Jurnal Kejuruteraan 34(6) 2022: 1111-1124
https://doi.org/10.17576/jkukm-2022-34(6)-12

An Effective Software Architecture of Islamic Inheritance System Employing Structured Paradigm

A. H. M. Sajedul Hoque[a], Sadia Tabassum[a], Abdullah Nazib[b], Rashed Mustafa[a] & Mohammad Osiur Rahman[a]

[a]*Dept. of Computer Science & Engineering, University of Chittagong, Chittagong, Bangladesh*
[b]*Dept. of Medical Physics, Memorial Sloan Kettering Cancer Center, USA*

**Corresponding authors : hoque.cse@cu.ac.bd, sadiatabassum64@gmail.com, abdullah.nazib@gmail.com, rashed.m@cu.ac.bd, drosi@cu.ac.bd*

Received 22 November 2021, Received in revised form 22 March 2022
Accepted 27 April 2022, Available online 30 November 2022

ABSTRACT

Distribution of wealth of a deceased person among the heirs is a critical and complicated problem in Islamic jurisprudence system. In a Muslim society, the demand of a wealth calculation system is very high and plays a pivotal role in day-to-day life of Muslim families. In this paper, we propose a structured and well-defined software solution that employs structured software engineering approach using Data Flow Diagram (DFD), structure chart and program description language (PDL). The proposed system distributes shares among the heirs based on Islamic rules. Many such systems have been built by different govt agencies of different Muslim majority countries. Many of those systems are error prone and are not well defined from the software engineering perspective. In many cases, the software architecture of such systems is undisclosed and their applicability in many complex deceased-heir relationship provides wrong calculation. The proposed solution on the other hand, makes the successful use of software engineering approach to interpret Islamic Inheritance rules in a more manageable, scalable and maintainable fashion. The system is rigorously tested on different complex scenarios and the results are verified by Muslim family lawyers. The system is developed using C programming language and we make the source code open (https://github.com/sajidcsecu/IIS).

Keywords: Structured analysis; structured design; DFD; structure chart; Islamic inheritance system

INTRODUCTION

Precise distribution of assets of deceased among his or her heirs is a crucial task in the society across the world in order to ensure the law and order. Computing property shares among the heirs vary from religion to religion or culture to culture or country to country (Yusuf 2017)the ownership or transfer of property rights in Islam is clearly regulated through inheritance, sale, gifts, grants, endowments, alms, and other lawful means, such as loans and mortgages. The ownership or transfer of property through inheritance is an important part of Islam. Inheritance relationship between offspring was not easily done, both based on the particular culture and religion. Among the Hindus, especially in Bali, girls do not receive inheritance. The same also applies to Western society in England some time ago. In Padang-Muslim society, men do not receive it. In Javanese society, inheritance is divided equally, without differentiating boys and girls. Such inheritance is based on the cultural standards and anthropocentric paradigm (man as the center of everything. In order to ease the computations, a software system called Islamic Inheritance System (IIS) is felt to develop and that system should be acceptable by the law of that land/culture/religion, technically maintainable and computationally efficient. In order to get the professional system, systematic software engineering approach should be taken into consideration to define the problem, analyze and design software architecture, and testing of the system (MATHA 2008). The main focus of the paper is to analyze, design and implement an inheritance system for Islamic Laws. In software engineering, the main architectural design and detail design of any professional system is found after design phase based on the artifacts found from analysis phase. Although many software solutions for Islamic Inheritance System (IIS) are already available, their architectural design and detail design are not published. So, the number of modules and their inter-dependency are not known. Consequently, the level of cohesiveness and coupling of responsible modules of the system cannot be determined. The cohesiveness and coupling of modules are two key features to measure the quality of IIS software product. High cohesive and loose coupled IIS software solution is always expected in software engineering community, because these increase the maintainability of the software (Stevens, Myers and Constantine 1999). For this reason, there is considerable doubt regarding those developed IISs that whether those

systems maintain software engineering approach at all. S. Tabassum et al. worked on IIS employing structured software engineering paradigm where Data Flow Diagrams (DFDs) were developed the analysis phase and the artifacts of design phase were completely avoided (Tabassum et al. 2019). That is, the structure chart and the data structure and algorithms of each module were not mentioned. Without those artifacts, the developed system cannot be maintainable and this is first target problem of this work. The second problem is S. Tabassum did not consider offspring for measuring eligibility of Father and Grandfather as prescribed shares. Finally, the mathematical model of Radd case developed by S. Tabassum increases the portion of spouse proportionately which is the violation Islamic rules. Based on these problems, this paper has three contributions. Firstly, an architectural design employing structure chart is developed for IIS. Secondly, the mathematical models for Father, Grandfather and Radd case are corrected according to the Islamic laws. Finally, algorithms and data structure for all modules of developed structure chart are designed using the developed mathematical models. The paper is organized into five sections. Section 2 describes the related published works on the Inheritance system. The thorough description of the proposed software architecture is discussed in section 3. Section 4 presents an experimental result with comprehensive analysis of the work. Finally, the paper is concluded mentioning some future works in section 5.

RELATED WORKS

Since 1960s, software is evolving and they are becoming part of our daily activities from professional level to personal level. Until 1975, there were no specific paradigm to develop a software for those activities in a systematic, reproducible and highly productive manner. A very powerful approach called structured paradigm was developed in between 1975 and 1985 (Schach 2010). In 2015, H. Alshahad et al first proposed an online Islamic Inheritance system (IIS). The authors gave an idea for calculating the shares of heirs where the system would take all necessary information about the deceased and the information were moved to a decision table including all rules of Islamic law for determining the portions (Alshahad and Abutiheen 2015). Next, Alaa N. Akkila et al. (2016) proposed an expert system for the computation consisting of 43 rules on Islamic law employing CLIPS language (Alaa N. Akkila and Samy S. Abu Naser, 2015). An android application of inheritance system was designed and developed by A. Zulkifli et al. in 2018 employing Rapid Application Development (RAD) model (Zulkifli, Batiha and Qasim, 2018). Then, S. Zouaoui et al. in 2018 provided a software architecture of Islamic inheritance system named AraFamOnto which was based on the family ontology representing the family relationships (Zouaoui and Rezeg 2018). It is obligatory to formulate mathematical expressions of heirs for the system in order to determine the precise portions. In 2017, K. Babalola presented all mathematical expressions for computing the desired shares of live heirs of deceased (Babalola 2017). The downside of these expressions was that those were not well-expressed enough for building system. Tabassum S. et al. proposed a set of all necessary well-expressed equations where related conditions were amalgamated through one or more Boolean functions for easing to construct the software system (Tabassum et al. 2019). Among those equations, the mathematical models for father, grandfather and adjustment are error prone in some situations. Thus, the first objective of this work is to correct those error-prone equations. In addition, an inheritance calculator named "Division of Inheritance" was developed by Dr. Halis Aydemir and the generated results are shown in percentage, fraction and bar chart (Aydemir, no date). Another calculator entitled ShariahStandards.org allocates the shares of legal heirs in percentage and fractions and the allocated shares are depicted in a circle through diverse shades of colors (ShariahStandards.org, no date). The demerits of both systems are that both systems are capable of computing shares for father, mother, sons, daughters, spouse, brothers and sisters exclusively and could not differentiate among full, paternal and uterine brothers and sisters. Moreover, the next inheritance calculator hosted on "lubnaa.com" computes the shares of legitimate heirs from nine heirs in fraction. The front page of the system is confusing and it produces confounding results. For instance, paternal and maternal brothers and sisters are mentioned, but there is no clear idea about full sisters and brothers (Lubnaa - Inheritance Calculator, no date). Furthermore, the office of the prime minister of Bangladesh launched a web application named Uttaradikhar under the Access to information (a2i) project. The system is based on the amended Islamic laws of Bangladesh constitution (Uttaradhikar 2016). Over and above, the other renowned web application (Islamic Inheritance Calculator) was built by Najeeb integrating all Islamic laws considering 29 legal heirs over five generations of heirs (Najeeb 2014). Finally, the most sophisticated inheritance calculator was developed by Muhammad Waqas incorporating all possible constraints of Islamic inheritance laws (Waqas 2017). Although all aforementioned systems are freely usable, their related internal software architectures are undisclosed. As the system components and their relationships are unknown, it is hard to determine their degree of cohesiveness and coupling which are related to the maintainability of those systems. Based on our analysis and observation, the reliability of those systems is in question and we are not certain whether or not they followed software engineering approach for implementation from the architectural point of view. Tabassum S. et al. worked on the software architecture of the inheritance system in her article (Tabassum et al. 2019) employing structured software engineering approach. Although the article gave level 0, 1 and 2 Data Flow Diagrams (DFDs) for structured analysis phase, the architectural design and detail design for structured design phase were not specified. Because both designs were not constructed, the developed DFDs were not

verified whether it is ok or not. Thus, the final targets of this proposed research work are to revise the proposed DFDs by Tabassum et al. develop the architectural and detailed design and implement those designs using structured programming language for justification.

METHODOLOGY

Developing the blueprint of IIS utilizing the Structured System Analysis and Design (SSAD) is the key task of the research work. In SSAD, proposed by Tom DeMarco, Edward Yourdon and Larry Constantine, the functional requirements are transformed into analysis specification and then into design specification employing functional decomposition technique. Here structured analysis called data-oriented approach employs DFD to depict flow of data among decomposed processes, while structure charts are used for representing the hierarchy of procedures over the DFD in structured design known as control-oriented technique (Sinha, 2019). According to the concept of SSAD, the procedure of the work consists of several steps. Firstly, the functional requirements of IIS are determined and suitable scenarios of those functional requirements are modeled analyzing the problem domain. After that, the scenarios are analyzed and specified through the Data Flow Diagrams. Then, the highest level DFD is transformed into structure chart as an architectural design. Furthermore, the detail design of each module in structure chart is modelled using Program Description Language (PDL). Finally, the detail designs are used for building the desired IIS. The whole procedure is explained step by step in detail in the following subsections.

STRUCTURED ANALYSIS

In structured analysis, the domain of the IIS is analyzed for identifying core functional requirements with the related scenarios. Then, the identified scenarios are specified through level 0 DFD to higher level DFD.

FUNCTIONAL REQUIREMENT IDENTIFICATION

The main use case of the IIS is to compute the portions of shares of deceased among the eligible heirs. Since the distribution process is based on Islamic law, the overall scenario for computing portions follow the holy Quran and Hadith (Adelina Zuleika, 2013) (Muhammad 2020) and is illustrated in Figure 1.

1. The system reads and validates the details of deceased including TES and list of live heirs.
2. The system computes the prescribed shares of all eligible prescribed heirs using TES and list of live heirs.
3. The system computes the residual shares of all eligible residual heirs TES and list of distributed prescribed shares.
4. The system sums the individual shares using the list of distributed prescribed and residual shares.
5. The system adjusts the total individual shares under TES.
6. The system displays the adjusted individual shares.

FIGURE 1. Scenario of IIS

At the beginning of the process, the *List of Valid Heirs (LVH) and Total Effective Shares (TES)* are taken and validated where *TES* for allotting is computed by deducing his or her loan, will and funeral costs from the shares left behind. Then, the *TES* is distributed among the prescribed heirs who are specified with their deserved portion in the Holy Quran. The eligible possible prescribed heirs are *spouse* (*Husband* or *Wives*), *Daughter, Father, Mother, Grandfather, Paternal Granddaughter, Sister(full), Paternal Grandmother and Maternal Grandmother*. Among those relatives, *Spouse* and *Mother* are confirmed heirs. That is, if they are alive, they must get shares. The rest of the prescribed relatives are conditional. For example, *Daughter* will be considered as prescribed heir, if the deceased has no *Son*. The preconditions of eligibility of heirs in prescribed category and the corresponding computational model for computing the portions are shown in Table I. Every computational model consists of one or more predicates which return integer value 1(one) whenever those are true. However, The models for precondition of *Father* and *Grandfather* proposed by S. Tabassum (Tabassum et al. 2019) have not been covered completely. The modified models are shown in Table I where if deceased has offspring, the *Father* will be eligible for prescribed shares. Similarly, the condition for *Grandfather* is not to have *Father* and *offspring*. However, the prescribed shares for all non-eligible prescribed heirs in *LVH* are assigned as zero. After that, if the *Total Allotted Prescribe Shares (TAPS)* is greater than *TES*, the residual share for each heir in *LVH* will be assigned as zero. Otherwise, the residual heirs are determined using the matching conditions illustrated in Table II. If residual heirs are not available, again all residual shares for all heirs in *LVH* are given as zero. If not, the deserved residual shares are computed and the computational models for residual

shares are straightforward where the ratio of male and female candidates is 2:1. If the number of male and female candidates are *NM* and *NF* respectively, the portion of *Male Residual Shares (MRS)* and *Female Residual Shares (FRS)* are computed as equation 1 and 2 (Tabassum et al. 2019).

TABLE 1. Summary of Prescribed Shares

No	Heirs	Preconditions	Computational Model	Acronyms
1	Husband	Confirmed	$\frac{\text{TES}}{4}(1+\text{hNC}())$	
2	Wives	Confirmed	$\frac{\frac{\text{TES}}{8}(1+\text{hNC}())}{\text{NW}}$	1. hCN(): hasNoChildren()
3	Daughters	Has no son	$\frac{\frac{\text{TES}}{2}+\frac{\text{TES}}{6}(1+\text{hSD}())}{\text{ND}}$	2. NW: Number of Wives
4	Father	Has Offspring	$\frac{\text{TES}}{6}$	3. hSD(): hasSingleDaughter()
5	Mother	Confirmed	$\frac{\text{TES}}{6}(1+(\text{hNC}()\ \text{AND}\ \text{hNMS}()))$	4. ND: Number of Daughters
6	Paternal Grandfather	Has no Father and Offspring	$\frac{\text{TES}}{6}$	5. hNMS(): hasNoMultipleSiblings()
7	Paternal Grandmother	Deceased has no mother and father	$\frac{\text{TES}}{12}(1+\text{hNMGM}())$	6. hNMGM(): hasNoMaternalGrandmother()
8	Maternal Grandmother	Has no mother	$\frac{\text{TES}}{12}(1+(1-(\text{hNF}()\ \text{AND}\ \text{hPGM}())))$	7. hNF(): hasNoFather()
9	Paternal Granddaughters	Has no son, grandson and at most one daughter	$\frac{\frac{\text{TES}}{6}\text{hSD}()+\frac{\text{TES}}{2}(1-\text{hSD}())+\frac{\text{TES}}{6}(1-\text{hSGD}())}{\text{NGD}}$	8. hPGM(): hasPaternalGrandmother()
				9. NGD: Number of Granddaughter
				10. hSGD(): hasSingleGranddaughter()
				11. hSS(): hasSingleSister()
				12. NS: Number of Sisters
10	Sisters(full)	Has no daughter, son, paternal grandson, paternal granddaughter, father, grandfather, brother	$\frac{\frac{\text{TES}}{2}+\frac{\text{TES}}{6}(1-\text{hSS}())}{\text{NS}}$	

So far, each heir in *LVH* has both prescribed and residual shares which are summed individually. Finally, the summation of individual earned share represented as *TIS* may be either equal to or less than or greater than *TES*. If *TIS* is equal to *TES*, the computed shares for each survivor till now are final. Otherwise, the computed shares are adjusted under *TES*. For this reason, the *Remaining Share (RS)* is calculated using the equation 3. If *RS* is greater than zero, then the individual earned share is proportionately increased except *Spouse*, whereas S. Tabassum considered *Spouse* in her model (Tabassum et al. 2019). This situation is called Radd case (Najeeb, 2014). The *TIS* is updated by deducting the earned shares of *Spouse*. However, if RS is less than zero known as Awal case (Najeeb 2014), the individual earned shares are proportionately decreased. The *Adjusted Shares(AS)* of *heir* in the list *LVH* is determined using the equation 4. Here the function *earnedShare(heir)* returns the earned individual shares of *heir* and *isSpouse(heir)* and *isPositive(RS)* are two predicates. The former predicate returns one if the *heir* is spouse, while the later predicate gives one for being positive value of *RS*. Otherwise, each predicate provides zero. It has seen that when both predicates are 1, which indicates the heir is spouse and the sum of total distributed shares is less than *TES*, the shares will not be increased. Otherwise, it will be increased.

TABLE 2. Summary of Residual Heirs

No	Heirs	Preconditions
1	Son	Confirmed
2	Daughter	Has son
3	Paternal Grandson	Has no son
4	Paternal Granddaughter	Has no Son AND Has paternal Grandson
5	Father	Has no (Son AND Grandson)
6	Grandfather	Has no (Son AND Grandson AND Father)
7	Brother	Has no (Son AND Grandson AND Father AN Grandfather)
8	Sister	Has no (Son AND Grandson AND Father AND Grandfather) AND Has (Brother OR Daughter OR Granddaughter)
9	Brother's Son	Has no (Son AND Grandson AND Father AND Grandfather AND Brother AND Daughter AND Granddaughter)
10	Son of brother's son	Has no (Son AND Grandson AND Father AND Grandfather AND Brother AND Daughter AND Granddaughter AND Brother's Son)
11	Paternal Uncle	Has no (Son AND Grandson AND Father AND Grandfather AND Brother AND Daughter AND Granddaughter AND Brother's Son AND Son of Brother's Son)
12	Paternal Cousin	Has no (Son AND Grandson AND Father AND Grandfather AND Brother AND Daughter AND Granddaughter AND Brother's Son AND Son of Brother's Son AND Paternal Uncle)

$$\mathrm{MRS}=2\ \frac{\mathrm{TES\text{-}TAPS}}{2\ \mathrm{NM}+\mathrm{NF}} \tag{1}$$

$$\mathrm{FRS}=\frac{\mathrm{TES\text{-}TAPS}}{2\ \mathrm{NM}+\mathrm{NF}} \tag{2}$$

$$\mathrm{RS}=\mathrm{TES\text{-}TIS} \tag{3}$$

$$\mathrm{AS(heir)}=\mathrm{earnedShare(heir)}\left(1+1\text{-}\mathrm{isSpouse(heir)}\ \mathrm{isPositive(RS)}\right)\ \frac{\mathrm{RS}}{\mathrm{TIS}} \tag{4}$$

FUNCTIONAL REQUIREMENT SPECIFICATION

Having fixed the scenario, DFD is used for modeling the decomposition of function into sub-functions. According to the SSAD, first of all, the context model of the IIS is constructed using level 0 DFD illustrated in Figure 2 where the big bubble represents the whole system. Then, as the scenario in Figure 1 says one use case, the system has one process called "Compute Shares", the derived level 1 DFD from level 0 DFD has only one data processing unit shown in Figure 3. Next, the "Compute Shares" functional unit is decomposed into six sub-functional units including reading details of deceased, computing prescribed shares, computing residual shares, summing individual shares, adjusting the shares and displaying the earned shares depicted in Figure 4 as level 2 DFD. Firstly, the "Read Details of Deceased" unit takes and validates the details of deceased comprising list of heirs, amount of property left behind. After that, "Compute Prescribed Share" sub-functional unit determines the amount of shares for eligible prescribed heirs. Next, the number of shares for eligible residual heirs is computed by the "Compute Residual Shares" sub unit. Then, the distributed shares for prescribed heirs and residual heirs are summed and adjusted under the total effective shares individually employing the "Sum Individual Shares" and "Adjust Individual Shares" sub-functional units respectively. Finally, "Display Individual Shares" unit shows the produced results to the user. Through building the level 2 DFD, the structured analysis is over and the DFD is ready to architect.

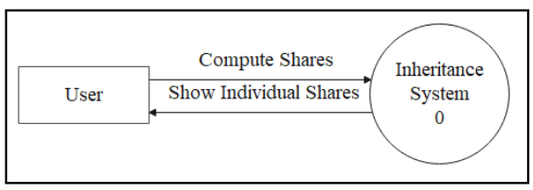

FIGURE 2. Level 0 DFD

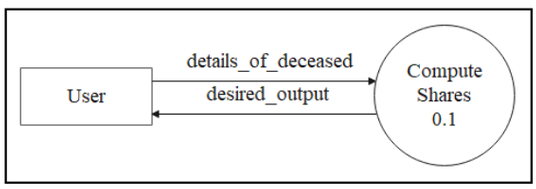

FIGURE 3. Level 1 DFD

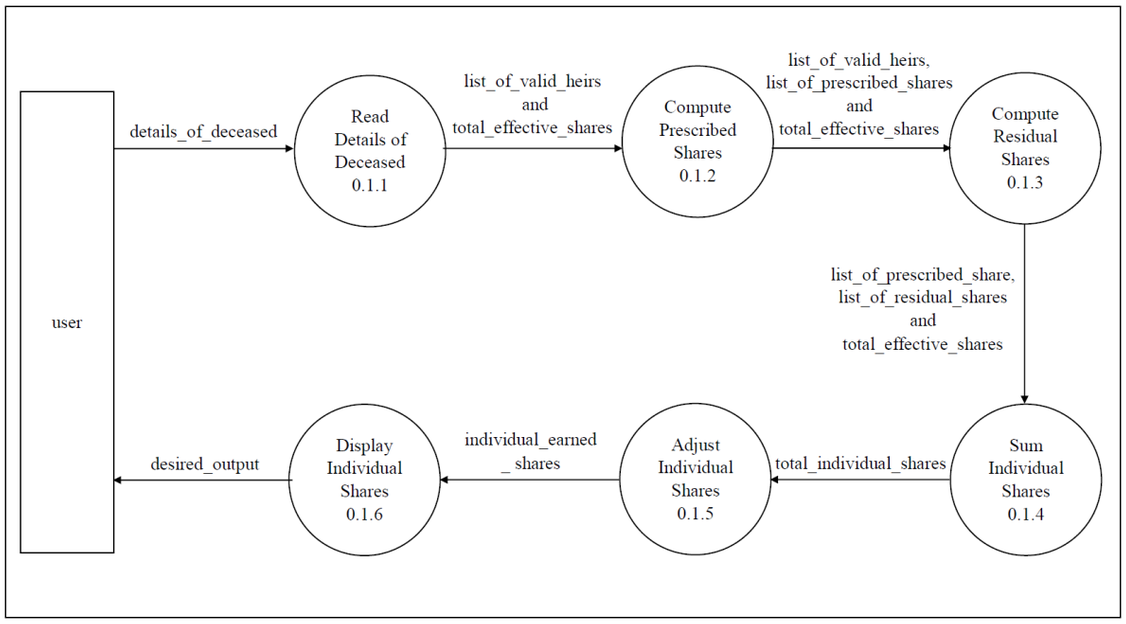

FIGURE 4. Level 2 DFD

STRUCTURED DESIGN

In structured design, the highest level of DFD of a system is transformed into two designs: architectural design and detail design. Architectural design possesses the modules of the whole system and their internal connections. And the structure chart is used to show the architectural design. Then, in detail design, the algorithm and data structure of each module are selected. Now both designs of IIS are explaining below.

ARCHITECTURAL DESIGN

The input of this phase is the highest level of DFD in Figure 4 and the output artifact is structure chart. The final structure chart is determined through two refinements. In the first refinement shown in Figure 5, the sub-functions of level 2 DFD are divided into three modules: input, output and logical processing. In this work, "Read Details of Deceased" and *"Display Individual Shares"* sub-functional units of level 2 DFD act as input module named *read_ details_ of_ deceased* and output module named *display_ individual_shares* respectively and the rest of the sub-functions are the part of logical processing module named *compute_shares*. These three modules are invoked from the *main* module.

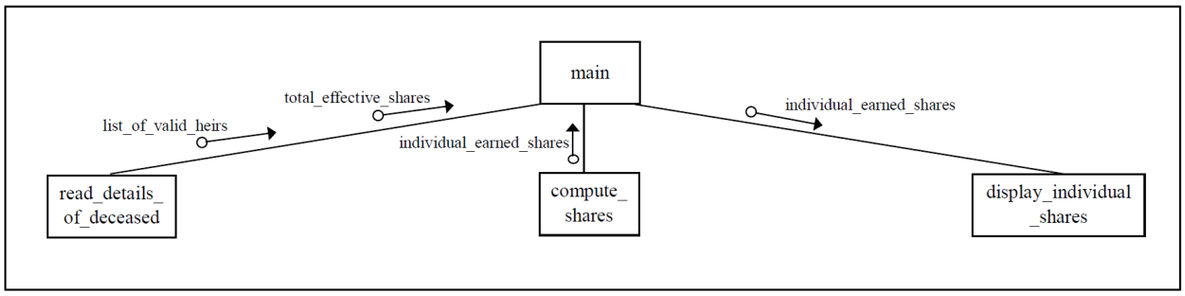

FIGURE 5. First Refinement of Structure Chart

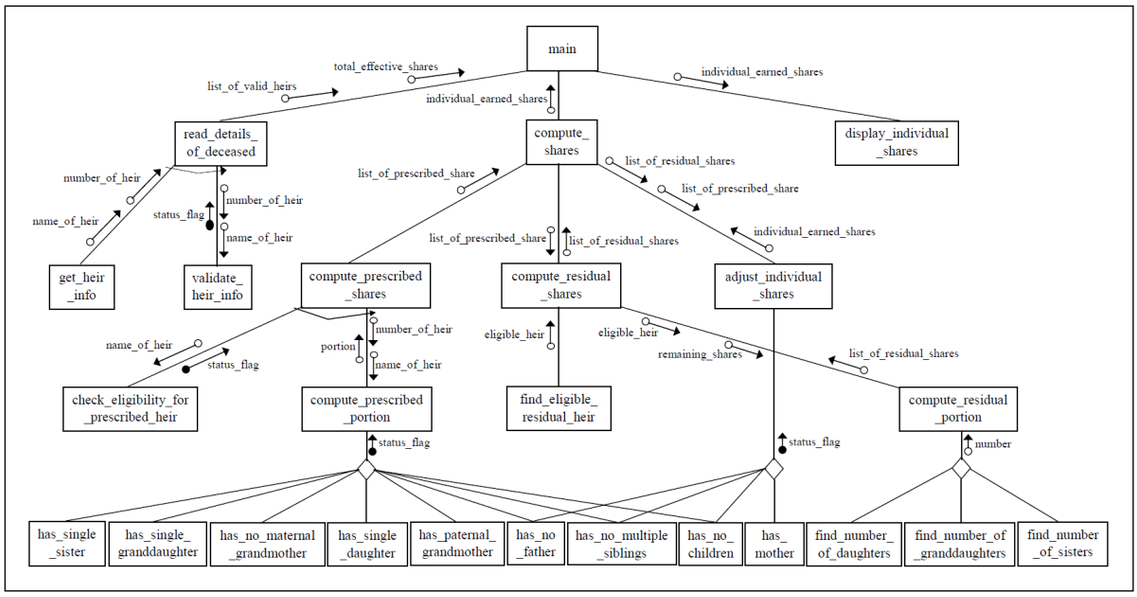

FIGURE 6. Second Refinement of Structure Chart

Although the *main* and *display_individual_shares* are cohesive modules, the rest of other modules are not. Because, the *read_details_of_deceased* module will perform two tasks: get and validate the information of one heir and the *compute_shares* consist of three basic tasks including computing prescribed shares, computing residual shares and adjusting individual shares. Thus, the structure chart in Figure 5 is further refined and shown in Figure 6. The module *compute_prescribed_shares* is divided into modules: *check_eligibility_for_prescribed_heir* for checking eligibility of each heir and *compute_prescribed_portion* for computing portion of eligible heir. Similarly, the *compute_residual_shares* module is decomposed into finding eligible residual heir and computing the desired residual portion done by the *find_eligible_residual_heir* and *compute_residual_portion* modules respectively. The last level of structure chart has 12 small modules to assist the *compute_prescribed_portion, compute_residual_portion* and *adjust_individual_shares*. The modules are: *has_no_children, has_no_multiple_siblings, has_mother, has_no_maternal_grandmother, has_no_father, has_paternal_grandmother, has_single_daughter, has_single_sister, has_single_granddaughter, find_number_of_daughters, find_number_of_granddaughters* and *find_number_of_sisters.*

Algorithm 1 The algorithm of `check_eligibility_for_prescribed_heir`

Input : `name_of_heir`
Output : `status_flag`

1: **if** `name_of_heir` is Husband OR Wives OR Mother **then**
2: `status_flag` ← 1
3: **else if** `name_of_heir` is Daughters AND Sons are not in `list_of_valid_heirs` **then**
4: `status_flag` ← 1
5: **else if** `name_of_heir` is Father AND (Sons OR Grandsons OR Daughters OR Granddaughters) are in `list_of_valid_heirs` **then**
6: `status_flag` ← 1
7: **else if** `name_of_heir` is Paternal Grandfather AND Father is not AND (Sons OR Paternal Grandsons OR Daughters OR Paternal Granddaughters) are in `list_of_valid_heirs` **then**
8: `status_flag` ← 1
9: **else if** `name_of_heir` is Paternal Grandmother AND (Father AND Mother) are not in `list_of_valid_heirs` **then**
10: `status_flag` ← 1
11: **else if** `name_of_heir` is Maternal Grandmother AND Mother is not in `list_of_valid_heirs` **then**
12: `status_flag` ← 1
13: **else if** `name_of_heir` is Paternal Granddaughters AND (Sons AND Paternal Grandsons) are not in `list_of_valid_heirs` AND number of Daughters is at most 1 **then**
14: `status_flag` ← 1
15: **else if** `name_of_heir` is Sisters AND (Sons AND Paternal Grandsons AND Daughters AND Paternal Granddaughters AND Father AND Grandfather AND Brothers) are not in `list_of_valid_heirs` **then**
16: `status_flag` ← 1
17: **else**
18: `status_flag` ← 0
19: **end if**

DETAILED DESIGN

From the structure chart, any software engineer can get idea about the modules of the target system. But this is not enough for coder to build the working system due to the lack of business logic of each module. Thus, the data structure and the algorithm of each module should be designed and this is called detailed design which is represented in Program Description Language (PDL) in this work. According to the architectural design of the system illustrated in Figure 6, there are 25 modules where a linked list named *heir* is used to maintain the information of heirs of deceased integrating *name*, *number, shares* and the address of next node, *next*. In this proposed architectural design, one heir type linked list named *list_of_valid_heirs* and one real type variable, *total_effective_shares*, are used as global variables. Due to the conciseness of the article, the detailed design of only core five out of twenty-five modules is illustrated. According to the business process, the *list_of_valid_heirs* and *total_effective_shares* are provided by using *read_details_of_deceased*. Then, the *compute_prescribed_shares* module sends each heir of the list to and receives a status flag indicating eligibility from the check_*eligibility_for_prescribed_heir* (Algorithm 1) module. If the status is ok, then the *compute_prescribed_portion* (Algorithm 2) takes the number of heirs to compute the portion of prescribed shares which is also used by *compute_prescribed_shares* to produce the *list_of_prescribed_shares*. Then, the *list_of_prescribed_shares* is passed to the module *compute_residual_shares* which finds the eligible residual heir by *calling find_eligible_residual_heir* and sends the heir to the *compute_residual_portion* to make another list named *list_of_residual_shares*. The business logic of *find_eligible_residual_heir* in Algorithm 3 and *compute_residual_portion* in Algorithm 4 is based on the Table 2, equation 1 and 2. After that, the prescribed portion and residual portion for each heir are integrated inside the Algorithm 5 named *adjust_individual_shares*. This algorithm takes *list_of_prescribed_shares* and *list_of_residual_shares* as inputs and produces *individual_earned_shares* as output. Inside the algorithm, firstly, the prescribed and residual portion of each heir are summed. Then, if the special *Umar* case (*spouse, mother* and *father*) (Najeeb, 2014) is in the list, *individual_earned_shares*, the distributed shares are adjusted by re-allotting the shares of father and mother using the ratio 2:1. Otherwise, the distributed shares are either proportionately increased or decreased considering both the *Radd* and *Awal* case (Najeeb, 2014) employing equation 3 and 4. If the distributed shares among all eligible heirs are exactly equal to the *TES*, then the shares are multiplied by 1 for making sure unchanged. This constraint is also included in the detailed design of *adjust_individual_shares*.

Algorithm 2 The algorithm of `compute_prescribed_portion` module

Input : `name_of_heir`,`number_of_heir`
Output : portion

1: **if** `name_of_heir` is Husband **then**
2: portion $\leftarrow \frac{\texttt{total\_effective\_shares}}{4} \cdot (1 + \texttt{has\_no\_children()})$
3: **else if** `name_of_heir` is Wives **then**
4: portion $\leftarrow \frac{\texttt{total\_effective\_shares}}{8} \cdot \frac{(1+\texttt{has\_no\_children()})}{\texttt{number\_of\_heir}}$
5: **else if** `name_of_heir` is Daughters **then**
6: portion $\leftarrow \frac{\texttt{total\_effective\_shares} \cdot (\frac{1}{2}+\frac{1}{8} \cdot (1-\texttt{has\_single\_daughter()}))}{\texttt{number\_of\_heir}}$
7: **else if** `name_of_heir` is Father **then**
8: portion $\leftarrow \frac{\texttt{total\_effective\_shares}}{6}$
9: **else if** `name_of_heir` is Mother **then**
10: portion $\leftarrow \frac{\texttt{total\_effective\_shares}}{6} \cdot (1 +$ (`has_no_children()` AND `has_no_multiple_siblings()`))
11: **else if** `name_of_heir` is Paternal Grandfather **then**
12: portion $\leftarrow \frac{\texttt{total\_effective\_shares}}{6}$
13: **else if** `name_of_heir` is Paternal Grandmother **then**
14: portion $\leftarrow \frac{\texttt{total\_effective\_shares}}{12} \cdot (1 +$ `has_ho_maternal_grandmother()`)
15: **else if** `name_of_heir` is Maternal Grandmother **then**
16: portion $\leftarrow \frac{\texttt{total\_effective\_shares}}{12} \cdot (1 + (1 -$ (`has_no_father()` AND `has_paternal_grandmother()`)))
17: **else if** `name_of_heir` is Paternal Granddaughters **then**
18: portion $\leftarrow$ `total_effective_shares` $\cdot$ $(\frac{\frac{1}{6} \cdot \texttt{has\_single\_daughter()}+\frac{1}{2} \cdot (1-\texttt{has\_single\_daughter()})+\frac{1}{6} \cdot (1-\texttt{has\_single\_granddaughter()})}{\texttt{number\_of\_heir}})$
19: **else if** `name_of_heir` is Sisters **then**
20: portion $\leftarrow \texttt{total\_effective\_shares} \cdot (\frac{\frac{1}{2}+\frac{1}{6} \cdot (1-\texttt{has\_single\_sister()})}{\texttt{number\_of\_heir}})$
21: **else**
22: portion $\leftarrow 0$
23: **end if**

RESULT AND DISCUSSION

The detailed design of the IIS is ready for implementing. In this work, the system has been built using C language through Code::Blocks IDE. The output screen of the module *read_details_of_deceased* and *display_individual_shares* are illustrated in Figure 7 and Figure 8 respectively. In order to evaluate the proposed system, it is necessary to test using recognized propositions. Chemma (Cheema 2021) mentioned twenty propositions in his article including all possible heirs. Since our work developed the models for full type heirs like full sisters, only twelve propositions out of twenty are used for system verification shown in Table III. The table consists of different cases with diverse perspective of Islamic Inheritance law. The first three cases are associated with both prescribed and residual shares. Case no 4 and 5 deal with the rule of *Awal* and *Hajb* (exclusion), whereas case 6 and 7 handle the principle of *Radd*. Finally, the proposition no 8 and 9 follow the Umariyyatan, while the proposition no 12 embodies the law of *Al-Akdariyya* (Cheema 2021). It is assumed that the worth of total effective shares left behind by deceased is 100 and has been seen that our proposed system produces exactly same result as all 12 selected propositions. Therefore, the experimental result in Table III states the proposed system is reliable. Moreover, every module in structure chart in Figure 6 has unique functionality. For example, *check_eligibility_for_prescribed_heir* module consists only the business logic for checking the eligibility of an heir which is completely separated from the module *compute_prescribed_portion* for computing prescribed portion. So, the architectural design of the proposed system is functionally cohesive that indicates highly cohesiveness of that system.

Algorithm 3 The algorithm of `find_eligible_residual_heir` module

```
Output :eligible_heir
if Sons are in list_of_valid_heir then
    eligible_heir ← "Sons"
else if Paternal Grandsons are in list_of_valid_heir then
    eligible_heir ← "Paternal Grandsons"
else if Father is in list_of_valid_heir then
    eligible_heir ← "Father"
else if Paternal Grandfather is in list_of_valid_heir then
    eligible_heir ← "Paternal Grandfather"
else if Brothers is in list_of_valid_heir then
    eligible_heir ← "Brothers"
else if Daughters are in list_of_valid_heir AND
find_number_of_sister() > 0 then
    eligible_heir ← "Daughters"
else if Paternal Granddaughters are in list_of_valid_heir AND
find_number_of_sister() > 0 then
    eligible_heir ← "Paternal Granddaughters"
else if Brother's Sons are in list_of_valid_heir then
    eligible_heir ← "Brother's Sons"
else if Sons of Brother's Sons are in list_of_valid_heir then
    eligible_heir ← "Sons Brother's Sons"
else if Father's Brothers are in list_of_valid_heir then
    eligible_heir ← "Father's Brothers"
else if Sons of Father's Brothers is in list_of_valid_heir then
    eligible_heir ← "Sons of Father's Brothers"
else
    eligible_heir ← 0
end if
```

Next, the architectural design is examined and evaluated in terms of the types of connections between modules. The strength of those connections is called coupling. In the proposed architectural design, there are four types of couplings: content coupling, common coupling, control coupling and stamp coupling (Stevens, Myers and Constantine, 1999). Firstly, it has been seen that if the content of *get_heir_info module* is changed, 16 different modules including all modules in level 4 and 3 of structure chart in Figure 6, *validate_heir_info* and *adjust_individual_shares* modules must be changed. Moreover, there are fifteen and five common couplings for *list_of_valid_heirs* and *total_*

effective_shares global variables respectively according to the detail design of the proposed system. In addition, four control couplings are in *read_details_of_deceased, compute_prescribed_shares, compute_prescribed_portion* and *adjust_individual_shares* modules due to the presence of *status_flag*. Finally, *list_of_prescribed_shares, list_of_residual_shares* and *individual_earned_shares* ensure three stamp couplings among *compute_prescribed_shares, compute_residual_shares, adjust_individual_shares display_individual_shares* modules. Similarly, another three stamp couplings are observed among *get_heir_info*, *validate_heir_info, check_eligibility_for_prescribed_heir, compute_prescribed_portion, find_eligible_residual_heir* and *compute_residual_portion*. Although the proposed architectural design is not absolute loose coupled, its high cohesiveness increases the maintainability of the system. Therefore, any required change can be implemented quickly and easily without changing the basic software architecture of proposed system which is the main strength of this work.

Algorithm 4 The algorithm of `compute_residual_portion` module

Input : `heir, remaining_shares`
Output : `list_of_residual_shares`

1: `list_of_residual_shares` ← `list_of_valid_heirs`
2: **if** `heir` is "Daughters" OR "Paternal Granddaughters" **then**
3: `number_of_male_heirs` ← 0
4: **else**
5: Set **male** to the node named **heir** in `list_of_residual_shares`
6: `number_of_male_heirs` ← male.number
7: **end if**
8: **if** `heir` is **"Sons" then**
9: `number_of_female_heirs` ← `find_number_of_daughters()`
10: **if** (`number_of_female_heirs` > 0) **then**
11: Set **female** to the node named **"Daughters"** in `list_of_residual_shares`
12: **end if**
13: **else if** `heir` is **"Paternal Grandsons" then**
14: `number_of_female_heirs` ← `find_number_of_granddaughters()`
15: **if** (`number_of_female_heirs` > 0) **then**
16: Set **female** to the node named **"Granddaughters"** in `list_of_residual_shares`
17: **end if**
18: **else if** `heir` is **"Brothers"** OR **"Daughters"** OR **"Granddaughters" then**
19: `number_of_female_heirs` ← `find_number_of_sisters()`
20: **if** (`number_of_female_heirs` > 0) **then**
21: Set **female** to the node named **"Sisters"** in `list_of_residual_shares`
22: **end if**
23: **else**
24: `number_of_female_heirs` ← 0
25: **end if**
26: **if** (`number_of_male_heirs` > 0 AND `number_of_female_heirs` = 0) **then**
27: male.shares ← $\frac{\texttt{remaining\_shares}}{\text{male.number}}$
28: **else if** (`number_of_male_heirs` = 0 AND `number_of_female_heirs` > 0) **then**
29: female.shares ← $\frac{\texttt{remaining\_shares}}{\text{female.number}}$
30: **else**
31: **if** (`number_of_male_heirs` ≠ 0 AND `number_of_female_heirs` ≠ 0) **then**
32: female.shares ← $\frac{\texttt{remaining\_shares}}{2 \cdot \texttt{number\_of\_male\_heirs} + \texttt{number\_of\_female\_heirs}}$
33: male.shares ← 2 · female.shares
34: **end if**
35: **end if**

Algorithm 5 The algorithm of `adjust_individual_shares` module

Input : `list_of_prescribed_shares`, `list_of_residual_shares`
Output : `individual_earned_shares`

```
t ← list_of_prescribed_shares
individual_earned_shares ← t
while (t ≠ NULL) do
    temp ← list_of_residual_heirs
    while (temp ≠ NULL) do
        if (t.name = temp.name) then
            t.shares ← t.shares + temp.shares
        end if
        t ← t.next
    end while
end while
t ← individual_earned_shares
while (t ≠ NULL) do
    if (t.name is Husband OR Wives) then
        spouse_shares ← t.shares
    end if
    sum ← sum + (t.shares · t.number)
    t ← t.next
end while
remaining_shares ← total_effective_shares − sum
if (spouse_shares > 0 AND (1 − has_no_father()) = 1 AND has_mother() = 1 AND has__no_children() = 1 AND has_no_multiple_siblings() = 1) then
    t ← individual_earned_shares
    Set temp to the node named "Father" in t
    temp.share ← 2 · (sum − spouse_shares) / 3
    t ← individual_earned_shares
    Set temp to the node named "Mother" in t
    temp.share ← (sum − spouse_shares) / 3
else
    if (remaining_shares > 0) then
        sum ← sum − spouse_shares
    end if
    adjustment_factor ← remaining_shares / sum
    t ← individual_earned_shares
    while (t ≠ NULL) do
        spouse_flag ← 1
        if ((t.name is Husband OR Wives) AND (remaining_share > 0)) then
            spouse_flag ← 0
        end if
        t.shares ← t.shares · (1 + (adjustment_factor · spouse_flag))
        t ← t.next
    end while
end if
```

```
        *****ISLAMIC INHERITANCE CALCULATOR *****

        *****List of Heirs *****

1. Husband                              2. Wives

3. Daughters                            4. Sons

5. Father                               6. Mother

7. Paternal GrandFather                 8. Paternal GranMother

9. Maternal GranMother                  10. Paternal GranSons

11. Paternal GrandDaughters             12. Brothers

13. Sisters                             14. Brothers' Sons

15. Sons of Brothers' Sons              16. Father's Brothers

17. Sons of Father's Brothers

Please choose the live heir by giving the associated numeric value:
```

FIGURE 7. Input Screen of IIS

```
Please enter the amount of deceased's property: 100

*****************************Prescribed Shares*********************************
--------------------------------------------------------------------------------
Heirs                                   Number                  Shares/person
--------------------------------------------------------------------------------
Husband                                 1                       25.000000
Father                                  1                       16.666667
Mother                                  1                       16.666667
Daughters                               2                       0.000000
Sons                                    4                       0.000000
--------------------------------------------------------------------------------

The total distributed prescribed shares : 58.333333
The Remaining Shares to be distributed for residual heirs  : 41.666667
The eligible heir is : 4

*****************************Residual Shares*********************************
--------------------------------------------------------------------------------
Heirs                                   Number                  Shares/person
--------------------------------------------------------------------------------
Husband                                 1                       0.000000
Father                                  1                       0.000000
Mother                                  1                       0.000000
Daughters                               2                       4.166667
Sons                                    4                       8.333333
--------------------------------------------------------------------------------
The Multiplication Factor : 0.000000

****************************Total Earned Shares********************************
--------------------------------------------------------------------------------
Heirs                                   Number                  Shares/person
--------------------------------------------------------------------------------
Husband                                 1                       25.000000
Father                                  1                       16.666667
Mother                                  1                       16.666667
Daughters                               2                       4.166667
Sons                                    4                       8.333333
--------------------------------------------------------------------------------
```

FIGURE 8. Output Screen of IIS

TABLE 3. Experimental Result

Case	Survivors	Proposed Inheritance Calculator			Proposed Propositions		
		Prescribed Shares per person	Residual Shares per person	Total Shares per person	Prescribed Shares per person	Residual Shares per person	Total Shares per person
1	Husband	25	0	25	25	0	25
	Father	16.67	0	16.67	16.67	0	16.67
	Mother	16.67	0	16.67	16.67	0	16.67
	Son	0	41.67	41.67	0	41.67	41.67
2	Father	16.67	0	16.67	16.67	0	16.67
	Mother	16.67	0	16.67	16.67	0	16.67
	Wife	12.5	0	12.5	12.5	0	12.5
	Daughter	50	0	50	50	0	50
3	Father	16.67	16.67	33.33	16.67	16.67	33.33
	Daughter	50	0	50	50	0	50
	Paternal Granddaughter	16.67	0	16.67	16.67	0	16.67
4	Father	16.67	0	14.81	16.67	0	14.81
	Mother	16.67	0	14.81	16.67	0	14.81
	Wife	12.5	0	11.11	12.5	0	11.11
	Daughter	50	0	44.44	50	0	44.44
	Paternal Granddaughter	16.67	0	14.81	16.67	0	14.81
5	Wife	12.5	0	12.5	12.5	0	12.5
	Daughter	50	0	50	50	0	50
	Full Sister	0	37.5	37.5	0	37.5	37.5
	Full Uncle's Son	0	0	0	0	0	0
6	Paternal Grandmother	16.67	0	20	16.67	0	20
	Daughter	50	0	60	50	0	60
	Paternal Granddaughter	16.67	0	20	16.67	0	20
7	Husband	25	0	25	25	0	25
	2 Daughters	33.33	0	37.5	33.33	0	37.5
8	Wife	25	0	25	25	0	25
	Mother	33.33	0	25	33.33	0	25
	Father	0	41.67	50	0	41.67	50
9	Husband	50	0	50	50	0	50
	Mother	33.33	0	16.67	33.33	0	16.67
	Father	0	16.67	33.33	0	16.67	33.33
10	Husband	50	0	50	50	0	50
	Mother	33.33	0	33.33	33.33	0	33.33
	Paternal Grandfather	0	16.67	16.67	0	16.67	16.67
11	Mother	33.33	0	33.33	33.33	0	33.33
	Full Sister	0	0	0	0	0	0
	Paternal Grandfather	0	66.67	66.67	0	66.67	66.67
12	Husband	50	0	50	50	0	50
	Mother	33.33	0	33.33	33.33	0	33.33
	Full Sister	0	0	0	0	0	0
	Paternal Grandfather	0	16.67	16.67	0	16.67	16.67

CONCLUSION

Any software system is maintainable if it is easier, faster and less expensive to implement, debug and modify. This can be ensured through decomposing the whole system into simple and independent modules. If the nature of the decomposed modules is function, this decomposition technique is called structured paradigm. This research has developed IIS to compute the deserved shares of live heirs of deceased employing the structured paradigm. During the developed process, firstly, three levels of DFDs (0,1 and 2 level) have been proposed in the analysis phase. Then, after doing design phase, those DFDs have been transformed into structure chart. Next, the algorithms and data structure of each module of structure chart have been developed integrating all mathematical models of heirs and represented in Program Description Language. Finally, the design has been realized using structured programming language C. It has been seen that the architectural design of the system is highly cohesive due to the functional cohesiveness of each module and moderately coupling for having content coupling, common coupling, control coupling and stamp coupling among modules. Therefore, the debugging and modification effort of the proposed system has been reduced, that is, the system is maintainable. However, the first limitation of the system is it does not work for the special cases like umar's case. Secondly, the internal computational models are grounded on only full type relatives like full sisters, not for step heirs like step sisters. Finally, since each module (function) of the structure chart cannot be extended or overridden, the degree of connections between modules is not extremely loose. In future, the system will be upgraded into a general inheritance system including all rules of all religions. Again, the functions of the architectural design of this work can be replaced by classes which have some inherent properties like inheritance, polymorphism, encapsulation etc. These properties make sure more maintainable IIS. So, the next future direction of this work is that the IIS will be analyzed, designed, implemented and tested employing object-oriented paradigm.

ACKNOWLEDGMENT

The authors thank to Sheikul Hadith Moqsud Ahmed, Darul Ulum Alia Madrasah, Chittagong, Bangladesh for giving his valuable time to verify the work from shariah's aspects and continuous support.

DECLARATION OF COMPETING INTEREST

None

REFERENCES

Adelina Zuleika, N.P.D. 2013. Islamic inheritance law (faraid) and its economic implication. *Tazkia Islamic Finance and Business Review* 8(1): 97-118.

Alaa N. Akkila, Samy S. Abu Naser. 2015. Proposed expert system for calculating inheritance in Islam. *World Wide Journal of Multidisciplinar Research and Development* 2(9): 38-48.

Alshahad, H.F., Abutiheen, Z.A. 2015. Computation of inheritance share in islamic law by an expert system using decision tables. *Quarterly Adjudicated Journal for Natural and Engineering Research and Studies* 1(No.1 and 2): 105-114.

Aydemir, D.H. (no date). Division of Inheritance. https://kurandersleri.net/miras/en/Mirasen.html. Accessed on 2021-08-11

Babalola, K.O. 2017. *The Inheritance of Legal Heirs: Mathematical Presentation*. Department of Mathematics, University of Ilorin

Cheema, S. A. 2021. Distribution of inheritance under islamic law: An appraisal of online inheritance calculators. *Journal of Islamic Thought and Civilization* 11(1). https://doi.org/10.32350/jitc.111.07

Lubnaa - Inheritance Calculator (no date), http://www.lubnaa.com/money/InheritCalc.php. Accessed on: 10 August 2021.

Matha, M. P. 2008. *Object-Oriented Analysis and Design Using UML: An Introduction to Unified Process and Design Patterns*. 4th edition. Delhi: PHI Learning Private Limited, Delhi

Muhammad, B. J. 2020. The Islamic law of inheritance: Introduction and theories. Abu Aisha Publishing, PhD in View

Najeeb. 2014. Islamic Inheritance Calculator. http://inheritance.ilmsummit.org/projects/inheritance/home.aspx. Accessed on 2021-08-10

Sinha, R. 2019. A Study on Structured Analysis and Design Tools. *International Journal of Management, IT & Engineering* 9 2(1): 79-97.

Schach, S. R. 2010. *Object-Oriented and Classical Software Engineering*. 8th edition. New York: McGraw-Hill.

ShariahStandards.org. (no date), http://www.inheritancecalculator.net/. Accessed on 2021-08-10

Stevens, W. P., Myers, G. J. & Constantine, L. L. 1999. Structured design. *IBM Systems Journal* 38(2): 231-256. https://doi.org/10.1147/sj.382.0231

Tabassum, S., Hoque, A.H.M.S., Twahura, S. & Rahman, M. O. 2019. Developing an Islamic Farayez System, applying software engineering. *Jurnal Kejuruteraan* 31(1): 25-38. https://doi.org/10.17576/jkukm-2019-31(1)-04

Uttaradhikar. 2016. Access to Information (a2i), Prime Minister's O_ce of Bangladesh. http://xn--d5by7bap7cc3ici3m.xn--54b7fta0cc/index.php?lang=en. Accessed on 2021-08-10

Waqas, M. 2017. Islamic Inheritance Calculator. http://www.inheritancecalculator.com/index.php. Accessed on 2021-08-11

Yusuf, A. 2017. Controversy of Islamic law on the distribution of inheritance to the heirs of different religion. *HUNAFA: Jurnal Studia Islamika* 14(2): 377-403, https://doi.org/10.24239/jsi.v14i2.490.377-403

Zouaoui, S. & Rezeg, K. 2018. Islamic inheritance calculation system based on arabic ontology (arafamonto). *Journal of King Saud University - Computer and Information Sciences* 33(1): 68-76. https://doi.org/10.1016/j.jksuci.2018.11.015

Zulkii, A. N., Batiha, Q. A. & Qasim, M. M. 2018. Design and development of M-Faraid: An islamic inheritance mobile app. *Journal of Advanced Research in Dynamical and Control Systems* 10(10 Special Issue): 1569-1575