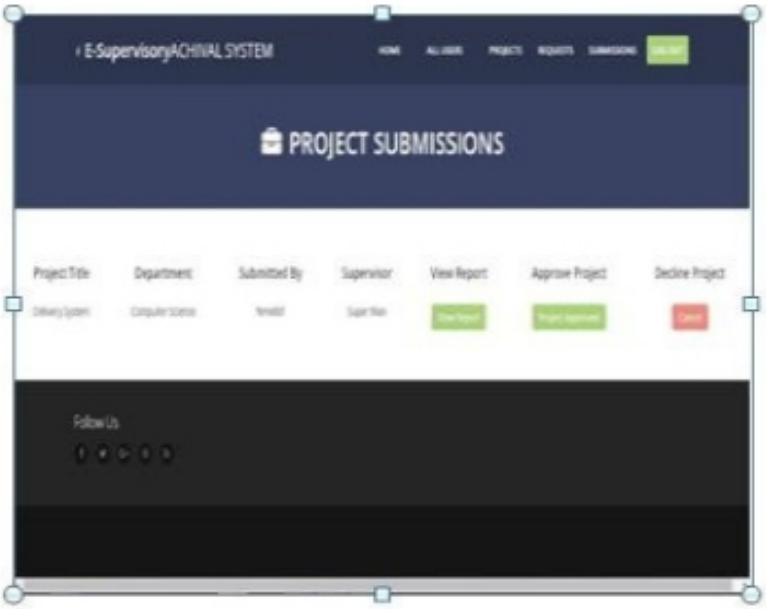


Figure 3: Project topics that has been submitted

Figure 1 shows the description of the relationship between the user, system and database while figures 2, 3, 4 shows the request interface of the E-Supervisory system. Figure 5 and 6 depicts the supervisor privilege and the project home page for students project status. Figure 7 depicts the platform where project topic is being request.

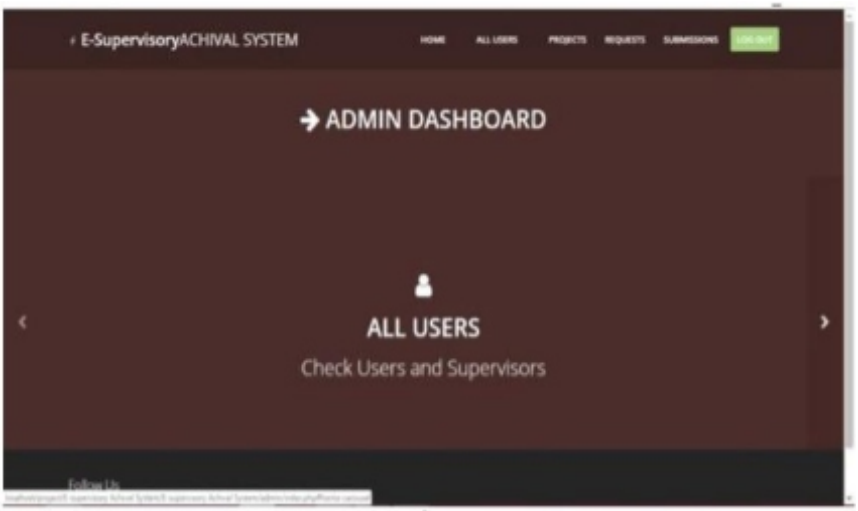


Figure 4: Admin Home page of the Designed Model

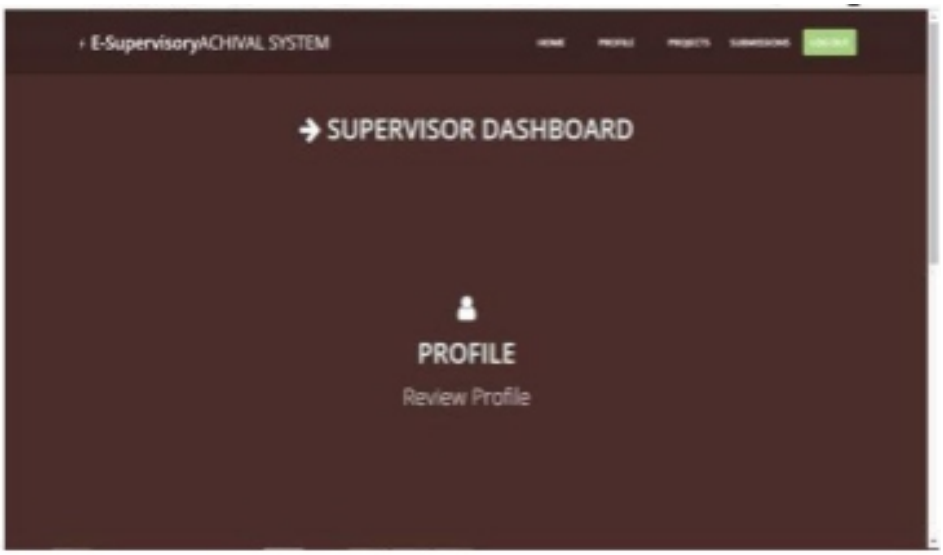


Figure 5: Supervisor Homepage

Figure 6: Home page of the student E-supervisory System

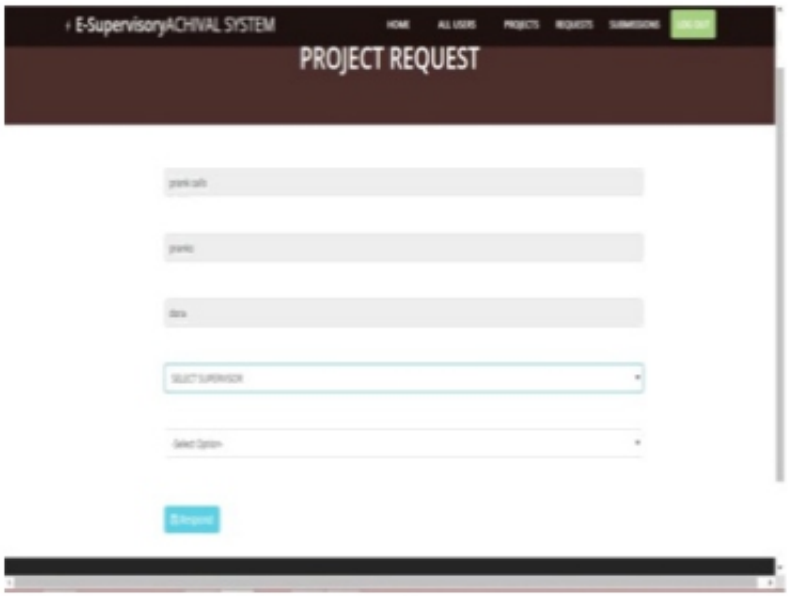


Figure 7: shows the platform used for project request

VI. CONCLUSION

This research has a positive impact in the academic development of the nation. Above all, this archival system was created in the hopes of making things better and easier for everyone that takes part in project writing. It‟s also helped in the storage of data for easy retrieval, curbing plagiarism and als in keeping track of different ongoing projects.

VII. RECOMMENDATION

For the betterment or further advancement of this research, a few measures can be carried out, like making password changes possible for every user and also add a list of projects yet to be done.

REFERENCE

[1] Alzahrani S. M., Salim N., and Abraham A., "Understanding plagiarism linguistic patterns, textual features, and detection methods," Trans. Sys. Man Cyber Part C, vol. 42, no. 2, pp. 133–149, Mar. 2012.

[2] Marini Abu, Norleyza Jailani, Zarina Shukur, Noor Faezah, Mohd Yatim," Final Year Supervision Management System as a Tool for Monitoring Computer Science Projects" Procedia Social and Behavioral Sciences 18 (2011) 273–281

[3] Plagiarism.org,http://www.plagiarism.org/plagiarism-101/typ es-of-plagiarism/ (last access August 18, 2019).

[4] Eneh, O.C. (2007a), "Improving Productivity in Nigerian Universities" Nigerian Journal of Development Studies, 6 (1): 35-56.

[5] Brin S., J. Davis, and H. Garcia-Molina, "Copy detection mechanisms for digital documents," in Proc. ACMSIGMOD Int. Conf.Manage. Data, New York, 1995, pp. 398–409.

[6] Imed Romdhani, Martin Tawse, Safa Habibullah, "Student Project Performance Management System for Effective Final Year and Dissertation Projects Supervision", In: Society, (Ed.) London International Conference on Education LICE. London, UK (2011).

[7] Mohammed Manaz, Mohammed Murshid, Mahammad Zameel , Avinash M " Medical Data Archive for Prevention of Epidemics" International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Special Issue 7, October 2015.

[8] Yousef k. Sanjalawe and Mohammed Anbar ."The Plagiarism Detection Systems for Higher Education - A Case Study in Saudi Universities" International Journal of Software Engineering & Applications (IJSEA), Vol.8, No.2, March 2017

[9] Romli, T. R. M., Othman, M. Z., Abdullah, M. H., Hamat, M. Z. A." Word Classification in the Online Database of Malay-Arabic Comparable Phrases". International Journal of Academic Research in Progressive Education and Development, 7(4), 255–266, 2018.

[10] Nixon, R. Learning PHP, MySQL, JavaScript, CSS & HTML5: A step-by-step guide to creating dynamic websites (3rd Edition). US: O'Reilly Media 2014.

[11] Setiyadi A. and Setiawan E B., "Information System Monitoring Access Log Database on Database Server", Conference. Series: Materials Science and Engineering 407 (2018) 012110

[12] Oliha F O., Ebietomere E P. and Ekuobase G O. " An Electronic Birth Record Management System For Nigeria" Nigerian Journal of Technology (NIJOTECH) Vol. 38, No. 3,July 2019, pp.763 –768

Modernized Assistive Technology to Expedite Judicial Procedure using Machine Learning and Natural Language Processing Techniques

[1] Saarah Asad, [2] S Aishwarya, [3] Payal A Makhija
[1,2,3]student B.E., Visvesvaraya Technological University, Bengaluru, India
E-mail: [1]saarahasad26@gmail.com, [2]aishwaryyaa12397@gmail.com, [3]payalmakhija17@gmail.com

ABSTRACT

The Indian Judiciary stacks a pile-up of more than 30 million cases, some in backlog for more than 10 years. Prominent cases such as the 1984 Sikh riots still await a final decision with many of the accused and those seeking justice having died in the 32-year long wait. This paper aims to present an application that would assist the jury and expedite the process of delivering a judgement. Given required documents at hand, judges take a minimum of one month to an infinite number of years to deliver the judgement. This paper proposes a system performing the same task in less than a minute. The application aims to summarize judicial support documents constructing an overview of them using NLP methods, suggest referral cases with IR techniques and propose a possible assistive judgement with machine learning techniques.

Keywords - Machine Learning (ML), Natural Language Processing (NLP), Information Retrieval (IR), Indian Judicial System, Document Summarization

I. INTRODUCTION

The Indian judicial system stands as prime pillar of the Indian democracy. But for all its sovereignty, it remains volatile, deficient in staff and above all slow. A person above 50 years of age, shouldn't even think of getting his case resolved through courts in his lifetime! The Judicial system is in crucial need of reforms if we are to continue to strengthen our society that dwells within a framework of fair justice for all.

Acquiring justice is riddled with obstacles from the very start for an average Indian, from filing an FIR to escalating it to lower courts. After the court proceedings end, a judge could take from a minimum of one month to more than 10 years to read documents relevant to the case, research and write the judgement. Additionally, the great number of vacancies existing in the courts makes the scenario truly distressing.

We propose a system that uses NLP, IR and machine learning techniques to perform the judge's tasks in less than a minute, originally taking a minimum of one month. The system aims to summarize judicial support documents constructing an overview of them, suggest referral cases similar to the one in focus and propose a possible assistive judgement. Therefore, accelerating the judicial process and decreasing the backlog of cases.

This paper is organized as follows. Section II describes the challenges faced by the judicial system and reason for backlog of cases. Technique for summarization of judicial support documents is given in section III. The technique and information retrieval model to suggest reference cases is discussed in Section IV; while Section V presents the technique used for constructing possible assistive judgement. Finally, conclusions are presented in section VI.

II. MODERNISATION OF COURTS

The Indian justice system is a principal pillar of the Indian democracy. India's judiciary follows an integrated three-tiered system. The Supreme Court, an all-india forum at the apex. High Courts crown each state's judiciary. Lower courts or district level courts form the third tier [1]. This section describes the judicial process after formulation of arguments and documents, reasons for delay in delivery of judgements and the methods to accelerate judgement delivery.

A. The Judicial Procedure

In this paper, we consider only those cases that come under the civil law sector and are under the district court jurisdiction. The process a case undergoes can be very briefly summarised as:

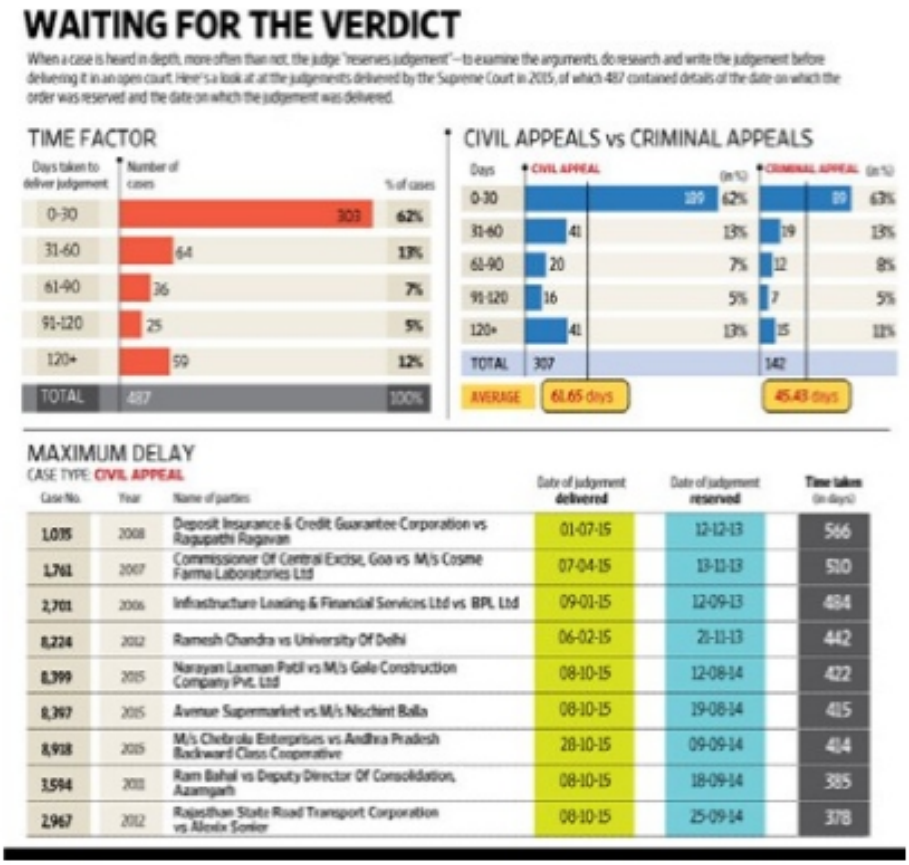


Fig 1 . Graph from Kumar, Alok Prasanna. "Delayed Justice: When Judgement Day Arrives Too Late." https://Www.livemint.com/, Livemint, 6 June 2016.

1. Once the FIR is filed, the police commence investigation and collect evidence pertaining to the case.
2. If the case is not resolved, it is documented and submitted to the lower court and proceedings commence. All proceedings are documented.
3. After the case has been heard in depth and arguments are advanced by both the parties, the judge usually reserves judgement for a period that could be minimum of three weeks to an infinite number of years, in order to examine the arguments, do research and write the judgement before delivering it.

B. Backlog of Cases

According to official figures, more than 27 million legal cases are pending in various courts in India, 6 million of which have been stuck in courts for 5 years or more. India has a mere 16,000 courtrooms and the bare minimum number of judges to preside over and carry out hearings. For every 73,000 people in India, there is only one judge.

A present-day survey of litigants conducted by Daksh, "a civil society organization that undertakes research and activities to promote accountability and better governance in India", showed that greater than 60% of the respondents found that the set-back in their own cases was due to the not delay of judges in making judgements [2].

It is the delay on step 3, from the previous subsection, that this paper aims to rectify. The figure above represents statistics of time taken for judges to deliver orders on cases.

C. The Fix

The figure summarises that it would approximately require 2 months for a judge to research and write the judgement. We propose a system performing the same task in less than a minute!

III. DOCUMENT SUMMARIZATION

Legal documents are documents that state some contractual relationship or grant some rights or could be documentation of court proceedings or could be documents submitted by lawyers or advocates. These documents provide information that is official in nature. Using text summarization, we aim to summarise these documents to save the time it would take a judge to read and research on them. We incorporate a technique which results in a high percentage accuracy in including crucial, case deciding statements of paramount importance in the overview. Text summarization addresses both the problem of selecting the most important portions of text (extractive text summarization)and the problem of

generating coherent summaries. Extractive text summarization methods work by identifying important sections of the text. The extractive text summarizer performs these three relatively independent tasks : Intermediate representation, Score sentences and Select summary sentences [3].

A. Intermediate Representation

Almost every text summarization system creates some form of intermediate representation of the text it intends to summarize. There are two approaches: "topic representation" and "indicator representation". "Topic representation approaches transform the text into an intermediate representation and interpret the topic(s) discussed in the text." It includes frequency, TF-IDF and topic word approaches, lexical chain approaches, latent semantic analysis and full blown Bayesian topic models. "Indicator representation approaches describe every sentence as a list of features (indicators) of importance such as sentence length, position in the document, having certain phrases, etc."

B. Score Sentences

On obtaining the intermediate representation, each sentence is assigned a score to indicate its importance. For "topic representation approaches", the score indicates how well a sentences expresses the important topics in the document or how well it combines information about different topics. "In most of the indicator representation methods, the score is computed by aggregating the evidence from different indicators".

C. Select Summary Sentences

The summarizer selects the k most important sentences to generate a summary. The factors that need to be taken into account could include, for example, the context in which the summary is being created. The type of the document can be another factor that could impact the sentences that are selected.

IV. SERVICEABLE REFERRAL TECHNOLOGY

Our application provides judges with references to similar cases, so as to assist them to deliver a judgement for the case in focus. Achieving this is a two step procedure: extraction of key terms from case in focus and information retrieval using vector space model.

A. Extraction of Key Terms from Case in Focus RAKE (Rapid Automatic Keyword Extraction) is an unsupervised algorithm to automatically extract keywords from documents. Using stop words and phrase delimiters, keyword extraction commences by parsing the text in the document into a set of candidate keywords, which are sequences of content words as they appear in the text. The document is split into an array of words, which is then split into sequences of contiguous words at phrase delimiters

and stop word positions. Words within a sequence are considered a candidate keyword. Once the candidate keywords are identified and the graph of word co- occurrences is constructed, a score is calculated for each candidate keyword. “After candidate keywords are scored, the top T scoring candidates are selected as keywords for the document”[4].

B. Information Retrieval using Vector Space Model

Information retrieval deals with the storage, organization, retrieval and evaluation of information relevant to the query constructed using the key terms. The retrieval techniques retrieve those documents that seem relevant to the query. Several different information retrieval models have been developed.

These models differ in the manner queries and documents are represented and process of information retrieval. One of the most well-studied retrieval models is the vector space model.

The vector space model represents queries and documents as vectors of features representing terms that occur in them. A multi-dimensional space represents these vectors. Each dimension represents a distinct term in the corpus of the document [5]. We explain it by walking you through an example.

Consider a collection C comprised of three documents:

D1: “disputed land complainant”

D2: “disputed land suit”

D2: “ponzi scheme complainant”

Some terms appear in more than one document. The total number of documents is N = 3. Therefore, the idf values for the terms are:

T1: complainant	log2(3/2)=0.584
T2: disputed	log2(3/2)=0.584
T3: land	log2(3/2)=0.584
T4: ponzi	log2(3/1)=1.584
T5: scheme	log2(3/1)=1.584
T6: suit	log2(3/1)=1.584

For all the documents, we calculate the term- frequency (tf) scores for all the terms in C.

	T1	T2	T3	T4	T5	T6
D1	0	0	1	0	1	1
D2	0	0	1	1	0	1
D3	1	1	0	0	1	0

Multiplying the tf scores by the idf values of each term, the documents-by-terms matrix is evaluated as:

	T1	T2	T3	T4	T5	T6
D1	0	0	0.58	0	0.6	0.58
D2	0	0	0.58	1.584	0	0.58
D3	1.58	1.58	0	0	0.6	0

Query: "disputed disputed complainant".

Similarly, we calculate the tf-idf vector for the query, also dividing the frequency by the maximum frequency which is 2 in this case.

a 0 0 (2/2)*0.584=0.584 0 (1/2)*0.584=0.292 0

The length of each document and query is calculated as:

$L_{D1} = \sqrt{(0.584^2+0.584^2+0.584^2)} = 1.011$

$L_{D2} = \sqrt{(0.584^2+1.584^2+0.584^2)} = 1.786$

$L_{D3} = \sqrt{(1.584^2+1.584^2+0.584^2)} = 2.316$

$L_{Q} = \sqrt{(0.584^2+0.292^2)} = 0.652$

The score of each document in C relative to this query, using the cosine similarity measure is calculated. Then the similarity values are:

cosSim(D1,Q)
=(0*0+0*0+0.584*0.584+0*0+0.584*0.292+0.584*0
) / (1.011*0.652)
= 0.776

Similarly,

cosSim(D2,Q) = 0.292
cosSim(D3,Q) = 0.112

With reference to these similarity values, the results for this query will be in the order: D1, D2, D3 [6]. Therefore, this technique is used to display the relevant cases to the case in focus, ranked in the order of relevance.

V. PREDICTIVE PROPOSITION TECHNIQUE USING MACHINE LEARNING

LJP (Legal Judgement Prediction) aims to predict the judgement of legal cases according to the facts of the case. Legal judgement consists of multiple subtasks, such as charges, term of penalty, fines, decision of the law articles that are applicable and so on. “Given the fact description of a specific case, a judge in the civil law system first decides which law articles are relevant to the scenario, and then determines the charges according to the instructions of relevant law articles. Based on these results, the judge further confirms the term of penalty and fines.” The topological dependencies among these subtasks are modelled with a DAG (Direct Acyclic Graph). “If the judgment of the j-th subtask tj depends on the output of the i-th subtask ti , then ti appears earlier than tj”. “TOPJUDGE” is a topological learning framework to handle multiple subtasks in LPJ and it models the explicit dependencies among these subtasks with scalable DAG forms [7].

VI. CONCLUSION

The Indian judicial system forms the foundations of India’s democracy. As the saying goes, justice delayed is justice denied. The judicial system has been failing to serve justice to millions with the pile-up of backlog cases that have been on the stack for years and years. Thus, we propose a system to accelerate the judicial process and decrease the backlog of cases. The application we propose summarizes judicial support documents constructing an overview of them using NLP methods, suggests referral cases with IR techniques and proposes a possible assistive judgement with machine learning techniques.

REFERENCES

[1] Bhuwania, Anuj. Courting the People: Public Interest Litigation in Post-Emergency India. Cambridge University Press, 2017.

[2] Kumar, Alok Prasanna. “Delayed Justice: When Judgement Day Arrives Too Late.” Https://Www.livemint .com/, Livemint, 6 June 2016.

[3] Allahyari, Mehdi, et al. “Text Summarization Techniques: A Brief Survey.” International Journal of Advanced Computer Science and Applications, vol. 8, no. 10, 2017.

[4] Rose, Stuart, et al. “Automatic Keyword Extraction from Individual Documents.” Text Mining, Apr. 2010.

[5] Siddiqui, Tanveer, and U. S. Tiwary. Natural Language Processing and Information Retrieval. Oxford University Press, 2008.

[6] SITE, www.site.uottawa.ca/~diana/.

[7] Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. “Legal judgment prediction via topological learning”. In Proceedings of EMNLP, 2018.

Instructions for Authors

Essentials for Publishing in this Journal

1 Submitted articles should not have been previously published or be currently under consideration for publication elsewhere.

2 Conference papers may only be submitted if the paper has been completely re-written (taken to mean more than 50%) and the author has cleared any necessary permission with the copyright owner if it has been previously copyrighted.

3 All our articles are refereed through a double-blind process.

4 All authors must declare they have read and agreed to the content of the submitted article and must sign a declaration correspond to the originality of the article.

Submission Process

All articles for this journal must be submitted using our online submissions system. http://enrichedpub.com/ . Please use the Submit Your Article link in the Author Service area.

Manuscript Guidelines

The instructions to authors about the article preparation for publication in the Manuscripts are submitted online, through the e-Ur (Electronic editing) system, developed by **Enriched Publications Pvt. Ltd**. The article should contain the abstract with keywords, introduction, body, conclusion, references and the summary in English language (without heading and subheading enumeration). The article length should not exceed 16 pages of A4 paper format.

Title

The title should be informative. It is in both Journal's and author's best interest to use terms suitable. For indexing and word search. If there are no such terms in the title, the author is strongly advised to add a subtitle. The title should be given in English as well. The titles precede the abstract and the summary in an appropriate language.

Letterhead Title

The letterhead title is given at a top of each page for easier identification of article copies in an Electronic form in particular. It contains the author's surname and first name initial .article title, journal title and collation (year, volume, and issue, first and last page). The journal and article titles can be given in a shortened form.

Author's Name

Full name(s) of author(s) should be used. It is advisable to give the middle initial. Names are given in their original form.

Contact Details

The postal address or the e-mail address of the author (usually of the first one if there are more Authors) is given in the footnote at the bottom of the first page.

Type of Articles

Classification of articles is a duty of the editorial staff and is of special importance. Referees and the members of the editorial staff, or section editors, can propose a category, but the editor-in-chief has the sole responsibility for their classification. Journal articles are classified as follows:

Scientific articles:

1. Original scientific paper (giving the previously unpublished results of the author's own research based on management methods).

2. Survey paper (giving an original, detailed and critical view of a research problem or an area to which the author has made a contribution visible through his self-citation);

3. Short or preliminary communication (original management paper of full format but of a smaller extent or of a preliminary character);

4. Scientific critique or forum (discussion on a particular scientific topic, based exclusively on management argumentation) and commentaries. Exceptionally, in particular areas, a scientific paper in the Journal can be in a form of a monograph or a critical edition of scientific data (historical, archival, lexicographic, bibliographic, data survey, etc.) which were unknown or hardly accessible for scientific research.

Professional articles:

1. Professional paper (contribution offering experience useful for improvement of professional practice but not necessarily based on scientific methods);
2. Informative contribution (editorial, commentary, etc.);
3. Review (of a book, software, case study, scientific event, etc.)

Language

The article should be in English. The grammar and style of the article should be of good quality. The systematized text should be without abbreviations (except standard ones). All measurements must be in SI units. The sequence of formulae is denoted in Arabic numerals in parentheses on the right-hand side.

Abstract and Summary

An abstract is a concise informative presentation of the article content for fast and accurate Evaluation of its relevance. It is both in the Editorial Office's and the author's best interest for an abstract to contain terms often used for indexing and article search. The abstract describes the purpose of the study and the methods, outlines the findings and state the conclusions. A 100- to 250-Word abstract should be placed between the title and the keywords with the body text to follow. Besides an abstract are advised to have a summary in English, at the end of the article, after the Reference list. The summary should be structured and long up to 1/10 of the article length (it is more extensive than the abstract).

Keywords

Keywords are terms or phrases showing adequately the article content for indexing and search purposes. They should be allocated heaving in mind widely accepted international sources (index, dictionary or thesaurus), such as the Web of Science keyword list for science in general. The higher their usage frequency is the better. Up to 10 keywords immediately follow the abstract and the summary, in respective languages.

Acknowledgements

The name and the number of the project or programmed within which the article was realized is given in a separate note at the bottom of the first page together with the name of the institution which financially supported the project or programmed.

Tables and Illustrations

All the captions should be in the original language as well as in English, together with the texts in illustrations if possible. Tables are typed in the same style as the text and are denoted by numerals at the top. Photographs and drawings, placed appropriately in the text, should be clear, precise and suitable for reproduction. Drawings should be created in Word or Corel.

Citation in the Text

Citation in the text must be uniform. When citing references in the text, use the reference number set in square brackets from the Reference list at the end of the article.

Footnotes

Footnotes are given at the bottom of the page with the text they refer to. They can contain less relevant details, additional explanations or used sources (e.g. scientific material, manuals). They cannot replace the cited literature.
The article should be accompanied with a cover letter with the information about the author(s): surname, middle initial, first name, and citizen personal number, rank, title, e-mail address, and affiliation address, home address including municipality, phone number in the office and at home (or a mobile phone number). The cover letter should state the type of the article and tell which illustrations are original and which are not.

Address of the Editorial Office:

Enriched Publications Pvt. Ltd.
S-9,IInd FLOOR, MLU POCKET,
MANISH ABHINAV PLAZA-II, ABOVE FEDERAL BANK,
PLOT NO-5, SECTOR -5, DWARKA, NEW DELHI, INDIA-110075,
PHONE: - + (91)-(11)-45525005