**Quiz Game Application**

**High Level Design & Low Level Design**

The purpose of this document is to provide a template for documenting both HLD & LLD.

**Document Control :**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Revision History** | | | | | | | | |
|  |  |  | |  |  |  |  |  |
| **Date** | **Version** | **Author** | **Brief Description of Changes** | | | | **Approver Signature** | |
| 24-12-2022 | QGA\_DesignDocument-v0.2 | 1. Gorantla Saranya | Changes in quiz round 2 fucntion | | | | Kishwar Sayed | |
| 24-12-2022 |  | 2. Potta Lakshmi Ravali | Changes in admin function | | | | Kishwar Sayed | |
|  |  | 3.Om Sri Amrutha Varshini Alla | Changes in coordinator function. | | | | Kishwar Sayed | |
| 24-12-2022 |  | 4.Vudumula Tejaswi | Done changes in contestant registration | | | | Kishwar Sayed | |
|  |  | 5. Vanapalli Bhargavi | Changes in quiz round1 function | | | | Kishwar Sayed | |
| 24-12-2022 |  | 6. Jyothi Swarupa Repalle | Changes in contestant login function | | | | Kishwar Sayed | |

[1. 1 Introduction……………………………………………………………………………….5](#_Toc122382953)

[1.1 Intended Audience 5](#_Toc122382954)

[1.2 Acronyms/Abbreviations 5](#_Toc122382955)

[1.3 Project Purpose 6](#_Toc122382956)

[1.4 Key Project Objective 6](#_Toc122382957)

1.5 Project Scope and Limitations…………………………………………………………………………………………………….6

1.5.1 In Scope……………………………………………………………………………………………………………………………….6

[1.6 Functional Overview 7](#_Toc122382966)

[1.7 Assumptions, Dependencies & Constraints 7](#_Toc122382967)

[1.8 Risks 7](#_Toc122382968)

[2 Design Overview 8](#_Toc122382969)

[The system consists of three entities: admin , coordinator and client. 8](#_Toc122382970)

[2.1 Design Objectives 8](#_Toc122382971)

[2.1.1 Recommended Architecture 9](#_Toc122382972)

[2.2. Architectural Strategies 9](#_Toc122382973)

[2.2.1 Reuse of Existing Common Services/Utilities 9](#_Toc122382975)

[2.2.2 User Interface Paradigms 9](#_Toc122382977)

[2.2.3 System Interface Paradigms 9](#_Toc122382979)

*2.2.4 Error Detection/ Exceptional Handling………………………………………………………………10*

*2.2.5 Memory Management……………………………………………………………………………………10*

[2.2.6 Performance 10](#_Toc122382980)

[2.2.7 Security 10](#_Toc122382981)

[2.2.8 Housekeeping and Maintenance 10](#_Toc122382983)

[3.QUIZ GAME FLOW CHART AND ER DAIGRAM 11](#_Toc122382984)

[3.1. Flow Chart Diagram 11](#_Toc122382985)

[3.2. ER Diagram 12](#_Toc122382986)

[3.2.1. Internal Interfaces 13](#_Toc122382989)

[3.2.2. External Interfaces 13](#_Toc122382990)

[4. Detailed System Design 13](#_Toc122382991)

[4.1. Key Entities 13](#_Toc122382992)

[5. Environment Description 14](#_Toc122382995)

[5.1. Time Zone Support 14](#_Toc122382996)

[5.2. Language Support 14](#_Toc122382997)

[5.3. User Desktop Requirements 14](#_Toc122382998)

[5.3.1. Deployment Considerations 14](#_Toc122383000)

[5.3.2. Application Server Disk Space 15](#_Toc122383001)

[6 Configuration 15](#_Toc122383010)

[6.1 Operating System 15](#_Toc122383006)

[6.2 Database 15](#_Toc122383007)

[6.3 Desktop 16](#_Toc122383009)

**7 REFERENCES………………………………………………………………………………...**16

[8 Conclusion 16](#_Toc122383011)

# **1 Introduction**

This ‘Quiz Game’ Project is designed for a question in which you can generate and manage a simple database for questions. The question number is automatically generated by the software and is stored in a binary file by “Question”. This database is basically used as a MASTER file to be used as a lookup table for information like quiz option, sub menu, either computer/mathematics/science or general and the return. In this software you can ask for any menu on the basis of the menu question will be asked to the user. There will be 2 rounds i.e warmup and challenge round. In first round there’ll be 5 questions and the user must be able to answer at least 3 out of 5 questions to enter the second round. In second round there’ll be total of 10 questions and the user get 20 seconds for answering each question. If the user is correct then he get 10 points on every question. If he is incorrect then he lose the game and the game is closed.

In past days quiz is conducted manually but in further resolution of the technology we are able to generate the score automatically.

According to this a user can play an interactive quiz without the need of having a book and searching various topics. He/she simply needs to open the game and test enhances the abilities.

## 1.1 Intended Audience

This document is created by developers, testers, project

manager and contestants. This is a technical document, and the terms should be

understood by all of them.

## 1.2 Acronyms/Abbreviations

|  |  |
| --- | --- |
| QGA | Quiz Game Application |

## 

## 1.3 Project Purpose

* The sole intention behind the consideration of this project is to generate and manage a simple database for question.
* This project is developed considering “Quiz” information keep in context of the contestant in mind. Here, data is stored in a binary file “Question”.
* The main objective of “The Quiz Game” is to facilitate a user friendly environment for all users and reduces the manual effort.

## Key Project Objective

## To reduce workload of staff.

## To reduce the delay in processing time.

## To reduce the delay in records updation.

## To provide the user-friendliness in all possible ways.

## To provide greater flexibility.

## Project Scope and Limitation

The Quiz Game Application in C programming helps users to gain knowledge without any manual effort. It also does a few other tasks, it will provide an easy interface where users can check the result of the last played game. Another choice is if the user left the quiz in the middle of the game, he can continue the game whenever he wants after relogging in.

Limitation: Technology is not always reliable, sometimes the system may breakdown.

* + 1. **In Scope**

The scope of this project very broad in terms of gaining knowledge and sharing knowledge among world.

Few of them are:

* Can be used anywhere any time as it is a web based application.
* This application will be used in educational institutions.

## Functional Overview

In this quiz game, we can store the username and password, view the highest score secured by a user, and even reset the score. Additionally, to make the game look a little more interesting, it is divided into 2 rounds. User must pass the first round to reach the second one.

Of the 2 rounds mentioned above, the first is called the Warm-Up Round; the second is the Challenge Round. In the warm-up round, the user is asked a total of five questions and they must be able to answer atleast three of them correctly to enter the next round. If the user is not capable of doing that, he is not permitted to proceed further.

In the second and more interesting round of this quiz game in C, the user will be asked 10 questions. For each question asked, there are 4 options, namely A, B, C and D. There are no negative markings. Player must answer the question within the timer limit of 20 seconds for each question.

The functions that are used for admin part is mentioned below:

* *admin():* This function definition is used to login the game as a admin. He’ll manage the coordinator and contestant.

The functions that are used for coordinator part is mentioned below:

* *coordinator():* This function is used to login as a coordinator and to manage quiz.
* *add\_coordinator()*: This function is used to create a new coordinator.
* *update\_coordinator():* This function is used to update the coordinator.
* *delete\_coordinator()*: This function is used to delete a coordinator.

The functions that are used for contestant part is mentioned below:

* *contestant():* This function is used to call the menu of the contestant.
* *contestant\_registration:* This function is used for registration of a new user.

The following functions are used for quiz part of the game:

* *play\_quiz():* This function is used to start a new quiz.
* *continue\_quiz():* This function is used to continue the previous quiz.
* *quiz\_question():* This function is used to store the questions.
* *get\_score():* This function is used to get the scores of contestant.
* *rules():* This function stores the guidelines of the quiz game.

## Assumptions, Dependencies & Constraints

Assumption, dependency & constraints - system doesn’t require any network connection. We have to store the credentials of the users

in a file safely. And we also have to store the website list in a separate file. User must be able to check the scores even after re-login to the application.

## Risks

Risk - the QGA itself provides authentication & security. No one can see the user's credentials that are stored in a separate file. There are also risks associated with the hackers & malicious insiders & careless employees.

# **2 Design Overview**

# 

# The system consists of three entities: admin, coordinator and a contestant.

## 2.1 Design Objectives

* + The QGA enables admin to manage the details of coordinator and contestant and provides an user-friendly interface.
  + The user can start the program whenever they want.
  + The user can play the game without internet.

### 2.1.1Recommended Architecture

**Client-side hardware interface**:

● Ubuntu/Linux machine

● Terminal

**Tools to be used:**

* gcc compiler
* Valgrind.
* Splint.

## 2.2. Architectural Strategies

The architectural strategy used in this project is an admin -admin can

manages the details of the coordinator and contestant like name, password, and

user id. Coordinator manages the details of contestant and quiz i.e update

/delete/add quiz. User strategy. This strategy consists of two entities, the admin

and the user. The user enters the destination and it will provide output to the

### 2.2.1 Reuse of Existing Common Services/Utilities

Quiz game application by using ubuntu/linux.

### 2.2.2 User Interface Paradigms

Linux/Ubuntu machine

gcc complier

**2.2.3 System Interface Paradigms**

* Operating system : LINUX/Ubuntu
* Linux Kernel version : GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86\_64
* Bash shell: 5.0.16(1)-release

### 2.2.4 Error Detection / Exceptional Handling

By using file handling we are detecting the errors and handle it using the

conditional statements. We have used valgrind for error detection and exceptional handling. We have integrated the project again and again to resolve the errors. We must manage exceptional handling for quiz functionalities. We have used the validation for exception handling.

### 2.2.5 Memory Management

Malloc() and calloc() functions are used to manage memory. The calloc() function is generally more suitable and efficient than the malloc() function. While both functions are used to allocate memory dynamically, calloc() can allocate multiple blocks at a single time.

There are 2 ways in which the memory is getting managed:

1. Static
2. Dynamic

The variables which are assigned in the function occupies the size according to their datatype. While we used linked list to manage the details of the actors, so according to the actors, the size is occupied dynamically.

### 2.2.6 Performance

To make the application run faster and smoother the code has to be written in optimized manner. The performance is based on the configuration of the system.

**2.2.7 Security**

Admin will store the credentials in a file no one can see. Users need to login by using their credentials. We have to provide authentication to access these files and also, we have to secure the data in the directory.

### 2.2.8 Housekeeping and Maintenance

It serves as a crucial tool in tracing the destination and the completion of particular initiatives.

To allow the admin to make updates and patches for the site and to maintain the quality of application.

# **Quiz Game Flow Chart & ER Diagram:**

## Flow Chart Diagram

Flow Chart Diagram is a diagrammatic representation of data movement through a system-manual or automated from inputs to outputs through processing. Flow chart helps in the analysis of the flow of data through a system and thus help identifying the system requirements.

Diagram

Description automatically generated

## Entity Relationship Diagram

The entity Relationship Diagram(ERD)depicts the relationship between the data objects. The ERD is the notation that is used to conduct the date modeling activity the attributes the attributes of each data object noted is the ERD can be described resign a data object descriptions. The primary purpose of the ERD is to represent data objects and their relationships. Diagram

Description automatically generated

### Internal Interfaces

The internal interfaces comprise interfaces through which the system interacts with the clients through which it provides them services.

### External Interfaces

The external interface comprises interfaces through which the users interact with the system.

* Desktop or Linux Machine

# **Detailed System Design**

The Quiz game contains a admin, coordinator & contestant. The admin manages coordinator and contestant details. The coordinator can updates his/her details and manages quiz. Contestant after registering/login he can play quiz or can see score board and also can reset score.

## Key Entities

Key Entities are associated with the systems are:-

**Admin:-**

The admin creates coordinator lists and stores them in files. An admin will also manages details of coordinator and contestant.

**Coordinator:-**

The coordinator entity will login with the credentials given by admin.

**Contestant:-**

The contestant entity can register or login into the application and he/she play game.

# **Environment Description**

The Environment description allows contestant to connect or play

quiz. Quiz is an interactive game that allows students to explore different choices and teaches the considerations and helps students to build the general knowledge.

Linux is a Unix-like, open source and community-developed operating system(OS) for computes, servers, mainframes, mobile devices and embedded devices. It is supported on almost every major computer platform, making it one of the most widely supported operating systems. GCC compiler is used to compile and run a C program on the Linux OS.

## Time Zone Support

It will support time zones as per Indian standard time(IST) in (GMT +5:30) and UST standard.

## Language Support

We are using C programming language and gcc compilation. We can

specify the Linux commands to do that task. C programming is a general-purpose, procedural, imperative computer programming language developed in 1972 by Dennis M. Ritchie at the Bell Telephone Laboratories to develop the UNIX OS. You can already run C code because Linux systems include the GNU c library. To write and build it, all you need is to install a compiler, open a terminal, and start coding

## User Desktop Requirements

User desktop requires a Linux environment, Operating system of Linux or Ubuntu. Internet connection is required to download Linux/ Ubuntu. Terminal/ command prompt also required.

### Deployment Considerations

Deployment considerations are,

● 500Mhz Processor

● 120GB HDD CPU

● 4GB RAM

### Application Server Disk Space

Disk Space - Less space is required.

### Configuration

### Operating System

Linux is an opensource OS. The OS sits between applications and hardware and makes the connections between all your software and the physical resources that do the work.

* 1. **Database**

File handling is used which refers to the method of storing data in the C program in the form of an output or input that might have been generated while running a C program in a data file.

**6.3 Desktop**

* Operating system – Linux / Ubuntu.
* RAM - 8GB.
* Processor - i3/i5/i7
* Linux or Ubuntu-like environments are required.

# **7 References**

**1.**[**Quiz game application - GeeksforGeeks**](https://www.geeksforgeeks.org/traceroute-in-network-layer/)

**2.https://www.geeksforgeeks.org/quiz-game-in-e-commerce/**

**3.Git hub**

# **8 CONCLUSION**

This quiz application provides facility to play quiz anywhere and anytime. It save time since user.

**Change Log**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **QMS Template Version Control (Maintained by QA)** | | | | | |
|  |  |  |  |  |  |
| **Date** | **Version** | **Author** | | **Description** | |
| 20-12-2022 | 1.0 | QA Team | | Initial Version | |
|  |  |  | |  | |
|  |  |  | |  | |
|  |  |  | |  | |