Integrating Waterfall Methodology and Object Oriented Approach within a Decision Support System Development Life Cycle Design

Zühre AYDIN
Energy Market Regulatory Authority, Energy Transition Department, Turkey

(Alınış / Received: 19.04.2022, Kabul / Accepted: 25.08.2022, Online Yayınlanma / Published Online: 30.12.2022)

Keywords
decision support systems,
object oriented programming approach,
software development life cycle design,
waterfall methodology

Abstract: This paper studies on design strategies and requirements of object-oriented programming approach within waterfall methodology, for developing a decision support system in a human resources system. The study proposes an integrated waterfall methodology and Object-Oriented Programming approach that is implemented during development phases. The aim of the decision support system is to help decision makers in the recruitment process and to propose a new integrated methodology for developers and managers. The application is a web-based system that supports structured decision making and provides information for company operations and decision making processes.

Bir Karar Destek Sistemi Geliştirme Yaşam Döngüsü Tasarımı Kapsamında Şelale Metodolojisi ve Nesneye Yönelik Yaklaşım

Keywords
karar destek sistemleri,
nesne yönelimli programlama yaklaşımı,
yazılım geliştirme yaşam döngüsü tasarımı,
şelale metodolojisi

Öz: Bu makale, bir insan kaynakları sisteminde karar destek sistemi geliştirmek için nesne yönelimli programlama içerisinde şelale metodolojisi yaklaşımının tasarım stratejileri ve gereksinimleri üzerinedir. Çalışma, geliştirme aşamalarında uygulanan bütünleşik bir şelale metodolojisi ve nesne yönelimli programlama yaklaşımı önermektedir. Karar destek sisteminin amacı, karar vericilere işe alım sürecinde yardımcı olmak, geliştiriciler ve yöneticiler için yeni bir bütünleşik metodoloji önermektir. Uygulama, yapılandırılmış karar vermeyi destekleyen, şirket operasyonları ve karar verme süreçleri için bilgi sağlayan web tabanlı bir sistemdir.

*Corresponding Author, email: zaydin@epdk.org.tr

1. Introduction

One of the big disputes about Decision Support System (DSS) implementations is that these software development's analysis and design phases take a substantial amount of time. Project plans and designing software architectures take much time of planning even with a large team. In addition to that project teams should choose an appropriate software development language and life cycle management methodology for better and adaptable software.

Object oriented software development provides many advantages for developing modular integrated processes. Object Oriented Programming (OOP) includes the actions of real or imagined objects. OOP languages, for instance C#, contains mixed designs within components like procedural designs and can be implemented for modelling at various levels of software model [1]. OOP based on data rather than facilities of integrated modules that include whole information to trigger data structures such as attributes and methods [2]. OOP is popular and it has many advantages agaiprocedural-orientedcedural oriented software programming. Object oriented designs are reusable since there is no need to build some facilities repititively. It is flexible and can avoid redundancy in the code. It can increase extensibility factor of classes and can develope secure programmable code. Functions, with number of processes, share the global data, transfer data and focus on systematical design from beginning to ending [3]. All of these factors provide decreasing time and investment during software developing.

OOP is the best choice in project planning of DSS developments, because of OOP software developing objectives such as security, polymorhism, inheritance, design benefits, extensibility, flexibility, reusability, data redundancy, code maintenance and easy troubleshooting.

In a Human Resources System (HRS) DSS life cycle management there should be definition of purpose of expected software, process analysis, project planning, database and software design, database and software development, security and end user testing, deployment, feedbacks and maintenance phases. According to these phases, the Waterfall life cycle management methodology was chosen in this study, since this methodology is a linear sequential life cycle model. It has a linear sequence in which the next step depends on the efficient completion of the previous one. Especially completion of requirements analysis and system design that are prepared carefully and meticulously are important phases. The programmers have to consider project phases when they plan to develope a DSS sequentially in development of the project that aims to help decision makers in the recruitment process and to propose a new integrated methodology for developers and managers as a web-based decision support system.

The study continues with literature review in section 2, research method with the proposed software design, logic and architecture subsections, implemented life cycle phases and waterfall methodology of the proposed DSS in section 3, implemented and proposed software life cycle with integration of Waterfall methodology and OOP approach in section 4 and conclusion in section 5.

1.1. Decision Support System

DSS helps decision makers to determine and solve semistructured and unstructured problems within data, models and knowledge. Decision making processes was firstly revealed by Carnegie Institute of Technology and was applied as a programming by Massachusetts Institute of Technology (MIT) in the 1960s. In the 1980s, detailed research of DSS was studied from single-user DSS to organizational DSS, group DSS and executive Information Systems (IS). Gorry et al. [4] developed DSS as an interactive software system that provided easy access to information for decision makers. In the 1990s, DSS improved through online analytical processing and data warehousing [5].

DSS could support decisions of theorganization’s operations for future strategies. DSS can be applied in different management information systems, such as diagnostic in healthcare, supply chain management, educational applications, human resource systems.
The selection of a DSS is depend on objective of the DSS problem, organization structure, problem structure (semistructured and unstructured), decision makers’ experience and organizationexpectancyy. Hence complexity and structure of the problem is very important in DSS developmentt.

1.2. Design Strategies and Requirements in Decision Support Systems’ Software Architecture

From early 1960s, to now, many DSS design strategies, requirement analysis and development methodologies in the context of software architecture have been developed. Sprague et al. [6,21] stated DSS framework components such as; data and model base, dialog generation and software management. The purpose of this research is to evaluate the most approprite methodology for a DSS within HRS structure. Modern DSS designs and modern procedural languages reduce time for decision making, reduce unnecessary project investments, provide necessary information, present effectiveness in organizational processes and competitive advantages.
Development methodologies of DSSs are decision driven, process driven, data driven and system driven methodologies and they specify squence of facilities of DSSs.

In the context of project management and software architecture, main strategies and requirements of DSS phases are; identification of objectives, cost benefit analysis, risk assessment, initial project complexity, planning of scope planning, project schedule, resource planning (software platform, hardware, project team, investments), risk management planning, control change management, analysis of business needs and analysis of requirements for database and model design, execution of functional, nonfunctional or unit tests, implementation, feedbacks and maintenance.

In design strategies choosing a software platform is very important. OOP languages depend on unified desing and its systematic procedural diagrams involve classes, use cases, development, communication and sequence diagrams, and these advantages provide dealing with complexity of the objects [7].

When OOP design is applied to DSS systems, it can provide easy to maintain and secure code development since its design strategy is based on classes, objects and in place facilities which are structural functional decomposition of DSS. The summary of the software goals and requirements an, advantages of a DSS through modern procedural languages which depend on OOP concepts are presented as in Figure 1.

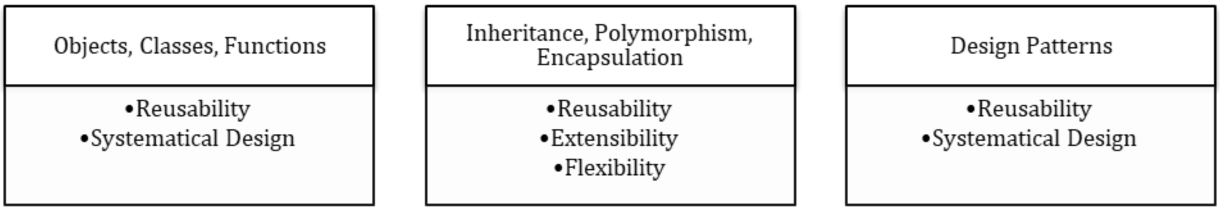


Figure 1. A DSS software requirements through OOP context.

According to the Figure 1, systematical design with objects and classes use attributes and functions to code given data. Inheritance provides using the same code blocks just by changing the definition of related variables. Reusability of attributes and methods ensures feasibility and easy coding. Polymorphism represents many different types of variables in only one code and binds the objects at various positions with various functions.

1.3. Waterfall Methodology

DSS success relies on selecting a software development life cycle approach that is most suitable for defined DSS objectives and related project phases. DSS project management and development are complex and have detailed facilities that requires detailed analysis of integrated modules, experience on analysis, development and testing phases, systematical system design, such as accurate database and programing designs.

The effects of change requests due to requirement imperfections on the outputs of software projects are essential for sequential developing in Waterfall methodology [8]. Since system requirement analysis and design are most essential part of a DSS project software life cycle management, Waterfall methodology was chosen in this paper for sequential phases.

The Waterfall methodology was presented by Royce [9] to fill the gap in sequential software project management and is widely implemented in projects. The sequential phases of the methodology are; requirement analysis, system design, coding, testing and debugging, deployment and maintenance as illustrated in Figure 2.

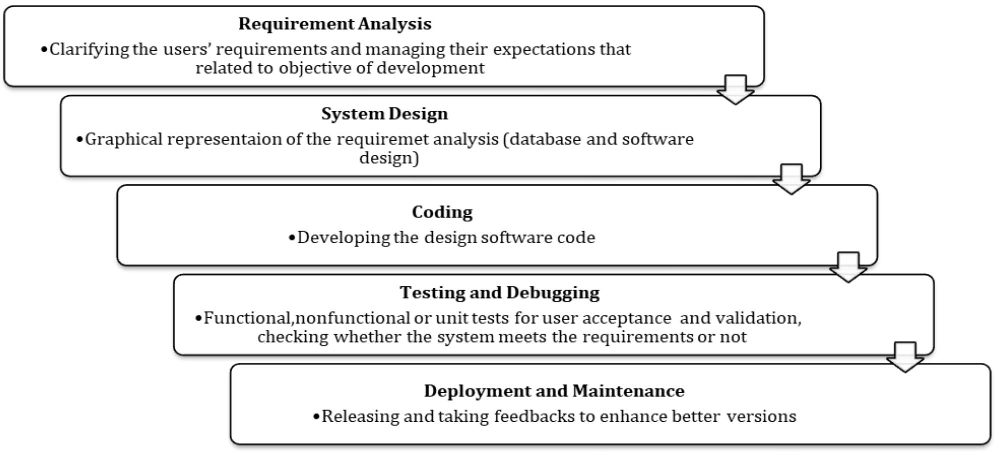

Requirement Analysis
•Clarifying the users' requirements and managing their expectations that related to objective of development
System Design
•Graphical representaion of the requiremet analysis (database and software design)
Coding
•Developing the design software code
Testing and Debugging
•Functional,nonfunctional or unit tests for user acceptance and validation, checking whether the system meets the requirements or not
Deployment and Maintenance
•Releasing and taking feedbacks to enhance better versions

Figure 2. Life cycle phases of Waterfall methodology [9]

Objectives of an organizational HRS DSS is designed in a stable context generally. This design strategy provides choosing Waterfall methodology because of its is easy implementation and manage, clear documentation of needed requirements, review processes of each phase, sequential design of phases. The disadvantage of the methodology is that, when starting to the software development it is diffucult to change the requirements, so it should be fixed at the beginning. On the other hand, fixing the requirements at starting can be an advantage for clear and easy Development [16].

2. Literature Review

To help the human resource decision makers of an organization, Luk et al. [10] stated using of automated processes with web services that can rise the efficiency and accuracy of choosing approprite employers for the organizational needs. Power et al. [11] researched model driven DSSs to make better decisions. They developed model driven DSS framework through using decision support, operational research and simulation technologies. Detienne [12] studied on design strategies on Object Oriented Programming within procedural and OOP languages. Veronica [13] stated that, DSS success relates with developers' competence on the development tools and processes and relates with determining a development approach that is suitable for the DSS problem. The purpose of the paper is to overview the DSS project life cycle methods through DSS developments. Study used waterfall methodology for system development life cycle within main stages such as; problem definition, analysis, design, programming and implementation. Samad [14] studied on human resource Decision Support System that includes recruitment process. According to research selecting potential employees through qualities such as education and qualification, is necessary for reducing time and effort in decision making. Chack et al. [15] studied on main tools that comprise integrated Human Resource DSS and they developed a multi agent DSS that is divided into strategic, tactical and operational management layers. Soesanto et al. [16] proposed a web based DSS which supports decisions in the scope of machinery and laboratory and it uses waterfall methodology. Subhan et al. [17] presented a detailed analysis of agile software development methodology by studying on requirements and design factors of different developments. Ramanathan et al. [18] determined the projects implemented by different studies to present ideas for automated intelligent DSS problems that decision makers face with in Human Resource Management (HRM) projects; and how these problems impact Software Project Management (SPM) outputs. Urdhwareshe [3] presented benefits, importance and impacts of OOP on software development that based on reusability, extensibility and flexibility. This study also explains the importance of OOP and independent factor model through; object, class, inheritance and polymorphism. Soleman [19] tried to create a DSS by Analytic Hierarchy Process (AHP) in weighting and by Profile Matching (PM) in ranking for an employee selection system. This system aims to save time and money by this automization and uses three testing methods for validation. Martins et al. [20] proposed a DSS that evaluates business ideas competing projects by using C# and AHP to serialize projects. This study provides AHP method to simplify the assessment process of competing.

The purpose of the paper is to overview the DSS project life cycle methods through DSS developments. This study used waterfall methodology for system development life cycle within main stages such as; problem definition, analysis, design, programming and implementation. Contribution to the literature will be with a developed DSS that implements Waterfall methodology and OOP approach and with a proposed methodology that integrates implemented approach.

3. Research Method

3.1. Development Objectives, Analysis and Sofware System Design

This study presents the logical design of a DSS software that is implemented for a company or an institution to choose confidently the Expert Staff Recruitment (ESR) DSS for designated projects. Company can use decision support system by recording the information of the applicants into the database and can create software employee team of the specified projects that need to be developed in order to make the budget and document transactions in electronic environment. The project development language is C # .NET development environment and the database platform is MySql. The proposed DSS is a web-based application and aims to rank qualified personnel in a final report, according to candidates' total scores. Candidates' registrations are entered during application process.

The development objectives of the system are;

- Computer-aided decision making for the company projects
- Supporting managers in planning and control
- Creating a successful application in the field of human resources
- Providing secure and fast access by adding the information of applicants to the database

- Achieving reliable results electronically by ensuring that the information added to the database is passed through a scoring system
- Implementing OOP concepts through OOP goals
- Following waterfall software project life cycle phases for serial and easy going
- Integrating OOP approach with Waterfall methodology

In system design; user qualifications of system user are determined as making the data entry of the candidates, organizing the data and performing the necessary reporting. In the development process; things to do before development are devised as analyzing of software requirements, processes, algorithms, database design, software design and preparing the design documents.

In the context of admin user requirements; secure logging into the system with username and password should be included. Admin user should have has the permissions such as; adding, editting, deleting, updating, selecting, viewing all the candidates' information. Finally expert candidates can be determined as a result of the system's guidance and decision makers can use the personnel recruitment decision support system and can make decisions neutrally and fairly as system users.

Use case diagram is used to model the processes of ESR DSS from the point of view of system user. In the analysis and design phase of a DSS Development, good modeling prevents future conflicts, so that UML is necessary for object oriented programming languages. UML breaks downs our defined objectives into parts, and creates connections between these parts. The requirements are mapped into the UML use case diagram as in Figure 3.

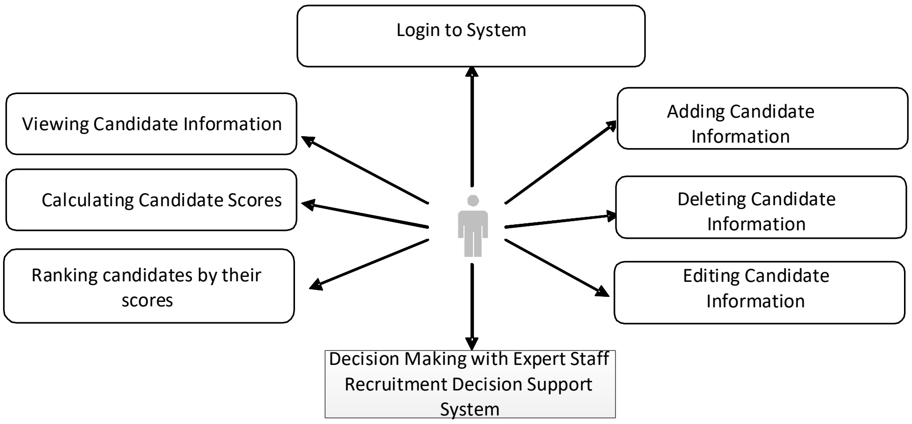

Login to System
Viewing Candidate Information
Calculating Candidate Scores
Ranking candidates by their scores
Adding Candidate Information
Deleting Candidate Information
Editing Candidate Information
Decision Making with Expert Staff Recruitment Decision Support System

Figure 3. UML use case diagram of ESR DSS

3.2. Logic and Workflow of the ESR DSS

In related ESR DSS, decision maker can calculate scores of the project candidates/applicants whose qualifications determined by the company through candidates' education, experience, previous project area. In addition, decision maker can record applicants' information in a score table as a final ranking report. The registration of these processes are visualized in Figure 4 as activity diagram.

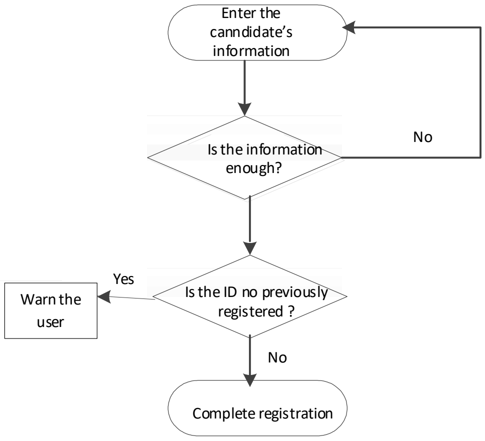

Enter the canndidate's information
Is the information enough?
No
Is the ID no previously registered ?
Yes
Warn the user
No
Complete registration

Figure 4. Activity diagram of registration process of ESR DSS

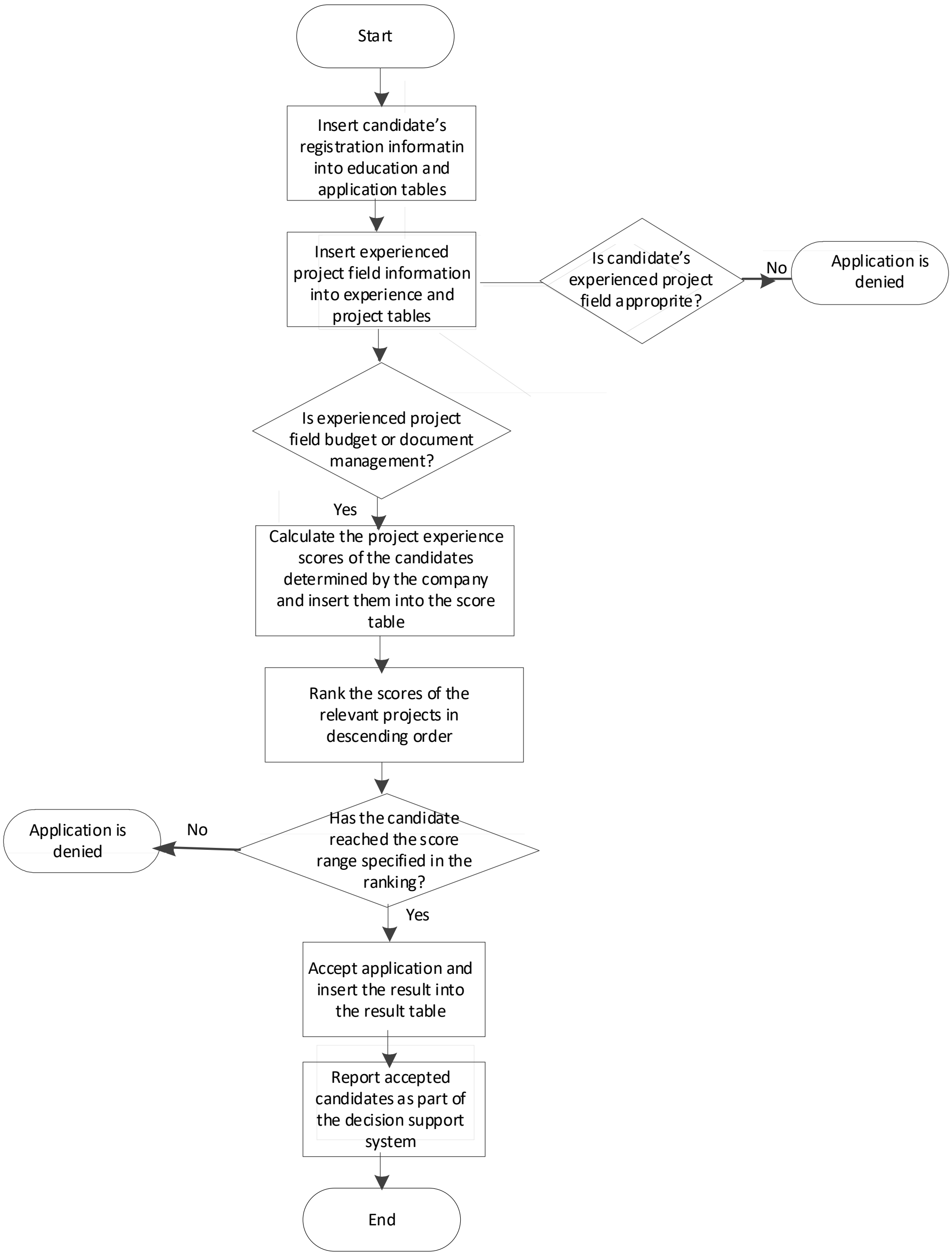

Start
Insert candidate's registration informatin into education and application tables
Insert experienced project field information into experience and project tables
Is candidate's experienced project field approprite?
No
Application is denied
Is experienced project field budget or document management?
Yes
Calculate the project experience scores of the candidates determined by the company and insert them into the score table
Rank the scores of the relevant projects in descending order
Has the candidate reached the score range specified in the ranking?
No
Application is denied
Yes
Accept application and insert the result into the result table
Report accepted candidates as part of the decision support system
End

Figure 5. Flowchart of proposed ESR DSS

In Figure 5 logic and workflow of the proposed ESR DSS is shown by a flowchart.

The database actor design consists of key users, end users, candidates and decision makers who will benefit from the reports of the decision support system. The first records of the database were made by the authorized user by entering the information of the candidates who completed their applications of the previous application of a solution partner company. The finalization of the system reports was ensured by the feedback of the decision makers.

In study's database design and requirements process relational MySql database platform is used. In this environment, enforcing relationships, determining each table's primary key, including a common field in related tables, using appropriate data types, including indexes during developing, avoiding data redundancy and standardizing variables, schemas and objects naming are carried out. Through these features, the database tables of ESR DSS are shown below.

Database tables:

DBTable1_Candidate Application Table:

- *Table ID Number*
- *Candidate identification number*
- *Candidate name and surname*
- *Education Table ID number*
- *Project Area Table ID number*
- *Experience Chart ID number*
- *Results Table ID number*

DBTable 2_Project Field Table:

- *Table ID Number*
- *Project name*
- *Project Score (is determined by company*)

DBTable 3_Education Table:

- *Table ID Number*
- *Education Name*
- *Education Score (is determined by company)*

DBTable 5_Experience Table:

- *Table ID Number*
- *Experience Name*

DBTable 6_Score Table:

- *Table ID Number*
- *Application ID number*
- *Total Score*

DBTable 7_Results Table:

- *Table ID Number*
- *Result (Information on whether the candidate has been accepted)*

ESR DSS security requirements should provide system and information security with username control, password policies, ports, database authorization, permissions, roles, schemas security in database and secure code development.

As privacy requirements; system does not allow unauthorized users to access the screens of different modules'. As computer resource requirements; the software requirements are configured by C# programming language and MySql database platform. As a design and implementation constraints of the development; user interaction with the system is provided by authorized users. In addition, every user interface will have a standard screen view and will differ according to its functionality. In staff requirements; staff is expected to have a basic knowledge of computer use in order to use the ESR DSS application. In system test requirements phase; after the system development phase is completed, functional system accuracy tests are made.

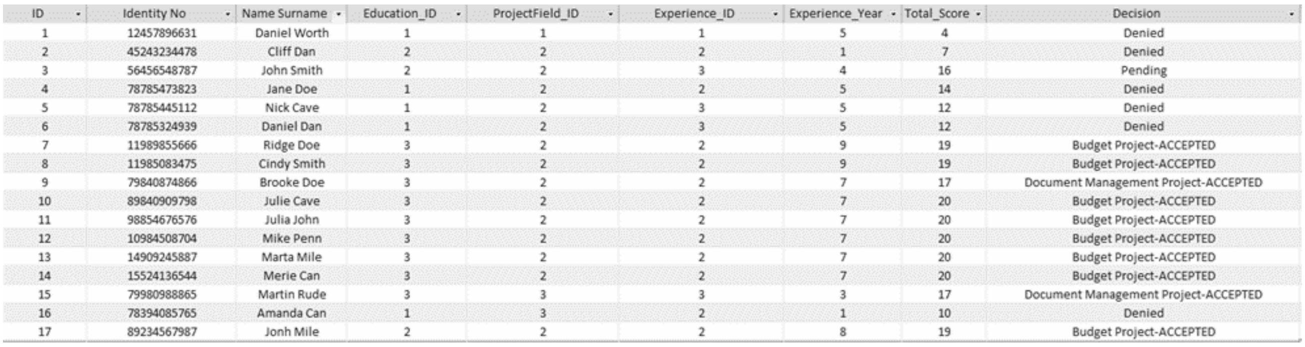
In terms of development design, inheritance and encapsulation in OOP provided heritage case in classes that have more than one function and these functions are used in other pages and classes. For capsulation case; ESR DSS defines public variable to access global variable that is created in outer class. Sometimes developers may need to use these variables even though they don't want to change their values. In this case, there is no need to encapsulate the variable. So, developer need to define this variable as private and remove write permission of this variable with a read only property.

Authorized user enters all the the required applicants information such as; identity no, name, surname, telephone no, e-mail, education, experience, project field into the application interface. User can search, edit and update previous records from application interface. In application check interface user can query for recorded candidates and can query candidates according to their ranked scores.

Total score of a candidate is calculated from education scores and project scores of related tables and these scores are determined by company. The addition of education and project scores are inserted into total score field.

Total scores of score table results determine the information whether the candidate has been accepted or not in the results table.

Application results interface includes both "query for candidate" button and "report" button. "Report" button provides ESR DSS objective that want to show decision on candidates according to their total score and want to choose best candidate for planned budget project and document project resource management. Figure 6 shows application result interface and ESR DSS's aimed result report.



ID	Identity No	Name Surname	Education_ID	ProjectField_ID	Experience_ID	Experience_Year	Total_Score	Decision
1	12457896631	Daniel Worth	1	1	1	5	4	Denied
2	45243234478	Cliff Dan	2	2	2	1	7	Denied
3	56456548787	John Smith	2	2	3	4	16	Pending
4	78785473823	Jane Doe	1	2	2	5	14	Denied
5	78785445112	Nick Cave	1	2	3	5	12	Denied
6	78785324939	Daniel Dan	1	2	3	5	12	Denied
7	11989855666	Ridge Doe	3	2	2	9	19	Budget Project-ACCEPTED
8	11985083475	Cindy Smith	3	2	2	9	19	Budget Project-ACCEPTED
9	79840874866	Brooke Doe	3	2	2	7	17	Document Management Project-ACCEPTED
10	89840909798	Julie Cave	3	2	2	7	20	Budget Project-ACCEPTED
11	98854676576	Julia John	3	2	2	7	20	Budget Project-ACCEPTED
12	10984508704	Mike Penn	3	2	2	7	20	Budget Project-ACCEPTED
13	14909245887	Marta Mile	3	2	2	7	20	Budget Project-ACCEPTED
14	15524136544	Merie Can	3	2	2	7	20	Budget Project-ACCEPTED
15	79980988865	Martin Rude	3	3	3	3	17	Document Management Project-ACCEPTED
16	78394085765	Amanda Can	1	3	2	1	10	Denied
17	89234567987	Jonh Mile	2	2	2	8	19	Budget Project-ACCEPTED

Figure 6. Interfaces and running principles of the ESR DSS-Application results interface and DSS's aimed result report.

3.3. Implemented Software Architecture Framework

A personnel recruitment system is very important for a company's HRS DSS. This is an essential decision for a company in which the organizational development fosters. A long term and necessary success of an organization depends on its capacity to assess personnels' experience in recruitment process. A HRS DSS development has more strategic aspect and it is the key of qualified personnel of the organization.

A DSS software architecture is a set of software components, their relationships with connectors and features of both components and connectors [23].

While preparing a DSS software, layers can be created according to intended use of the software or it can be run on a single layer. Data layer, information layer, knowledge layer and presentation layer can be stated within the concept of sofware layers. Data layer is the first step that is used to bring and manage distributed data and meta information from the database system. Business layer enable data that is organized and controlled in accordance with our business. In detail, it is a set of business logic, a group of domain specific components and connectors that give valuable information for decision making and it includes components and connectors. Presentation layer has interfaces that receive the meta information that is proccessed in the previous layers and it also uses the web for a web based DSS [23,24,25,26].

In this paper, a multi layered sofware achitecture is planned in ESR DSS. The hierarchical sofware design consists of data access, business, application, security and presentation layers. These layers are devised through OOP design and concepts. They are used to share OOP components to make most effective use of code.

In our ESR DSS OOP software architecture; data layer designs database tables, variables and relations through analysis and requirements, in addition it includes sql queries, functions, views and stored procedures. Data access layer includes data access components and service gateways that configures security roles for providing multiple databases access to application. Business layer provides value added information from database to aggregate data into information by business logic for reporting on decision making process. Application level security layer links the credentials to active directory, manages authentication for user accounts on domain, in summary ensures security and access to application. In big DSS systems security layer should include logging and security credentials should be encrypted and stored in database. In conclusion; presentation layer presents user interface to provide communication between the user and the DSS. This layer includes forms, reports, user and application controls, master pages, scripts, configuration files, presentation logic.

This study, implements a hierarchical structure of ESR DSS processes. Figure 7 presents the multi layered software architecture of web-based ESR DSS as implemented software architecture.

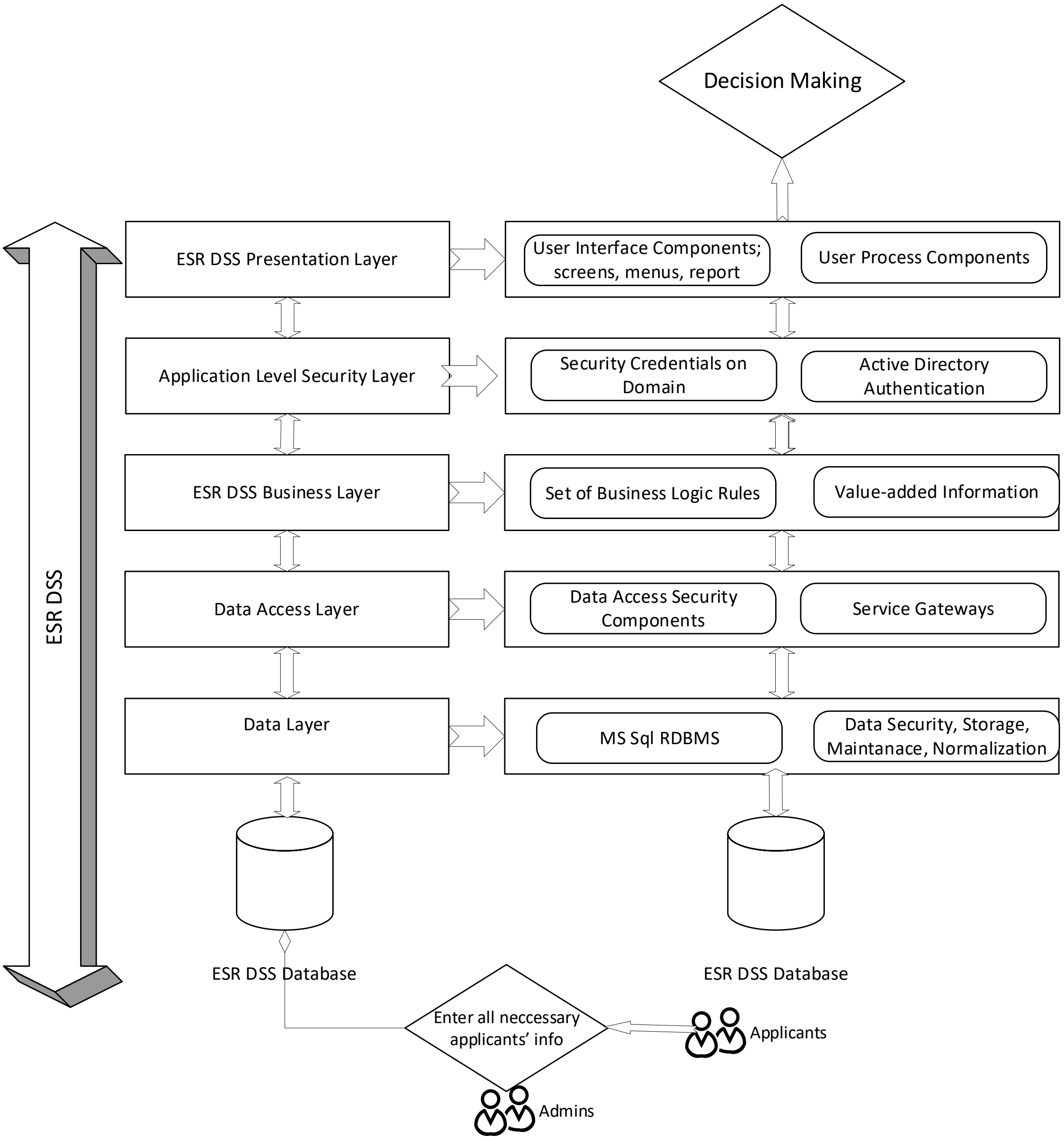

Decision Making
ESR DSS
ESR DSS Presentation Layer
User Interface Components; screens, menus, report
User Process Components
Application Level Security Layer
Security Credentials on Domain
Active Directory Authentication
ESR DSS Business Layer
Set of Business Logic Rules
Value-added Information
Data Access Layer
Data Access Security Components
Service Gateways
Data Layer
MS Sql RDBMS
Data Security, Storage, Maintanace, Normalization
ESR DSS Database
ESR DSS Database
Enter all neccessary applicants' info
Applicants
Admins

Figure 7: Implemented and proposed layered ESR DSS Framework

In designing the software the study tried to apply waterfall methodology for easy going and easy arrangement by a sequential project life cycle management and implemented OOP concepts.

4. The Implemented and Proposed Software ESR DSS Life Cycle with Integration of Waterfall Methodology and OOP Approach

During the development, implementation of waterfall methodology was done by requirement analysis, system design, coding, testing and debugging, deployment and maintenance life cycle project management phases.

These phases are implemented through ESR DSS objectives that try to provide easy and accurate decision making on a recruitment decision and assignment and evaluation of candidates for determined projects of company.

In development process; software developers must identify what they are going to develop. Structured analysis and design techniques are mentioned in section 3 with activity diagram, use case diagram, flowchart and relational database table components. To overcome the different understanding levels of objectives between stakeholders, system requirements should be written down as a requirement document and in the literature this document known as Software Requirements Specification (SRS) document [21]. SRS contains both functional and nonfunctional requirements, such as; what the application does and what the end user wants. In this study's concept, requirements gathered from stakeholders should be written in a SRS document for preventing conflicts and saving time between developers and stakeholders. This document also decreases project costs since it provides agreement and time saving.

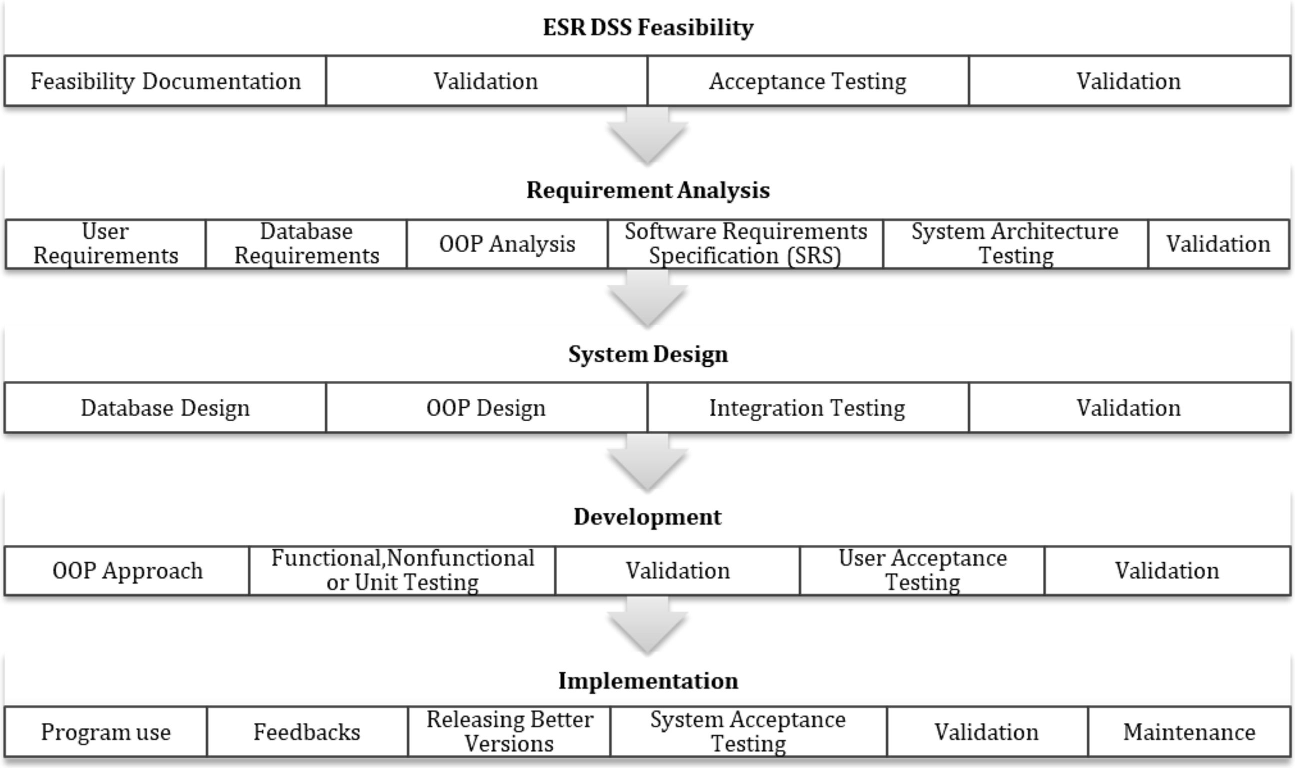
SRS helps fast and easy going system design, coding, testing and debugging, deployment and maintenance facilities.

OOP software design is a visualization of SRS that consists of application architecture, detailed software design, implementations of all customers' specifications, relationships between different components of system such as database and software relationships. In previous sections, study explained how ESR DSS software design is planned through user requirements, software requirements, processes, algorithm, database design and OOP approach software design. This phase allows developers to provide the accuracy of the all system since it is a communicator between SRS document and implementation.

In HRS DSS projects, design of database is an essential part of system design phase and development phase which includes instance mapping, data and schemas. In the database concepts of the OOP approach provides, efficiency for reusability of its application layers and transferring the existing data to planned database shema.

Functional, non-functional, unit tests should be made for user acceptance and validation, to see whether the system meet the requirements or not. System should be user friendly, easy to use and should has user application permissions. The deployment and maintenance phase should consist of taking feedbacks to enhance better versions and requirements. For example, there should be verification of ID number and project types can be determined by the end user on application interface.

In this perspective; study wants to present an integrated waterfall methodology and OOP approach that is used during ESR DSS project life cycle management.



ESR DSS Feasibility

Feasibility Documentation	Validation	Acceptance Testing	Validation

Requirement Analysis

User Requirements	Database Requirements	OOP Analysis	Software Requirements Specification (SRS)	System Architecture Testing	Validation

System Design

Database Design	OOP Design	Integration Testing	Validation

Development

OOP Approach	Functional,Nonfunctional or Unit Testing	Validation	User Acceptance Testing	Validation

Implementation

Program use	Feedbacks	Releasing Better Versions	System Acceptance Testing	Validation	Maintenance

Figure 8: Proposed ESR DSS software life cycle with integration of Waterfall methodology and OOP approach

The software development project life cycle management framework of our study is presented in Figure 8 within OOP design strategies and Waterfall methodology. This framework integrates a hierarchical life cycle with related OOP analysis, design and approach, documentation phases, testing phases before validations. This framework will help gaining necesseary testing, reducing project time and cost, implementing a hierarchical life cycle management and releasing better versions.

5. Conclusion

The quality of a DSS depends on approprite project life cycle methodology, accurately specified requirements, objectives, software architectural design within OOP concepts and developer's competence through expected implementation. This study discussed and introduced an accurate and hierarchical DSS software life cycle framework to assist in the development of web based HRS DSSs. The intention of this research is to propose a DSS that provides necessary information and guidance to human resource systems' decision makers to be more neutral and fair in their recruitment process evaluation and to make true decision at present and in the future for their software project team.

The study tried to show importance of the development phases of a specified DSS project. The proposed system helps to solve human resource DSS problems by reducing project time and cost and assisting the managers in decision making. In addition, sequential project phases help to be consistent in development as in waterfall methodology due to it provides consistency with specified requirements analysis and system design at the beginning. The proposed multi-layered ESR DSS framework of the study divides each layer into two components in the context of security, value-added information and authentication within the decision support system. With the multi-layered structure, functional and adaptable architecture is aimed in line with the needs of the system and the users. The proposed ESR DSS software life cycle with integration of Waterfall methodology and OOP approach provides gaining necesseary testing, reducing project time and cost, implementing hierarchical life cycle managemenet and releasing better versions.

In the future, it is intended to try DSS with other project life cycle methodologies and develope new modules in a HRS DSS.

References

[1] Acock, B. Reddy, V.R. ,1997. "Designing an object-oriented structure for crop models", Ecological Modeling, 94:33–44.

[2] Lousky, D. K., 2013. "Decision Support System for Human Resource Management of the Organization", 4th IFAC Conference on Modelling and Control in Agriculture,Horticulture and Post Harvest Industry , Espoo, Finland.

[3] Urdwareshe, A.,2016. "Object-Oriented Programming and its Concepts" International Journal of Innovation and Scientific Research, Vol. 26 ,No.1,1-6.

[4] Gory, G.A., Morton, M.S.S. ,1989. "A Framework for Management Information Systems", Sloan Management Review, 49–61.

[5] Shim, J., Warkentin, M., Courtney, J., Power, D., Sharda, R., Carlsson, C., 2002. "Past, present and future of decision Support technology", Decision Support Systems 33, 111– 126.

[6] Sprague, H.R., Carlson, E.D., 1982. "Building Effective Decision Support Systems", Prentice Hall Englewood Cliffs, New Jersey.

[7] Munassar, N. M., Govardhan, A., 2011. "Comparison between traditional approach and object oriented approach in software engineering development", International Journal of the Advanced Computer Science and Applications, Volume 2 Issue 6, 70-76.

[8] Vohra, P. , Singh, A., 2013. "A Contrast and Comparison of Modern Software Process Models", International Conference on Advances in Management and Technology, Patiala, Hindistan, 23-27.

[9] Royce, W.W. ,1970. "Managing the Development of Large Software Systems: Concepts and Techniques", Proceedings of the 9th International Conference on Software Engineering, 26, 328-338.

[10] Luk, G. H., Chiu, D.K., 2006. "Web service based human resource recruitment by using matchmaking decision support" EDOCW'06, Enterprise Distributed Object Computing Conference Workshops, 1-8.

[11] Power, R. R., Sharda, R., 2007. "Model-driven decision support systems: Concepts and research directions", Decision Support Systems 43, 1044 – 1061.

[12] Detienne, F., 2007. "Design Strategies and Knowledge in Object-Oriented Programming: Effects of Experience", Human-Computer Interaction 10, 129 – 170.

[13] Veronica, R. R., 2007. "Decision Support Systems Development", Journal of the Faculty of Economics, steconomice.uoradea.ro.

[14] Samad, D.,2012. "A Decision Support System that Provides an Automated Short Listing of Potential Employees in Recruitment and Hiring of any Organization ", www.academia.edu.

[15] Chack, M., Linker, R., Teitel, M., 2013. "Development of an Object-Oriented Version of TOMGRO for a Web-based Decision Support System", Intenational Journal of Management Research and Business Strategy, , Vol. 2,No.3.

[16] Soesanto, R. P., Kurniawati, A., Ambarsari, N.,2014. "Human Resource Decision Support System Design in Maintenance Department of xxl Company Using Waterfall Method", Proceeding 7 th International Seminar on Industrial Engineering and Management.

[17] Subhan, Z., Bhatti, A.T.,2015. "Requirements Analysis and Design in the Context of Various Software Development Approaches" International Journal for Research in Applied Science & Engineering, Vol. 3 , Issue 6.

[18] Ramanathan, K.,2016. "Intelligent Decision Support Systems - A Tool for Human Resource Allocation in Information Technology Projects" International Journal of Computer Science and Information Technologies, Vol. 7 (5) , 2356-2364.

[19] Soleman, S.,2019. "Decision Support System for Employee Candidate Selection Using AHP and PM Methods", Intenational Journal of Advanced Computer Science and Applications, Vol. 10,No.11, 220-231.

[20] Martins, D. Assis, R., Coelho, R., Almeida, F., 2019. "Decision Support System for Business Ideas Competitions", Journal of Information Systems Engineering & Management, 4(3), em0093.

[21] Braude, E. J., Bernstein, M. E., 2010. "Software Engineering: Modern Approaches: An Object-Oriented Perspective", New York,Wiley&Sons,Inc., 2nd Edition.

[22] Sprague, H.R., Watson, J.H.,1993. "Decision Support Systems. Putting theory into practice", Prentice Hall International, Inc., New Jersey.

[23] R. Allen, D. Garlan, 1997. A formal basis for architectural connection, ACM Transactions on Software Engineering and Methodology 6 -3.

[24] E. Gamma, R. Helm, R. Johnson, J. Vlissides, 1995. Design Patterns: Elements of Reusable Object-Oriented Software, AddisonWesley, Reading, MA, ISBN: 0-201-63361-2.

[25] M. Shaw, R. Deline, D.V. Klein, T.L. Ross, D.M. Young, G. Zelesnik, 1996. Abstractions for software architecture and tools to support them, IEEE Transactions on Software Engineering 21- 4.

[26] S. Goddard, S. Zhang, W. Waltman, D. Lytle, S. Anthony, 2002. A software architecture for distributed geospatial decision support system, Proc. of 2002 National Conference for Digital Government Research,pp. 45 – 52.