Logo, company name

Description automatically generatedLogo, company name

Description automatically generated

**Project senior 2**

**Project Title: smart learning “Advance Kasit”.**

**Supervised By Prof: Amjad Hudaib. & Prof Ammar Alhunaity.**

**Fall Semester.**

**2021 - 2022.**

**Icon

Description automatically generated**

|  |  |  |
| --- | --- | --- |
| # | Student name | Student ID |
| 1- | **Shahad Altamimi.** | 0181212. |
| 2- | **Ammar Arab.** | 0178071 |
| 3- | **Huda Almanaseer.** | 0178763 |
| 4- | **Malak AL Khawaldh.** | 0147015 |

***Abstract.***

**When someone wants to learn by traditional ways, students’ needs to go to class room & lost time money.**

**by our website students will not need to lose time & money when leaving home, therefore, we suggested that the solution be website by helping students to search for open source material from our houses.**

**Acknowledgments.**

**We would take this opportunity to express our sincere thanks and gratitude to our supervisor**

**Prof . Amjad Hudaib & Prof . Ammar Alhunaity for their vital support and guidance in completing this project.**

**We are also thankful to our families and fellows.**

Table of Contents

[1.0 Introduction 7](#_Toc92328711)

[1.1 Overview: 7](#_Toc92328712)

[1.2 Problem Motivation : 7](#_Toc92328713)

[1.3 Problem Statement: 7](#_Toc92328714)

[1.4 Project Aim and Objectives: 8](#_Toc92328715)

[1.5 Project Software and Hardware Requirements: 8](#_Toc92328716)

[1.6 Constrains 9](#_Toc92328717)

[1.7 Project Expected Output: 9](#_Toc92328718)

[1.8 Project Schedule: 10](#_Toc92328719)

[1.9 Report Outline : 11](#_Toc92328720)

[2.0 Related Existing Systems 13](#_Toc92328721)

[2.1 Introduction: 13](#_Toc92328722)

[2.2 Existing Systems: 13](#_Toc92328723)

[2.3 Overall Problem Statement: 13](#_Toc92328724)

[2.4 Overall Solution Approach: 14](#_Toc92328725)

[2.5 Summary: 14](#_Toc92328726)

[3.0 System Requirements & Analysis 15](#_Toc92328727)

[3.1 Introduction: 15](#_Toc92328728)

[3.2 Feasibility Study: 15](#_Toc92328729)

[3.3 Requirements Elicitation Techniques: 19](#_Toc92328730)

[3.4 Software Development Life Cycle 19](#_Toc92328731)

[3.5 Targeted Users: 20](#_Toc92328732)

[3.6 Functional Requirements Definition: 21](#_Toc92328733)

[3.7 Functional Requirements Specification : 23](#_Toc92328734)

[3.8 Non-Functional Requirements: 26](#_Toc92328735)

[3.9 Architectural Design***.*** 27](#_Toc92328736)

[3.10 Summary: 28](#_Toc92328737)

[4.0 System Design 29](#_Toc92328738)

[4.1 Introduction: 29](#_Toc92328739)

[4.2 UML Class Diagram 29](#_Toc92328740)

[. 29](#_Toc92328741)

[4.3 Use Case Diagram 30](#_Toc92328742)

[4.4 Sequence Diagram 34](#_Toc92328743)

[4.5 Summary 37](#_Toc92328744)

[5.0 System Implementation 38](#_Toc92328745)

[5.1 Introduction: 38](#_Toc92328746)

[5.2 Code Implementation: 38](#_Toc92328747)

[5.3 Database Implementation: 38](#_Toc92328748)

[5.4 Graphical User Interface Implementation 47](#_Toc92328749)

[5.5 Summary 56](#_Toc92328750)

[6.0 System Testing and Installation 57](#_Toc92328751)

[6.1 Introduction: 57](#_Toc92328752)

[6.2 Heuristics Evaluation: 58](#_Toc92328753)

[6.3 Cooperative Evaluation: 59](#_Toc92328754)

[6.3.1 Pre-Evaluation Procedures: 60](#_Toc92328755)

[6.3.2 Evaluation Procedures: 60](#_Toc92328756)

[6.3.3 Post-Evaluation Procedures: 65](#_Toc92328757)

[6.4 System Installation: 66](#_Toc92328758)

[6.5 Summary: 66](#_Toc92328759)

[7.0 Project Conclusion and Future Work 67](#_Toc92328760)

[7.1 Introduction: 67](#_Toc92328761)

[7.2 Overall Weaknesses: 67](#_Toc92328762)

[7.3 Overall Strengths: 67](#_Toc92328763)

[7.4 Future Work: 67](#_Toc92328764)

[7.5 Summary: 67](#_Toc92328765)

[8.0 Tools & reference 68](#_Toc92328766)

[APPENDIX A 69](#_Toc92328767)

[Appendix B 80](#_Toc92328768)

[APPENDIX C 81](#_Toc92328769)

[Questionnaire Result 82](#_Toc92328770)

|  |
| --- |
| List of tables |
| Table 1.1 : Software requirements for Advance KASIT |
| Table 1.2 : Hardware requirements for Advance KASIT. |
| Table 1.3 : Project Schedule Management in developing Advance KASIT. |
| Table 3.1 : Personal Costs. |
| Table 3.2 : New Hardware’s & Software’s. |
| Table 3.3 : Tangible Benefits. |
| Table 3.4 : Payback Analysis. |
| Table 3.5 : Administration requirement definition |
| Table 3.6 : Instructor requirement definition |
| Table 3.7 : Student requirement definition |
| Table 3.8 : Administrator functional requirement’s |
| Table 3.9 : Instructor functional requirement’s |
| Table 3.10 : Student functional requirement’s |
| Table 3.11 : Non Functional Requirements. |
| Table 6.1 : Test Cases |
| Table 6.2 : Summary of Violations by Heuristics. |
| Table 6.3 Summary of Violations by Severity Rating For Participant (1). |
| Table 6.4 : Summary of Violations by Severity Rating For Participant (2). |
| Table 6.5 Summary of Violations by Severity Rating For Participant (3). |
| Table 6.6 : Participants Details. |
| Table 6.7 : Cooperative Evaluation for Advance KASIT for Participant (1). |
| Table 6.8 : Cooperative Evaluation for Advance KASIT for Participant (2). |
| Table 6.9 Cooperative Evaluation for Advance KASIT for Participant (3). |
| Table 6.10 : Task Completion Times in Minutes and Seconds. |
| Table 6.11 : Participants Responses to the Post-Test Questionnaire. |

|  |
| --- |
| List of figures |
| Figure 1.1 : Gant chart for Adv-Kasit. |
| Figure 3.1 : Waterfall SDLC. |
| Figure 3.2: Architectural Design. |
| Figure 4.1 : Class Diagram. |
| Figure 4.2 : Admin Use Case Diagram. |
| Figure 4.3 : Instructors Use Case Diagram. |
| Figure 4.4 : Students Use Case Diagram. |
| Figure 4.5 : Administrator Login Sequence Diagram. |
| Figure 4.6 : Administrator Logout Sequence Diagram. |
| Figure 4.7 : Instructors Login Sequence Diagram. |
| Figure 4.8 : Instructors Logout Sequence Diagram. |
| Figure 4.9 : Students Login Sequence Diagram. |
| Figure 4.10 : Students Logout Sequence Diagram. |
| Figure 5.1 : Database implementation . |

# Introduction

# 1.1 Overview:

We all become superior a certain subject in university to the extent that we feel like we are very creative and intelligent while studying it, therefore, we as students see in ourselves the potentials and abilities to teach this subject to other students who are facing difficulties to understand it, we as information technology students should help those students by developing a web application that connects superior students and students who need additional help besides their instructors help.

We expect a high quality of service and support to the users of our project and this is the main target we are hardly working to reach.

This documentation is an accurate and specific explanation about how we worked to build a good based platform to sustain user’s requests and replies.

This project is especially made for King Abdullah School of Information Technology, as well as its applicable at any other faculty and educational institutes, which runs on any browser.

# 1.2 Problem Motivation :

We as students and web developers find this a very essential web application that is required in our faculty, therefore this motivate us to apply our skills with passion.

# 1.3 Problem Statement:

There is no existing for a specific online platform specialized only in sharing superior students expertise with other students.

# 1.4 Project Aim and Objectives:

The aim of this system consider under the following notes:

- Facilitating and accelerating the process of finding the qualified student who teaches a certain course.

- Participate in the process of making our faculty a (smart faculty).

- A chance for students with same interests to build relationships that maybe will lead them to cooperate in useful efforts in future.

# 1.5 Project Software and Hardware Requirements:

* Software requirement:

|  |  |
| --- | --- |
| **Requirement** | **Software** |
| Windows 8.1 or higher operating systems | Operating system |
| Internet Explorer 8.0 (64-bit Internet Explorer 7.0 is not supported), Firefox 3.0, Safari 4.1.2, and Google Chrome 11.0, all of those browsers in the minimum versions. | Browser |
| Notepad ++ | Development Tools |
| Apache Server 1.7.1 or Higher included MySQL | SQL Server |

Table 1.1 : Software requirements for Advance KASIT

1. Hardware requirement:

Table 1.2 Hardware requirements for Advance KASIT.

|  |  |
| --- | --- |
| **Requirement** | **Hardware** |
| Core i5 - 1480 MHz Pentium minimum, V - 1 GHz or higher recommended | Computer |
| Intel Core 3 Duo E7300 Opteron / Xeon Server CPU(Opteron 2356/Xeon5300)ييلا | CPU |
| 2 GB or 4 GB | Memory (RAM) |
| 500 GB | Hard disk |

# 1.6 Constrains

We can’t be sure that the new volunteered students are qualified enough to teach a certain course due to reviews, as well as we may not find volunteered students for all courses.

# 1.7 Project Expected Output:

We expect from our system to provide a suitable online environment that correlate students, increase the communication among them, and increase the expertise for both volunteered and regular students through a web platform that allow students to sign up and create their profiles and fill them with their personal as well as educational information.

# 1.8 Project Schedule:

Project management is the discipline of planning, organizing, securing and managing resources to achieve specific goals. The following table displays the project management:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task** | **Description** | **Start Date** | **End Date** | **Duration** | **Dependency** |
| T1 | Information Gathering | 30-10-2021 | 1-11-2021 | 3 Days |  |
| T2 | Planning | 5-11-2021 | 10-11-2021 | 6 Days | T1 |
| T3 | Analysis | 11-11-2021 | 23-11-2021 | 12 Days | T2 |
| T4 | Design | 18-12-2021 | 23-12-2021 | 6 Days | T3 |
| T5 | Implementation | 25-12-2021 | 27-12-2021 | 3 Days | T4 |
| T6 | Testing | 28-12-2021 | 2-1-2022 | 6 Days | T5 |
| T7 | Documentation | 1-12-2021 | 5-1-2022 | 36 Days | T1, T2, T3, T4, T5, T6 |
| T8 | Submission |  | 6-1-2022 |  | T7 |

Timeline

Description automatically generated with medium confidenceTable 1.3 : Project Schedule Management in developing Advance KASIT.

Figure 1.1 : Gant chart for Adv-Kasit.

# 1.9 Report Outline :

**Chapter one: ‘Introduction’** shows a overview in section 1.1, the project motivation is elaborated in Section 1.2, in additional the problem statement is stated in Section 1.3,Section 1.4 list the project aim and objectives, Section 1.5 provide the project software and hardware requirement and Section 1.6 highlights the project constrains, furthermore project expected output is addressed in Section 1.7, whereas the project is scheduled is Section 1.8, the report outline in finally present in Section 1.9.

**Chapter two “existing system”** show an introduction in Section 2.1, the existing system is elaborated in Section 2.2, in addition the overall problems of existing system are stated in Section 2.3, Section 2.4, over-all solution approach, whereas the summary is identified in Section 2.5.

**Chapter three “System Requirements & Analysis”** show an introduction in Section 3.1, requirements elicitation techniques are elaborated in Section 3.2, in addition the targeted users are stated in Section 3.3, Section 3.4, Functional requirements definition and functional requirement specification is identified in Section 3.5, Section 3.6 highlights the non-functional requirements, furthermore summary output is addressed in Section 3.8.

**Chapter four “System Design”** an introduction in Section 4.1 in addition the UML, Sequence Diagram is stated in Section 4.2, Section 4.3 include UML Sequence Diagram, Section 4.4 provide UML Class Diagram, furthermore summary output in addressed in Section 4.5.

**Chapter five “System Implementation”** show an introduction in Section 5.1, coding implementation include in Section 5.2, database implementation is elaborated in Section 5.3, in addition the graphical user interface implementation is stated in Section 5.4, Section 5.5, summary.

**Chapter six** **“Testing”** show an introduction in Section 6.1, heuristic evaluation in Section 6.2, in addition the cooperative evaluation is stated in Section 6.3, Section 6.4, requirements validation and completeness Section 6.5 provide system installation and Section 6.6 highlights 6.7 summaries.

**Chapter seven “Project Conclusion and Future Work “** shows an introduction in Section 7.1, overall weaknesses are elaborated in Section 7.2, and in addition the overall strengths are stated in Section 7.3 Section 7.4 future work.

# 2.0 Related Existing Systems

# 2.1 Introduction:

Most of students need additional help in their studying journey which will lead them to face some difficulties while finding someone to assist and guide them, either it was through social media tools, or face to face. And of course volunteered students want a suitable platform to enhance as well as share their knowledge.

# 2.2 Existing Systems:

There’s no existing system currently operating to provide online learning platform between students the same as KASIT Ost Hathi, in the conventional ways the students used to ask others for help face to face, through social media tools such as Facebook.com, or searching on student’s relevant references for the subject they want to learn at libraries such as TESTO inside JU, which will be time consuming.

# 2.3 Overall Problem Statement:

* The volunteered students cannot find the specialized platform to share and enhance their knowledge.
* It’s hard for new students to get help.
* The students who face a problem in finding others to teach them they usually enroll a paid course.
* There is no organized official platform to correlate students.

# 2.4 Overall Solution Approach:

We suggest online platform, that will let students comfortably and easily communicate with each other’s, and allow them to find their suggested volunteered colleagues to help them understand and master their enrolled courses.

Advance KASIT is new web platforms that will organize and schedule the teaching process between students and help them build relationships through learning as well as enhance their capabilities.

# 2.5 Summary:

We summarized at this chapter the related existing systems and trends with their problems. And we tried to explain our solutions that we found it helpful and useful. this chapter composed of five sections, the first one was an introduction about the existing system, second section talked about the existing system, The third section showed the problems in the existing system, The forth section presented the solutions for that problems, and the last section is chapter’s summary.

# 3.0 System Requirements & Analysis

# 3.1 Introduction:

This chapter include in details about the system requirements and system analysis with full description  
about the functionality of the system including the definition and the specification for each requirement.

# 3.2 Feasibility Study:

* **Technical Feasibility:**
* The system is developing as online web portal for e-services as web application online tool opening through internet browsers that supported by Web 2.0 feature, and this system could be run in any time or any place if internet connection is available.
* Design Language: HTML, CSS, JQuery.
* Programming Language: PHP, and MySQL for database codes.
* Hardware: Any technical device include internet browser can open this application, and internet access.
* **Operational Feasibility:**
* Web applications are familiar to wide users, no need too much for training on it, it is easy to use.
* Only the administrator of the system can access the database and modify the data inside it.
* **Economic Feasibility:**
* **Development Cost:**
* **Personal Cost:**

Table 3.1 : Personal Costs.

|  |  |  |  |
| --- | --- | --- | --- |
| **Employee** | **Cost per hour** | **Hours** | **Total cost per hour** |
| 1 System Analyst | 30 JD | 20 hr | 600 JD |
| 1 System Administrator | 15 JD | 30 hr | 450 JD |
| 2 Programmer | 20 JD | 130 hr | 2600 JD |
| 1 GUI designer | 10 JD | 20 hr | 200 JD |
| 1 Database Specialist | 10 JD | 15 hr | 150 JD |
| Total 4000 JD | | | |

* **New Hardware’s and Software’s:**

Table 3.2 : New Hardware’s and Software’s.

|  |  |
| --- | --- |
| **Hardware & Software** | **Cost** |
| 1 Computer | 550 JD |
| 1 DBMS Software | 350 JD |
| 1 Development Server | 500 JD |
| Total | 1400 JD |

* **Development Cost** = 4000 + 1400 = 5400 JD - (For Development Year).
* **Operating Cost:**
  + **Employees:**
    - * 2 Programmers (130 hr each 20 JD/hr) = 1300 JD.
      * 1 System Administrator (1000 hr each 5 JD/hr) = 5000 JD.
      * Total= 6300 JD.
  + **Expenses:**
    - * 1 Maintenance agreement for server (350 JD).
      * 1 Maintenance agreement for DBMS Software (200 JD).
      * Total= 550 JD.
  + **Operating Cost** = 6300 + 550 = 6850 JD - (For First Operating Year).
* **Benefits:**
* **Tangible Benefits:** Benefits that can be measured in money and with certainty that resulted from information forms.

Table 3.3 : Tangible Benefits.

|  |
| --- |
| **TANGIBLE BENEFITS WORKSHEET : Year 1 – 5** |
| 1. Cost reduction or avoidance communication gap in the old process 1,650 JD |
| 2. Error reduction 2,800 JD |
| 3. Increased flexibility 2,750 JD |
| 4. Increased speed of activity 2,400 JD |
| 5. Improvement in management planning or control 2,100 JD |
| **Total Tangible Benefits = 11700 JD** |

* **Intangible Benefits:** Benefits that cannot be easily measured in money and with certainty that resulted from information forms.
* Payback Analysis:

Table 3.4 : Payback Analysis.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cash flow | Year 0 | Year 1 | Year 2 | Year 3 | Year 4 | Year 5 |
| Development Cost | (5400) | 0 | 0 | 0 | 0 | 0 |
| Operating Cost | 0 | (6850) | (7850) | (8850) | (9850) | (10850) |
| Benefits | 0 | 11700 | 12700 | 13700 | 14700 | 15700 |
| Discount Rate (10%) | 1 | 0.9091 | 0.8264 | 0.7513 | 0.6830 | 0.6209 |
| Time Adjusted Cost | (5400) | (6227) | (6487) | (6649) | (6728) | (6737) |
| Time Adjusted Benefits | 0 | 10636 | 10495 | 10293 | 10040 | 9748 |
| Cumulative Time Adjusted Cost | (5400) | (11627) | (18114) | (24763) | (31491) | (38228) |
| Cumulative Time Adjusted Benefits | 0 | 10636 | 21131 | 31424 | 41464 | 51212 |
| Cumulative Time Adjusted (Cost& Benefits) | (5400) | (991) | 3017 | 6661 | 9973 | 12984 |

* Payback Year = The Second Year (Year 2).
* Life Time ROI (Return on Investment) = (51212 - 38228) / 38228 = 3%, (Estimated Lifetime Benefits - Estimated Lifetime Cost) / Estimated Lifetime Cost = 3%.
* Annual ROI = Lifetime ROI / Lifetime of The System = 0.06 = 1%
* Net Present Value = Cumulative Benefits - Cumulative Cost = 51212 - 38228 = 12984

# 3.3 Requirements Elicitation Techniques:

Good requirements start with good sources. Finding those quality sources is an important task. After you have identified these sources, there are a number of techniques that may be use to gather requirements.

**We used the following techniques:**

* **Questionnaires:**A form has been created and published on the university groups to survey students' opinions about the application and the results have been placed inside “Appendix C ”.

# C:\Users\Saleh Basyouni\Desktop\Graduation Project 2015-2016\KASIT College Automation System\Figures\SDLC\SDLC.png3.4 Software Development Life Cycle

Figure 3.1 : Waterfall SDLC.

* **Project Initiation and Planning:** In this section, we define project feasibility, identify project and identify its scope, and we discuss it with our supervisors to make sure that we choose a good idea to work on it.
* **Project Analysis:** In this phase, we will understand system needs and requirements, then we structure these requirements; we mean providing description for system activities and processes, also we make a fully feasibility study about the whole project, then define the functionality of the system.
* **Project Design:** The main phase activities is to identify Database relations, transfer these relations to ER-Diagram that show all overall system relations then transform ER-Diagram to physical Tables, after that we will design system Forms and Reports.
* **Implementation:** This phase includes Coding, installations and documentation completion of the system that we start to implement the design of the system then we connected with the functionality code
* **Testing: During this phase**, this system will be test with real data to check about every function and how it works as a live online test.
* **Maintenance:** This is the final phase of the SDLC to check the Information System if it is systematically repaired and improved.

# 3.5 Targeted Users:

The website is especially created for “King Abdullah School of Information Technology (KASIT)” students, but it is also available to all students who may get benefit from the services provided by the website.

# 3.6 Functional Requirements Definition:

**-Administrator section:**

Table 3.5 : Administration requirement definition

|  |  |
| --- | --- |
| **Name** | **Description** |
| **Login** | Administrator can access the system by special username and password. |
| **Logout** | Administrator can logout from the system by send request to end the current session. |
| **Courses Management** | Administrator will manage courses table including the following functions (Add, Edit, Delete and Display). |
| **Instructors Management** | Administrator will manage instructors table including the following functions (Add, Edit, Delete and Display). |
| **Libraries Management** | Administrator will manage libraries table including the following functions (Add, Edit, Delete and Display). |
| **Volunteers Courses Packages Management** | Administrator will manage courses packages table including the following functions (Delete and Display). |

**- Instructors Section:**

Table 3.6 : Instructor requirement definition

|  |  |
| --- | --- |
| **Name** | **Description** |
| **Login** | Instructors user can access the system by a special username and password. |
| **Logout** | Instructors user can logout from the system by send request to end the current session. |
| **Volunteers Courses Packages Management** | Instructors user will manage courses packages table including the following functions (Edit, Delete and Display). |

**-Students Section:**

Table 3.7 : Student requirement definition

|  |  |
| --- | --- |
| **Name** | **Description** |
| **Login** | Students user can access the system by a special username and password. |
| **Logout** | Students user can logout from the system by send request to end the current session. |
| **Volunteers Courses Packages Management** | Students user will manage courses packages table including the following functions (Add, Delete and Display). |

# 3.7 Functional Requirements Specification :

**- Administration Section:**

Table 3.8 : Administrator functional requirement’s

|  |  |  |
| --- | --- | --- |
| **R#** | **Title** | **Description** |
| FR1 | **Login** | The administrator will access the administration section through the admin login area, after enter the username and the password for that admin. This shall be the basis for determining the class privilege corresponding with the assigned admin username and password. The login should be validated before access to the system is approved, and there is a counter of failed login process that will enter the user in block if they try more than four times as a failed login status. |
| FR2 | **Logout** | The administrator can logout from the system by press on logout button to destroy the current session. |
| FR3 | **Courses Management** | The administrator can manage courses table after success login to administrator area by active all management functions (Add, Edit, Delete and Display the table list). |
| FR4 | **Instructors Management** | The administrator can manage instructors table after success login to administrator area by active all management functions (Add, Edit, Delete and Display the table list). |
| FR5 | **Libraries Management** | The administrator can manage libraries table after success login to administrator area by active all management functions (Add, Edit, Delete and Display the table list). |
| FR6 | **Volunteers Courses Packages Management** | The administrator can manage courses packages table after success login to administrator area by active all management functions (Delete and Display the table list). |

**- Instructors Section:**

Table 3.9 : Instructor functional requirement’s

|  |  |  |
| --- | --- | --- |
| **R#** | **Title** | **Description** |
| FR1 | **Login** | The instructor’s user will access the instructor’s section through the instructor’s login area, after enter the username and the password for that instructor’s. This shall be the basis for determining the class privilege corresponding with the assigned instructor’s username and password. The login should be validated before access to the system is approved, and there is a counter of failed login process that will enter the user in block if they try more than four times as a failed login status. |
| FR2 | **Logout** | The instructor’s user can logout from the system by press on logout button to destroy the current session. |
| FR3 | **Volunteers Courses Packages Management** | The instructor’s user can manage courses packages table after success login to instructor’s area by active all management functions (Edit, Delete and Display the table list). |

**-Students Section:**

Table 3.10 : Student functional requirement’s

|  |  |  |
| --- | --- | --- |
| **R#** | **Title** | **Description** |
| FR1 | **Login** | The student’s user will access the student’s section through the student’s login area, after enter the username and the password for that student’s. This shall be the basis for determining the class privilege corresponding with the assigned student’s username and password. The login should be validated before access to the system is approved, and there is a counter of failed login process that will enter the user in block if they try more than four times as a failed login status. |
| FR2 | **Logout** | The student’s user can logout from the system by press on logout button to destroy the current session. |
| FR3 | **Volunteers Courses Packages Management** | The student’s user can manage courses packages table after success login to student’s area by active all management functions (Add, Delete and Display the table list). |

# 3.8 Non-Functional Requirements:

Table 3.11 : Non Functional Requirements.

|  |  |  |
| --- | --- | --- |
| **Non-Functional Requirement #** | **Title** | **Description** |
| NFR1 | Security | Our website is full secure by a username and password for the administrator, to make sure all the control features only with the admin. Also each user inside the system have an encryption password using MD5 algorithm. |
| NFR2 | Performance | The website should run fast with no propagation, in order to provide the best service in files management. |
| NFR3 | Reliability | The system should be as reliable as possible since we have critical information about all users and all process types. |
| NFR4 | Ease Of Use | Our website has a friendly interface that can be used by the different users. |
| NFR5 | Availability | Our system will be available for all users through a web application and using web browsers, and this system will be available for users anywhere and anytime through internet service, once they need it**.** |
| NFR6 | Flexibility | If system management intends to increase or extend the functionality of the system after it is developed, that should be planned from the beginning; it influences choices made during the design, development, testing, and deployment of the system.  So far, our system is an open source system that system management can customize the functionality of the system as they need. |

## 3.9 Architectural Design.

Administration : edit & delete course, add new course , add new package , edit & delete package , accept & reject student request , edit & delete library.

Course & packages : view course package , course name , add new package view package file , delete course package , view course schedule.

Library : library list , view library , library name & phone number , view location ,

Diagram

Description automatically generated

Figure 3.2 : Architectural Design.

# 3.10 Summary:

Finally based on the previous titles, this chapter define the SDLC module that we worked on it and include our system requirements and its analysis with full description about the functionality of the system including the definition and the specification for each requirement; also, this chapter include the non-functional requirements that support our system.

# 4.0 System Design

# 4.1 Introduction:

This chapter includes manly important figures that describe our system process, it will include class diagram & use cases diagrams, sequences diagrams, class diagrams and user interface (GUI) designs.

# 4.2 UML Class Diagram

Diagram

Description automatically generated with medium confidence

Figure 4.1 : Class Diagram.

# .

# 4.3 Use Case Diagram

* **Administrator Section:**

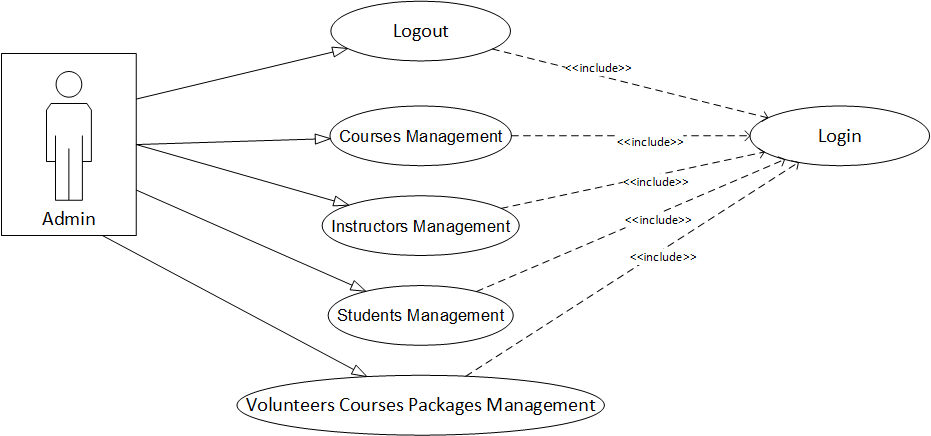


Figure 4.2: Admin Use Case Diagram.

|  |  |
| --- | --- |
| Use Case Name | Login & logout |
| Actor | Admin. |
| Description | The administrator will enter a username and password to access the admin profile.& the administrator can log out from the system by sending request to end the current session. |
| Response | Admin access the administrator profile inside the system.  Admin logout from the administrator profile inside the system. |
| Comments |  |

|  |  |
| --- | --- |
| Use Case Name | Courses Management |
| Actor | Admin. |
| Description | The administrator will manage the courses table by active all management functions (Add, Edit, Delete and Display the table list) |
| Response | The courses table inside the database will be managed by the administrator |
| Comments |  |

|  |  |
| --- | --- |
| Use Case Name | Instructors Management |
| Actor | Admin. |
| Description | Administrator will manage instructors table by active all management functions (Add, Edit, Delete and Display the table list) |
| Response | Instructors table inside the database will be managed by the administrator |
| Comments |  |

|  |  |
| --- | --- |
| Use Case Name | Students Management |
| Actor | Admin. |
| Description | The administrator will manage students table by active all management functions (Edit, Delete, and Display the table list) |
| Response | Students table inside the database will be managed by the administrator |
| Comments |  |

* **Instructors Section:**

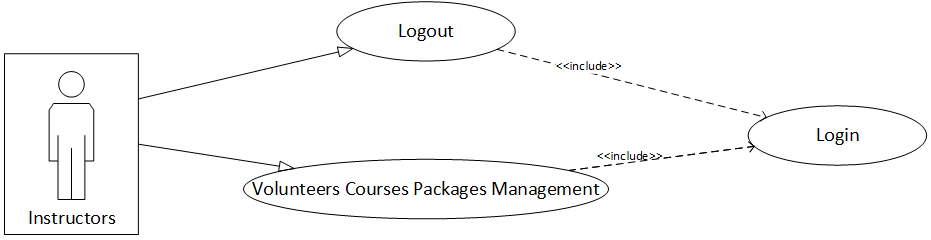


Figure 4.3 : Instructors Use Case Diagram.

|  |  |
| --- | --- |
| Use Case Name | Login & logout |
| Actor | Instructors. |
| Description | The instructors will enter a username and password to access the instructor’s profile & Instructors can log out from the system by sending a request to end the current session. |
| Response | Instructors access the instructor’s profile inside the system. Instructors log out from the instructor’s profile inside the system. |
| Comments |  |

|  |  |
| --- | --- |
| Use Case Name | Courses Packages Management |
| Actor | Instructors. |
| Description | Instructors will manage the courses packages table by active the management functions (Edit, Delete, and Display the table list) |
| Response | Courses Packages table inside the database will be managed by Instructors |
| Comments |  |

**-Students Section:`**

Diagram

Description automatically generated with low confidence

Figure 4.4 : Students Use Case Diagram.

|  |  |
| --- | --- |
| Use Case Name | Login & logout |
| Actor | Students. |
| Description | The students will enter a username and password to access the student’s profile , and students can log out from the system by sending a request to end the current session. |
| Response | Students access the student’s profile inside the system. Students log out from the student’s profile inside the system. |
| Comments |  |

|  |  |
| --- | --- |
| Use Case Name | Courses Packages Management |
| Actor | Students. |
| Description | Students will manage the courses packages table by active all management functions (Add, Edit, Delete, and Display the table list) |
| Response | The courses Packages table inside the database will be managed by Students |
| Comments |  |

# 4.4 Sequence Diagram

* **Administrator Section:**
  + **Login:**

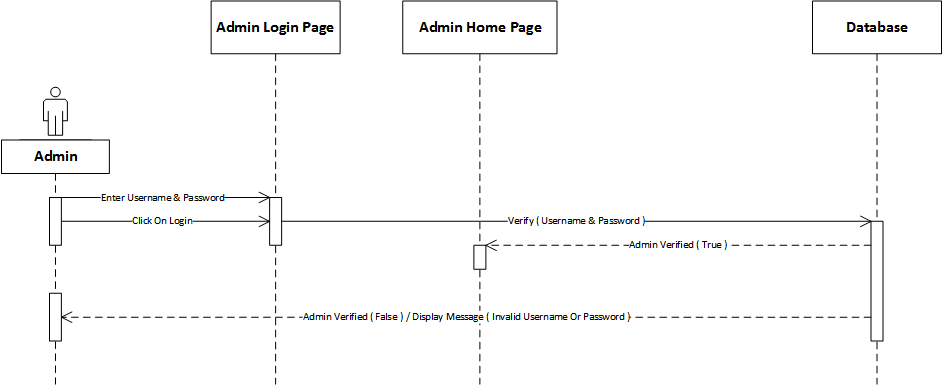


Figure 4.5 : Administrator Login Sequence Diagram.

* + **Logout:**

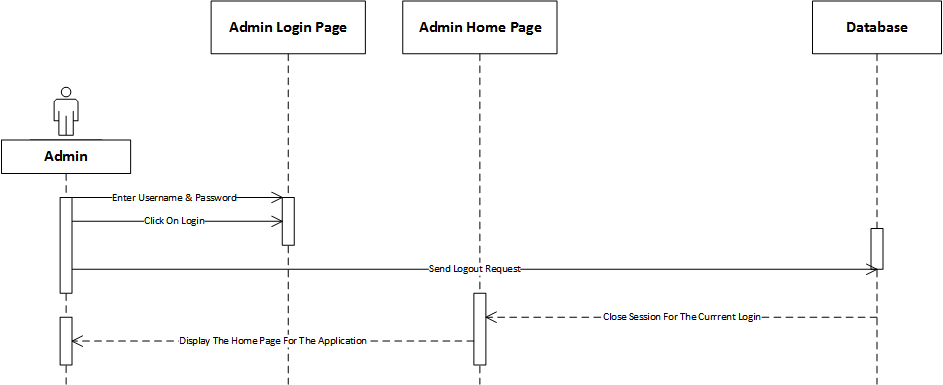


Figure 4.6 : Administrator Logout Sequence Diagram.

* **Instructors Section:**
  + **Login:**

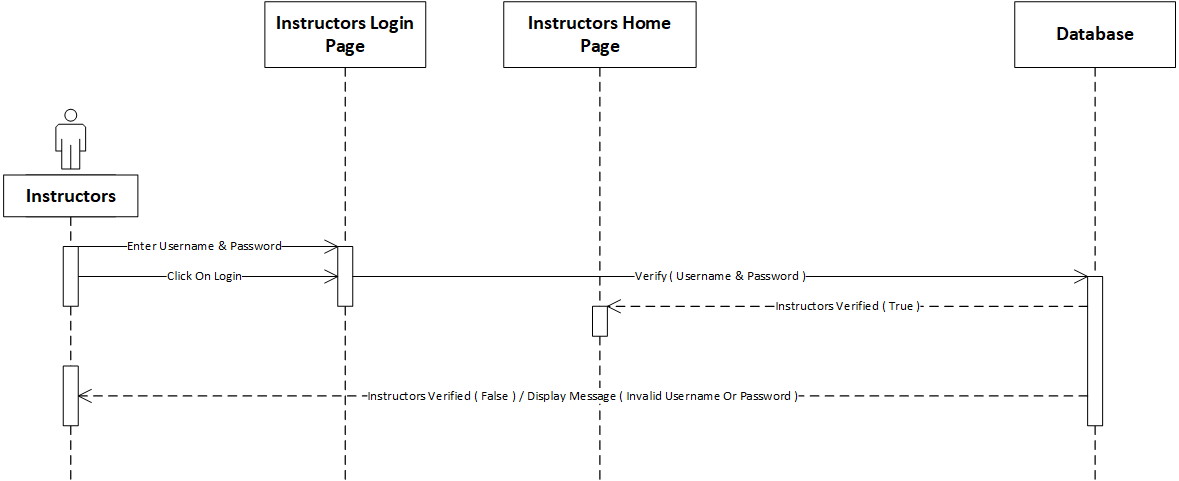


Figure 4.7 : Instructors Login Sequence Diagram.

* + **Logout:**

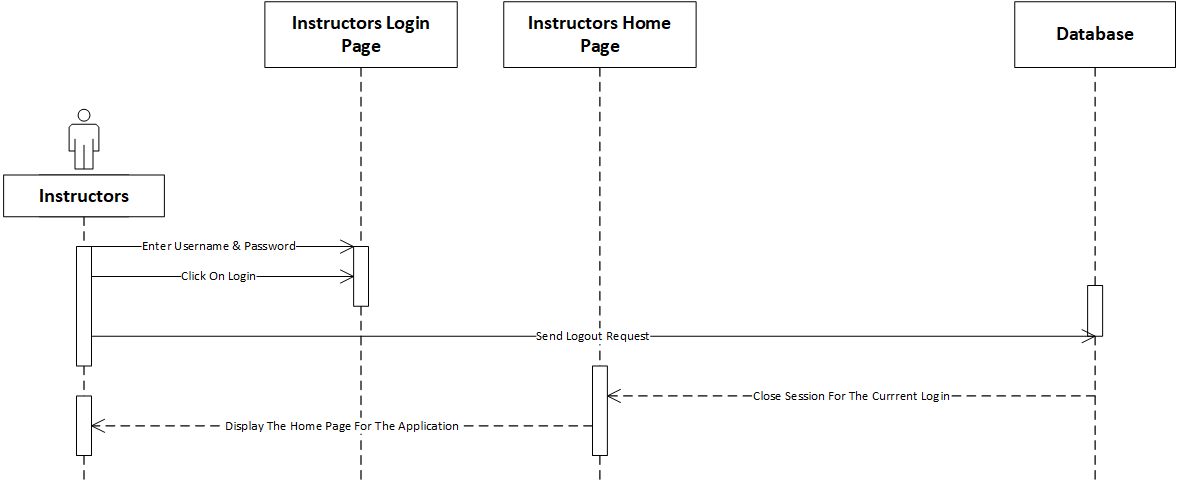


Figure 4.8 : Instructors Logout Sequence Diagram.

* **Students Section:**
  + **Login:**

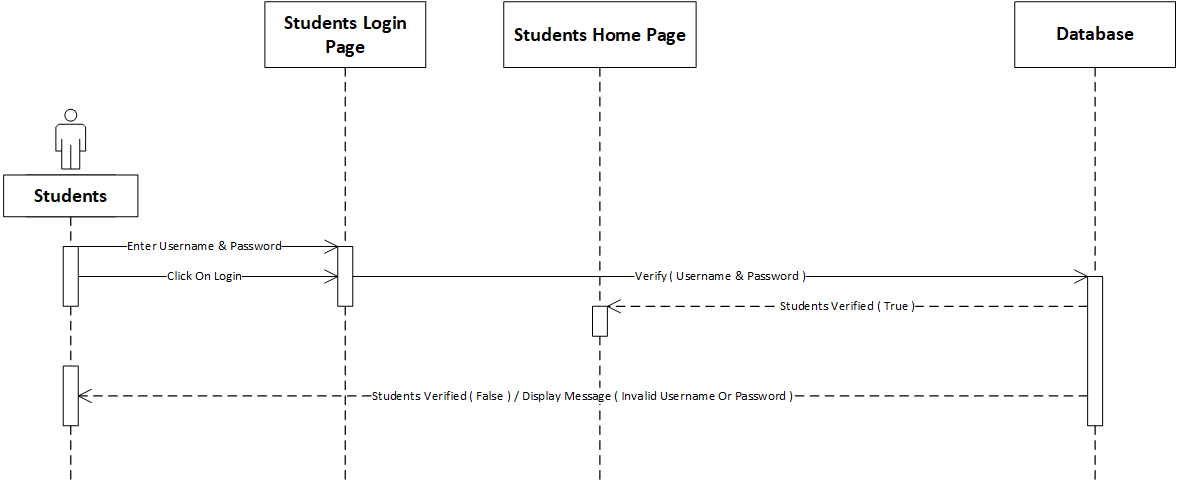


Figure 4.9 : Students Login Sequence Diagram.

* + **Logout:**

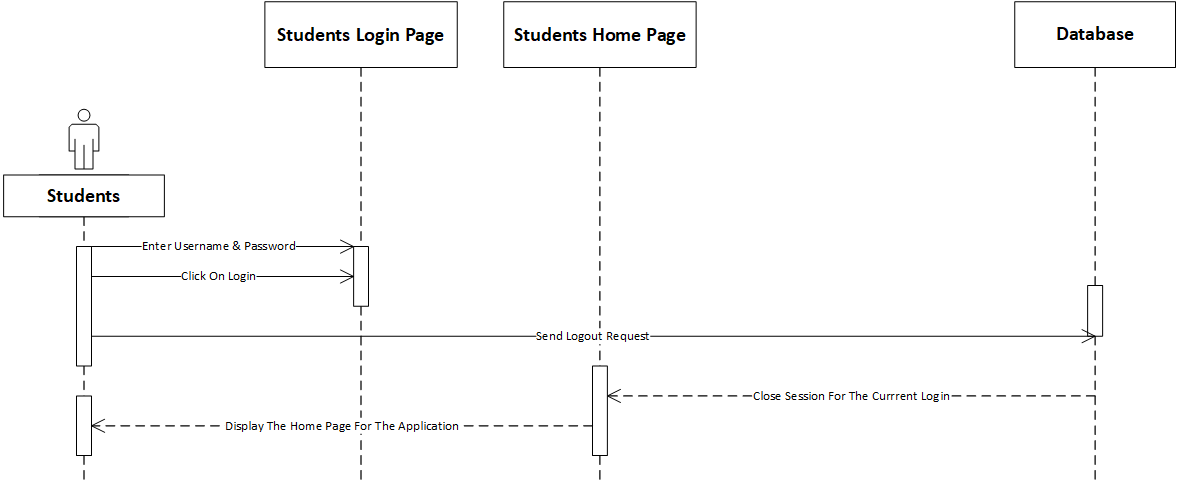


Figure 4.10 : Students Logout Sequence Diagram.

# 4.5 Summary

At final, this chapter define our system functionality and how the data transfer inside it the methodology of the system, that we build the context level and 0 level for the system functions, then design the entity relationship diagram for database, after that we identified use cases diagrams, sequence diagrams and class diagram.

# 5.0 System Implementation

# 5.1 Introduction:

System implementation it’s an important phase is the system, at this stage designer start design the system from interface to end of the database implementation. Therefore, this chapter describe the database implementation relations, also the graphical user interface implementation figures.

# 5.2 Code Implementation:

The source code are inside the file project

# 5.3 Database Implementation:

Diagram

Description automatically generated with medium confidence

Figure 5.1 : Database implementation .

**-Database Dictionary:**

classes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **Media (MIME) type** |
| ID (Primary) | int(20) | No |  |  |  |  |
| Name | varchar(250) | No |  |  |  |  |
| Department\_Name | varchar(250) | No |  |  |  |  |

Indexes

| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PRIMARY | BTREE | Yes | No | ID | 0 | A | No |  |

courses

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **Media (MIME) type** |
| ID (Primary) | int(20) | No |  |  |  |  |
| Course\_Name | varchar(250) | No |  |  |  |  |
| Department\_Name | varchar(250) | No |  |  |  |  |
| I\_ID | int(20) | No |  | instructors -> ID |  |  |

Indexes

| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PRIMARY | BTREE | Yes | No | ID | 0 | A | No |  |
| I\_ID | BTREE | No | No | I\_ID | 0 | A | No |  |

courses\_files

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **Media (MIME) type** |
| ID (Primary) | int(20) | No |  |  |  |  |
| P\_ID | int(20) | No |  | courses\_packages -> ID |  |  |
| Title | varchar(250) | No |  |  |  |  |
| File\_Path | varchar(250) | No |  |  |  |  |
| Total\_Rate | int(20) | No |  |  |  |  |
| Status | varchar(250) | No |  |  |  |  |
| Add\_Date\_Time | timestamp | No | current\_timestamp() |  |  |  |

Indexes

| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PRIMARY | BTREE | Yes | No | ID | 0 | A | No |  |
| P\_ID | BTREE | No | No | P\_ID | 0 | A | No |  |

courses\_packages

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **Media (MIME) type** |
| ID (Primary) | int(20) | No |  |  |  |  |
| S\_ID | int(20) | No |  | students -> Uni\_Number |  |  |
| C\_ID | int(20) | No |  | courses -> ID |  |  |
| Department\_Name | varchar(250) | No |  |  |  |  |
| Package\_Name | varchar(250) | No |  |  |  |  |
| Add\_Date\_Time | timestamp | No | current\_timestamp() |  |  |  |

Indexes

| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PRIMARY | BTREE | Yes | No | ID | 0 | A | No |  |
| S\_ID | BTREE | No | No | S\_ID | 0 | A | No |  |
| C\_ID | 0 | A | No |
| C\_ID | BTREE | No | No | C\_ID | 0 | A | No |  |

courses\_schedules

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **Media (MIME) type** |
| ID (Primary) | int(20) | No |  |  |  |  |
| Course\_ID | int(20) | No |  | courses -> ID |  |  |
| Class\_ID | int(20) | No |  | classes -> ID |  |  |
| I\_ID | int(20) | No |  | instructors -> ID |  |  |
| Days | varchar(250) | No |  |  |  |  |
| Time | varchar(250) | No |  |  |  |  |

Indexes

| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PRIMARY | BTREE | Yes | No | ID | 0 | A | No |  |
| Class\_ID | BTREE | No | No | Class\_ID | 0 | A | No |  |
| Course\_ID | BTREE | No | No | Course\_ID | 0 | A | No |  |
| I\_ID | BTREE | No | No | I\_ID | 0 | A | No |  |

files\_rates

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **Media (MIME) type** |
| ID (Primary) | int(20) | No |  |  |  |  |
| S\_ID | int(20) | No |  | students -> Uni\_Number |  |  |
| F\_ID | int(20) | No |  | courses\_files -> ID |  |  |
| Rate | int(20) | No |  |  |  |  |

Indexes

| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PRIMARY | BTREE | Yes | No | ID | 0 | A | No |  |
| S\_ID\_2 | BTREE | No | No | S\_ID | 0 | A | No |  |
| F\_ID | BTREE | No | No | F\_ID | 0 | A | No |  |

instructors

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **Media (MIME) type** |
| ID (Primary) | int(20) | No |  |  |  |  |
| Full\_Name | varchar(250) | No |  |  |  |  |
| Department\_Name | varchar(250) | No |  |  |  |  |
| Office\_Hours | varchar(250) | No |  |  |  |  |
| Username | varchar(250) | No |  |  |  |  |
| Password | varchar(250) | No |  |  |  |  |

Indexes

| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PRIMARY | BTREE | Yes | No | ID | 0 | A | No |  |

libraries

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **Media (MIME) type** |
| ID (Primary) | int(20) | No |  |  |  |  |
| Name | varchar(250) | No |  |  |  |  |
| Phone\_Number | varchar(250) | No |  |  |  |  |
| Email\_Address | varchar(250) | No |  |  |  |  |
| Address | varchar(250) | No |  |  |  |  |
| Longitude | varchar(250) | No |  |  |  |  |
| Latitude | varchar(250) | No |  |  |  |  |
| Add\_Date\_Time | timestamp | No | current\_timestamp() |  |  |  |

Indexes

| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PRIMARY | BTREE | Yes | No | ID | 0 | A | No |  |
|  |  |  |  |  |  |  |  |  |

students

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **Media (MIME) type** |
| Uni\_Number (Primary) | int(7) | No |  |  |  |  |
| Full\_Name | varchar(250) | No |  |  |  |  |
| Department\_Name | varchar(250) | No |  |  |  |  |
| Email\_Address | varchar(250) | No |  |  |  |  |
| Phone\_Number | varchar(250) | No |  |  |  |  |
| Password | varchar(250) | No |  |  |  |  |
| Volunteer\_Status | varchar(250) | No |  |  |  |  |
| Total\_Rates | int(20) | No |  |  |  |  |
| Add\_Date\_Time | timestamp | No | current\_timestamp() |  |  |  |

Indexes

| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PRIMARY | BTREE | Yes | No | Uni\_Number | 0 | A | No |  |

students\_ratings

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **Media (MIME) type** |
| ID (Primary) | int(20) | No |  |  |  |  |
| S\_ID | int(20) | No |  | students -> Uni\_Number |  |  |
| V\_ID | int(20) | No |  | students -> Uni\_Number |  |  |
| Rate | int(20) | No |  |  |  |  |

Indexes

| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PRIMARY | BTREE | Yes | No | ID | 0 | A | No |  |
| S\_ID | BTREE | No | No | S\_ID | 0 | A | No |  |
| V\_ID | BTREE | No | No | V\_ID | 0 | A | No |  |

volunteers\_courses

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **Media (MIME) type** |
| ID (Primary) | int(20) | No |  |  |  |  |
| S\_ID | int(20) | No |  | students -> Uni\_Number |  |  |
| C\_ID | int(20) | No |  | courses -> ID |  |  |

Indexes

| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PRIMARY | BTREE | Yes | No | ID | 0 | A | No |  |
| S\_ID | BTREE | No | No | S\_ID | 0 | A | No |  |
| C\_ID | 0 | A | No |
| C\_ID | BTREE | No | No | C\_ID | 0 | A | No |  |

[Open new phpMyAdmin window](http://localhost/phpmyadmin/db_datadict.php?db=advance_kasit_f_2022&table=&server=1&target=)

# 5.4 Graphical User Interface Implementation

**Main interfaces “visitor” :**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Graphical user interface, text, application, email

Description automatically generatedStudent interfaces :**

**Graphical user interface, application, table

Description automatically generated**

**Graphical user interface, application, website

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated**

**Graphical user interface, application

Description automatically generatedInstructor interfaces :**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application

Description automatically generatedAdministration interfaces :**

**Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedA picture containing graphical user interface

Description automatically generatedText

Description automatically generated with medium confidence**

**Graphical user interface, text, application, Word, email

Description automatically generated**

**Library interface:**

**Graphical user interface, table

Description automatically generated**

**About interface:**

**Graphical user interface, application, Word

Description automatically generated**

**Contact us interface:**

**Graphical user interface, text, application

Description automatically generated**

# 5.5 Summary

In this chapter shows all implementation details for “Advance KASIT” including the database and graphical user interface implementation.

The output of this chapter is very necessary for the next chapter, whereby the testing and evaluation take place.

# 6.0 System Testing and Installation

6.1 Introduction:In this chapter will we make the test system by certified test globally the Heuristic Evaluation Test and Cooperation Evaluation Test. This chapter present introduction in Section 6.1, heuristic evaluation is elaborate in Section 6.2 in addition the cooperative evaluation is stated in Section 6.3, Section 6.4, system installation. Furthermore, this chapter is finally summarized in Section 6.5.

* **Test Cases:**

Table 6.1 : Test Cases

|  |  |  |
| --- | --- | --- |
| Input | Expected Output | Actual Output |
| User’s sign-in information (Username, Password) | Confirm user sign-in information and confirm the validation of it | Open user account inside the website |
| Add New Package | Save packages information inside the database | View packages list after saving it in database |
| Add New File | Save file information inside the database | View files list after saving it in database |

* **Test Cases:**

**Agile methodology:** Traditional software development methodologies work on the premise that software requirements remain constant throughout the project. Incremental testing is used in agile development methods and hence, every release of the project is tested thoroughly. This ensures that any bugs in the system are fixed before the next release

# 6.2 Heuristics Evaluation:

Table 6.2 : Summary of Violations by Heuristics.

|  |  |  |
| --- | --- | --- |
| **Heuristic Numbering Scheme** | **Frequency** | **Ratio (%)** |
| H1 | 5 | 7.576 |
| H2 | 5 | 7.676 |
| H3 | 5 | 7.676 |
| H4 | 10 | 15.152 |
| H5 | 7 | 10.606 |
| H6 | 6 | 9.09 |
| H7 | 8 | 12.121 |
| H8 | 5 | 7.576 |
| H9 | 5 | 7.576 |
| H10 | 10 | 15.152 |
| **Total** | **66** | **100%** |

Table 6.3 Summary of Violations by Severity Rating For Participant (1).

|  |  |  |
| --- | --- | --- |
| **Severity Rating** | **Frequency** | **Ratio (%)** |
| 0 | 15 | 0.23 |
| 1 | 12 | 0.181 |
| 2 | 10 | 0.151 |
| 3 | 9 | 0.14 |
| 4 | 20 | 0.30 |
| Total | 66 | 100% |

* For more details, refer to Appendix A.

Table 6.4 : Summary of Violations by Severity Rating For Participant (2).

|  |  |  |
| --- | --- | --- |
| **Severity Rating** | **Frequency** | **Ratio (%)** |
| 0 | 9 | 0.14 |
| 1 | 12 | 0.181 |
| 2 | 15 | 0.23 |
| 3 | 10 | 0.151 |
| 4 | 20 | 0.30 |
| Total | 66 | 100% |

* For more details, refer to Appendix A.

Table 6.5 Summary of Violations by Severity Rating For Participant (3).

|  |  |  |
| --- | --- | --- |
| **Severity Rating** | **Frequency** | **Ratio (%)** |
| 0 | 12 | 0.181 |
| 1 | 10 | 0.151 |
| 2 | 15 | 0.23 |
| 3 | 9 | 0.14 |
| 4 | 20 | 0.30 |
| Total | 66 | 100% |

* For more details, refer to Appendix A.

# 6.3 Cooperative Evaluation:

After using the tool and answering the Advance KASIT test, please indicate the extent to which you agree or disagree with each of the following statements regarding to your experience with the system.

* Cooperative Evaluation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Criteria** | **Participant 1** | **Participant 2** | **Participant 3** |
| **1.** | **Gender** |  |  |  |
| **2.** | **Age** |  |  |  |
| **3.** | **Educational Level** |  |  |  |
| **4.** | **Programmer Taken** |  |  |  |
| **5.** | **Institution** | University of Jordan | University of Jordan | University of Jordan |

Table 6.6 : Participants Details.

# 6.3.1 Pre-Evaluation Procedures:

Participants were contacted through telephone conversations asking them the possibility to participate in the co-operative evaluation. A brief introduction to Advance KASIT was given to the participants 10 minutes before they started the evaluation, and participants were asked to read that introductory document. The document also has a list of tasks, which will be performed by the participants throughout the co-operative evaluation. Users were told that they need to think aloud when facing any problem in the system. They were also told that, each task they perform is monitored and timed.

# 6.3.2 Evaluation Procedures:

During the evaluation session, a moderator accompanied the users to do the co-operative evaluation. A comment from shown in Appendix B was used by the moderator to write down the comments of each user for each task. Users were helped when they really face serious problems performing the tasks. The following tables show the comments from pre-pared by the moderator for each participant.

Table 6.7 : Cooperative Evaluation for Advance KASIT for Participant (1).

|  |  |  |  |
| --- | --- | --- | --- |
| **Task No.** | **Test** | **Time Taken to Complete the Task** | **Comments** |
| **A.** | **Administrator** | | |
| **1.** | Login |  |  |
| **2.** | Logout |  |  |
| **4.** | Courses Management |  |  |
| **5.** | Instructors Management |  |  |
| **6.** | Libraries Management |  |  |
| **7.** | Students Management |  |  |
| **8.** | Volunteers Courses Packages Management |  |  |
| **B.** | **Instructors** | | |
| **1.** | Login |  |  |
| **2.** | Logout |  |  |
| **3.** | Volunteers Courses Packages Management |  |  |
| **C.** | **Students** | | |
| **1.** | Login |  |  |
| **2.** | Logout |  |  |
| **3.** | Volunteers Courses Packages Management |  |  |

Table 6.8 : Cooperative Evaluation for Advance KASIT for Participant (2).

|  |  |  |  |
| --- | --- | --- | --- |
| **Task No.** | **Test** | **Time Taken to Complete the Task** | **Comments** |
| **A.** | **Administrator** | | |
| **1.** | Login |  |  |
| **2.** | Logout |  |  |
| **4.** | Courses Management |  |  |
| **5.** | Instructors Management |  |  |
| **6.** | Libraries Management |  |  |
| **7.** | Students Management |  |  |
| **8.** | Volunteers Courses Packages Management |  |  |
| **B.** | **Instructors** | | |
| **1.** | Login |  |  |
| **2.** | Logout |  |  |
| **3.** | Volunteers Courses Packages Management |  |  |
| **C.** | **Students** | | |
| **1.** | Login |  |  |
| **2.** | Logout |  |  |
| **3.** | Volunteers Courses Packages Management |  |  |

Table 6.9 Cooperative Evaluation for Advance KASIT for Participant (3).

|  |  |  |  |
| --- | --- | --- | --- |
| **Task No.** | **Test** | **Time Taken to Complete the Task** | **Comments** |
| **A.** | **Administrator** | | |
| **1.** | Login |  |  |
| **2.** | Logout |  |  |
| **4.** | Courses Management |  |  |
| **5.** | Instructors Management |  |  |
| **6.** | Libraries Management |  |  |
| **7.** | Students Management |  |  |
| **8.** | Volunteers Courses Packages Management |  |  |
| **B.** | **Instructors** | | |
| **1.** | Login |  |  |
| **2.** | Logout |  |  |
| **3.** | Volunteers Courses Packages Management |  |  |
| **C.** | **Students** | | |
| **1.** | Login |  |  |
| **2.** | Logout |  |  |
| **3.** | Volunteers Courses Packages Management |  |  |

* It is important to compare the time taken by each participant to complete each single task com-pared to the default time allocated by the moderator as shown in next table.

Table 6.10 : Task Completion Times in Minutes and Seconds.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task No.** | **Default** | **Participant 1** | **Participant 2** | **Participant 3** |
| **A. Administrator Activity** |  | | | |
| **1.** |  |  |  |  |
| **2.** |  |  |  |  |
| **3.** |  |  |  |  |
| **4.** |  |  |  |  |
| **5.** |  |  |  |  |
| **6.** |  |  |  |  |
| **7.** |  |  |  |  |
| **8.** |  |  |  |  |
| **B. Instructors Activity** |  | | | |
| **1.** |  |  |  |  |
| **2.** |  |  |  |  |
| **3.** |  |  |  |  |
| **C. Students Activity** |  | | | |
| **1.** |  |  |  |  |
| **2.** |  |  |  |  |
| **3.** |  |  |  |  |
| **Total Completion Time** |  |  |  |  |

# 6.3.3 Post-Evaluation Procedures:

After completing the co-operative evaluation, participants were given a post-test questionnaire to fill in, which is shown in Appendix B. This questionnaire was important to capture their thoughts and feelings about Advance KASIT while they were still fresh. The questionnaire was then followed by a short interview and discussion, which mainly focused on the initial modified design of the Advance KASIT. Table (19) shows the responses of the 3 participants to the post-test questionnaire.

Table 6.11 : Participants Responses to the Post-Test Questionnaire.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Statement** | **Participant 1** | **Participant 2** | **Participant 3** | **Average** |
| **1** | Is the system stable? | 5 | 4 | 4 | **4.3** |
| **2** | Is the system ease of use? | 5 | 5 | 5 | **3** |
| **3** | Are the functionality of the system achieve user’s needs? | 5 | 4 | 5 | **4.6** |
| **Average** | | **5** | **4.3** | **4.6** |  |

# System Installation:

**6.4.1 Apache XAMPP Server For Windows:**

<https://www.apachefriends.org/index.html>.

* + 1. **Adobe Dreamweaver CC:**

[**https://www.adobe.com/**](https://www.adobe.com/)

* + 1. **Operating System (Microsoft Windows 10):**

<http://windows.microsoft.com/en-us/windows/downloads>

* + 1. **Microsoft Office 2019:**

<http://www.microsoftstore.com/store/msusa/en_US/cat/All-Office/categoryID.69403900>

* + 1. **Microsoft Office Visio 2019:**

<https://products.office.com/en/visio/flowchart-software>

# 6.5 Summary:

This chapter showed an intestine testing and evaluation for Advance KASIT. The heuristic evaluation was conducted system with 3 expert users. The Heuristic and cooperative evaluation have also shown competitive and acceptable performance for the system indicating that system is easy to use and has less usability problems.

# 7.0 Project Conclusion and Future Work

# 7.1 Introduction:

The aim of this system considers under the following notes:

- Facilitating and accelerating the process of finding the qualified student who teaches a certain course.

- Participate in the process of making our faculty a (smart faculty).

- A chance for students with same interests to build relationships that maybe will lead them to cooperate in useful efforts in future.

# 7.2 Overall Weaknesses:

We can’t be sure that the new volunteered students are qualified enough to teach a certain course due to reviews, as well as we may not find volunteered students for all courses.

# 7.3 Overall Strengths:

We expect from our system to provide a suitable online environment that correlate students, increase the communication among them, and increase the expertise for both volunteered and regular students through a web platform that allow students to sign up and create their profiles and fill them with their personal as well as educational information

# 7.4 Future Work:

As future work, our website portal it could be transfer to Android and IOS mobile application on.

# 7.5 Summary:

In this chapter we written weaknesses and strengths in order with hope to reduce the weaknesses and focus on future work.

# 8.0 Tools & reference

**Tools we used :**

1. Creately.
2. Google Form.
3. Visual studio code .
4. Microsoft Word.
5. Note Pad.
6. Star UML.
7. Xampp

**Reference:**

1. E-Learning (project seior-2).

**Format of the documentation:**

1. Font Type: Times New Roman (Headings CS)Font Size: 12 & 14 points.
2. Spacing: Double Spacing.
3. Horizontal Alignment: Justify.
4. Font Color: Black
5. Page Bolder & Page Number.

# APPENDIX A

**Heuristic Evaluation – A System Checklist**

Disclaimer: This list is a simplified one of the original list which was developed by Xerox Corporation (© Usability Analysis & Design, Xerox Corporation, 1995) and was downloaded from  [ftp://cs.uregina.ca/pub/class/305/lab2/example-he.html](file:///C:\Users\user\AppData\Local\Temp\Temp1_Documentation.zip\Documentation\%20ftp:\cs.uregina.ca\pub\class\305\lab2\example-he.html). It has been simplified to suite the purpose it is used for, which is to evaluate the Advance KASIT in order to identify current problems as experienced by the users, which is part of our graduation project that is submitted to King Abdullah II School for Information Technology, The University of Jordan. The number of questions was reduced; however, the individual questions were left intact.

Please fill in the evaluation form below, which is a form of checklist, by writing “X” in the appropriate place which mostly describes the best answer to the corresponding criterion. This form is to be filled after you have investigated the system interface i.e. have looked at, and examined the interface. The answer to each criterion is either:

* "**0**" which means “I don’t agree that this is a usability problem at all”.
* "**1**" which means “Cosmetic problem only: need not be fixed unless extra time is available on project”.
* "**2**" which means “Minor usability problem: fixing this should be given low priority”.
* "**3**" which means ”Major usability problem: important to fix, so should be given high priority”.
* "**4**" which means ”Usability catastrophe: imperative to fix this before product can be released".

Thank you for your willingness to evaluate this system. Your time and effort are highly appreciated.

**H1. Visibility of System Status**

The system should always keep user informed about what is going on, through appropriate feedback within reasonable time.

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Review Checklist** | **0 1 2 3 4** | **Comments** |
| 1.1 | Does every display begin with a title or header that describes screen contents? | ( ) ( ) ( ) ( ) ( ) |  |
| 1.2 | Do menu instructions, prompts, and error messages appear in the same place(s) on each menu? | ( ) ( ) ( ) ( ) ( ) |  |
| 1.3 | Is there some form of system feedback for every operator action? | ( ) ( ) ( ) ( ) ( ) |  |
| 1.4 | Are responses times appropriate to the users cognitive processing? | ( ) ( ) ( ) ( ) ( ) |  |
| 1.5 | Is there visual feedback in menus or dialog boxes about which choices are selectable? | ( ) ( ) ( ) ( ) ( ) |  |

**H2. Match between System and the Real World**

The system should speak the user’s language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Review Checklist** | **0 1 2 3 4** | **Comments** |
| 2.1 | Are icons concrete and familiar? | ( ) ( ) ( ) ( ) ( ) |  |
| 2.2 | Are menu choices ordered in the most logical way, given the user, the item names, and the task variables? | ( ) ( ) ( ) ( ) ( ) |  |
| 2.3 | Do related and interdependent fields appear on the same screen? | ( ) ( ) ( ) ( ) ( ) |  |
| 2.4 | When prompts imply a necessary action, are the words in the message consistent with that action? | ( ) ( ) ( ) ( ) ( ) |  |
| 2.5 | On data entry screens, are tasks described in terminology familiar to users? | ( ) ( ) ( ) ( ) ( ) |  |

**H3. User Control and Freedom**

Users should be free to select and sequence tasks (when appropriate), rather than having the system does this for them. Users often choose system functions by mistake and will need a clearly marked “emergency exit” to leave the unwanted state without having to go through an extended dialogue. Users should make their own decisions (with clear information) regarding the costs of exiting current work. The system should support undo and redo.

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Review Checklist** | **0 1 2 3 4** | **Comments** |
| 3.1 | When a user's task is complete, does the system wait for a signal from the user before processing? | ( ) ( ) ( ) ( ) ( ) |  |
| 3.2 | Are users prompted to confirm commands that have drastic, destructive consequences? | ( ) ( ) ( ) ( ) ( ) |  |
| 3.3 | Are character edits allowed in data entry fields? | ( ) ( ) ( ) ( ) ( ) |  |
| 3.4 | If menu lists are long (more than seven items), can users select an item either by moving the cursor or by typing a mnemonic code? | ( ) ( ) ( ) ( ) ( ) |  |
| 3.5 | If the system uses a pointing device, do users have the option of either clicking on menu items or using a keyboard shortcut? | ( ) ( ) ( ) ( ) ( ) |  |

**H4. Consistency and Standards**

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Review Checklist** | **0 1 2 3 4** | **Comments** |
| 4.1 | Has a heavy use of all uppercase letters on a screen been avoided? | ( ) ( ) ( ) ( ) ( ) |  |
| 4.2 | Are icons labeled? | ( ) ( ) ( ) ( ) ( ) |  |
| 4.3 | Are there no more than twelve to twenty icon types? | ( ) ( ) ( ) ( ) ( ) |  |
| 4.4 | Does each window have a title? | ( ) ( ) ( ) ( ) ( ) |  |
| 4.5 | Is vertical and horizontal scrolling possible in each window? | ( ) ( ) ( ) ( ) ( ) |  |
| 4.6 | Are menu choice lists presented vertically? | ( ) ( ) ( ) ( ) ( ) |  |
| 4.7 | Are menu titles either centered or left-justified? | ( ) ( ) ( ) ( ) ( ) |  |
| 4.8 | Are menu items left-justified, with the item number or mnemonic preceding the name? | ( ) ( ) ( ) ( ) ( ) |  |
| 4.9 | Do embedded field-level prompts appear to the right of the field label? | ( ) ( ) ( ) ( ) ( ) |  |
| 4.10 | Are attention-getting techniques used with care? | ( ) ( ) ( ) ( ) ( ) |  |

**H5. Help Users Recognize, Diagnose, and Recover From Errors**

Error messages should be expressed in plain language (NO CODES).

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Review Checklist** | **0 1 2 3 4** | **Comments** |
| 5.1 | Is sound used to signal an error? | ( ) ( ) ( ) ( ) ( ) |  |
| 5.2 | Are error messages worded so that the system, not the user, takes the blame? | ( ) ( ) ( ) ( ) ( ) |  |
| 5.3 | Do error messages suggest the cause of the problem? | ( ) ( ) ( ) ( ) ( ) |  |
| 5.4 | Do error messages indicate what action the user needs to take to correct the error? | ( ) ( ) ( ) ( ) ( ) |  |
| 5.5 | If the system supports both novice and expert users, are multiple levels of error-message detail available? | ( ) ( ) ( ) ( ) ( ) |  |
| 5.6 | If an error is detected in a data entry field, does the system place the cursor in that field or highlight the error? | ( ) ( ) ( ) ( ) ( ) |  |
| 5.7 | Do error messages inform the user of the error's severity? | ( ) ( ) ( ) ( ) ( ) |  |

**H6. Error Prevention**

Even better than good error messages is a careful design which prevents a problem from occurring in the first place.

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Review Checklist** | **0 1 2 3 4** | **Comments** |
| 6.1 | Are menu choices logical, distinctive, and mutually exclusive? | ( ) ( ) ( ) ( ) ( ) |  |
| 6.2 | Are data inputs case-blind whenever possible? | ( ) ( ) ( ) ( ) ( ) |  |
| 6.3 | Does the system prevent users from making errors whenever possible? | ( ) ( ) ( ) ( ) ( ) |  |
| 6.4 | Does the system warn users if they are about to make a potentially serious error? | ( ) ( ) ( ) ( ) ( ) |  |
| 6.5 | Do data entry screens and dialog boxes indicate the number of character spaces available in a field? | ( ) ( ) ( ) ( ) ( ) |  |
| 6.6 | Do fields in data entry screens and dialog boxes contain default values when appropriate? | ( ) ( ) ( ) ( ) ( ) |  |

**H7. Recognition Rather Than Recall**

Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Review Checklist** | **0 1 2 3 4** | **Comments** |
| 7.1 | For question and answer interfaces, are visual cues and white space used to distinguish questions, prompts, instructions, and user input? | ( ) ( ) ( ) ( ) ( ) |  |
| 7.2 | Are inactive menu items grayed out or omitted? | ( ) ( ) ( ) ( ) ( ) |  |
| 7.3 | Do data entry screens and dialog boxes indicate when fields are optional? | ( ) ( ) ( ) ( ) ( ) |  |
| 7.4 | Are prompts, cues, and messages placed where the eye is likely to be looking on the screen? | ( ) ( ) ( ) ( ) ( ) |  |
| 7.5 | Are field labels close to fields, but separated by at least one space? | ( ) ( ) ( ) ( ) ( ) |  |
| 7.6 | Have items been grouped into logical zones, and have headings been used to distinguish between zones? | ( ) ( ) ( ) ( ) ( ) |  |
| 7.7 | Are borders used to identify meaningful groups? | ( ) ( ) ( ) ( ) ( ) |  |
| 7.8 | Is color coding consistent throughout the system? | ( ) ( ) ( ) ( ) ( ) |  |

**H8. Flexibility and Minimalist Design**

Accelerators-unseen by the novice user-may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. Provide alternative means of access and operation for users who differ from the “average” user (e.g., physical or cognitive ability, culture, language, etc.)

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Review Checklist** | **0 1 2 3 4** | **Comments** |
| 8.1 | If menu lists are short (seven items or fewer), can users select an item by moving the cursor? | ( ) ( ) ( ) ( ) ( ) |  |
| 8.2 | If the system uses a pointing device, do users have the option of either clicking on fields or using a keyboard shortcut? | ( ) ( ) ( ) ( ) ( ) |  |
| 8.3 | On data entry screens, do users have the option of either clicking directly on a field or using a keyboard shortcut? | ( ) ( ) ( ) ( ) ( ) |  |
| 8.4 | On menus, do users have the option of either clicking directly on a menu item or using a keyboard shortcut? | ( ) ( ) ( ) ( ) ( ) |  |
| 8.5 | In dialog boxes, do users have the option of either clicking directly on a dialog box option or using a keyboard shortcut? | ( ) ( ) ( ) ( ) ( ) |  |

**H9. Aesthetic and Minimalist Design**

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Review Checklist** | **0 1 2 3 4** | **Comments** |
| 9.1 | Are all icons in a set visually and conceptually distinct? | ( ) ( ) ( ) ( ) ( ) |  |
| 9.2 | Does each icon stand out from its background? | ( ) ( ) ( ) ( ) ( ) |  |
| 9.3 | Does each data entry screen have a short, simple, clear, distinctive title? | ( ) ( ) ( ) ( ) ( ) |  |
| 9.4 | Are field labels brief, familiar, and descriptive? | ( ) ( ) ( ) ( ) ( ) |  |
| 9.5 | Are there pop-up or pull-down menus within data entry fields that have many, but well-defined, entry options? | ( ) ( ) ( ) ( ) ( ) |  |

**H10. Help and Documentation**

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user’s task, list concrete steps to be carried out, and not be too large.

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Review Checklist** | **0 1 2 3 4** | **Comments** |
| 10.1 | Are on-line instructions visually distinct? | ( ) ( ) ( ) ( ) ( ) |  |
| 10.2 | If menu choices are ambiguous, does the system provide additional explanatory information when an item is selected? | ( ) ( ) ( ) ( ) ( ) |  |
| 10.3 | Is the help function visible; for example, a key labeled help or a special menu? | ( ) ( ) ( ) ( ) ( ) |  |
| 10.4 | Navigation: Is information easy to find? | ( ) ( ) ( ) ( ) ( ) |  |
| 10.5 | Presentation: Is the visual layout well designed? | ( ) ( ) ( ) ( ) ( ) |  |
| 10.6 | Conversation: Is the information accurate, complete, and understandable? | ( ) ( ) ( ) ( ) ( ) |  |
| 10.7 | Is the information relevant? | ( ) ( ) ( ) ( ) ( ) |  |
| 10.8 | Can users easily switch between help and their work? | ( ) ( ) ( ) ( ) ( ) |  |
| 10.9 | Is it easy to access and return from the help system? | ( ) ( ) ( ) ( ) ( ) |  |
| 10.10 | Can users resume work where they left off after accessing help? | ( ) ( ) ( ) ( ) ( ) |  |

# Appendix B

**Cooperative for Advance KASIT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task No.** | **Test** | **Time Taken to Complete the Task** | **Comments** |
| **A.** | **Administrator** | | |
| **1.** | Login |  |  |
| **2.** | Logout |  |  |
| **4.** | Courses Management |  |  |
| **5.** | Instructors Management |  |  |
| **6.** | Libraries Management |  |  |
| **7.** | Students Management |  |  |
| **8.** | Volunteers Courses Packages Management |  |  |
| **B.** | **Instructors** | | |
| **1.** | Login |  |  |
| **2.** | Logout |  |  |
| **3.** | Volunteers Courses Packages Management |  |  |
| **C.** | **Students** | | |
| **1.** | Login |  |  |
| **2.** | Logout |  |  |
| **3.** | Volunteers Courses Packages Management |  |  |

# APPENDIX C

**Advance KASIT Usability Test (Post-test Questionnaire)**

**Gender:** M / F **Age:** \_\_\_\_\_

**Educational Level:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Programmer Taken:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Institution:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

After using the system and answering the Advance KASIT test, please indicate the extent to which you agree or disagree with each of the following statements regarding to your experience with the system.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Statements** | **Strongly Disagree** | **Disagree** | **Neutral** | **Agree** | **Strongly Agree** |
| **1** | **2** | **3** | **4** | **5** |
| **1.** | Is the app stable? |  |  |  |  |  |
| **2.** | Is the app ease of use? |  |  |  |  |  |
| **3.** | Are the functionality of the app achieve user’s needs? |  |  |  |  |  |

# Questionnaire Result

Chart, pie chart

Description automatically generatedThe questionnaire was filled out **by 11 people**, who shared their suggestions:  
Questionnaire link : https://forms.gle/chq7EomzZA45RR6t9

Chart, pie chart

Description automatically generatedChart, pie chart

Description automatically generatedA picture containing chart

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated