A PROJECT REPORT

on

**Quiz App**

*Submitted by*

**Mr. Ganesh Umesh Tiwari**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF SCIENCE**

in

**COMPUTER SCIENCE**

*under the guidance of*

**Prof. Vipul Saluja**

**Department of Computer Science**



**R. D. & S.H. National College & S. W. A. Science College**

**Bandra, Mumbai – 400050.**

**(Sem V)**

**(2018 – 2019)**

**DECLARATION**

I, Mr. Ganesh Umesh Tiwari, hereby declare that the project entitled “**Quiz App**” submitted in the partial fulfillment for the award of **Bachelor of Science** in **Computer Science** during the academic year **2018 – 2019** is my original work and the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

**Signature of the Student:**

**Place:**

**Date:**

Success is my only option, failures not.

-Marshall Bruce Mathers III

|  |
| --- |
| **PRELIMINARY INVESTIGATION** |

1.1 SYNOPSIS

**Introduction: -**

Quiz is a Educational app which provides users with educational true or false or Multiple Choice Questions based upon their choice. They also have an option to select their favourite subjects from a list of preincluded subjects.

**Modules: -**

1. MCQ Quiz
2. Boolean Quiz
3. Settings
4. Send Suggestions

**Database Modules: -**

1. mcqQuestions
2. booleanQuestions
3. users
4. scores

1.2 Organizational Overview

QuizApp is a Educational app which provides users with easy to use ui and a very user friendly interface so that they can test their knowledge on a platform on which they would love to do that. Quiz App not only provides users with mcq as well as boolean questions for their thirst of knowledge they can choose from a range of ever growing subjects which currently include 7 subjects namely: - science, history, geography, films, sports, mythology and computer science.

* Users can play quiz
* Users can see detailed history about their performance in earlier games
* Users can compare themselves from players all around the world.

1.3 Proposed System

* The proposed system allows user to play quiz games in mcq as well as boolean format.
* Works on REST API framework so it modular and database can be changed without affecting application codebase.
* Has a Nice, Catchy UI as well as a easy to use interface so that even a newcomer will understand the flow of Application.

1.4 Tools and Technologies to be used.

1. Flutter SDK.
2. Visual Studio Code.
3. Android SDK.
4. Dart Language.
5. Eclipse.
6. Jersey Rest Framework.
7. Heroku Web Platform.
8. MySQL Server.
9. Php.
10. OpenJdk 11.

**Flutter SDK: -**

Flutter is an open-source mobile application development SDK created by Google. It is used to develop applications for Android and iOS, as well as being the primary method of creating applications for Google Fuchsia.

**Visual Studio Code: -**

Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is also customizable, so users can change the editor's theme, keyboard shortcuts, and preferences. It is free and open-source, although the official download is under a proprietary license.

**Android SDK: -**

The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows 7 or later.

**Dart Language: -**

Dart is a general-purpose programming language originally developed by Google and later approved as a standard by Ecma (ECMA-408). It is used to build web, server, and mobile applications. Dart is an object-oriented, class defined language using a C-style syntax that transcompiles optionally into JavaScript. It supports interfaces, mixins, abstract classes, reified generics, static typing, and a sound type system.

**Eclipse: -**

Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE.[6] It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins, including Ada, ABAP, C, C++, C#, COBOL, D, Fortran, Haskell, JavaScript, Julia,[7] Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala, Clojure, Groovy, Scheme, and Erlang.

**Jersey Rest Framework: -**

Jersey RESTful Web Services framework is an open source framework for developing RESTful Web Services in Java. It provides support for JAX-RS APIs and serves as a JAX-RS (JSR 311 & JSR 339) Reference Implementation.

**Heroku Web Platform: -**

Heroku is a cloud platform as a service (PaaS) supporting several programming languages. Heroku, one of the first cloud platforms, has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go. For this reason, Heroku is said to be a polyglot platform as it lets the developer build, run and scale applications in a similar manner across all the languages. Heroku was acquired by Salesforce.com in 2010 for $212 million.

**MySQL Server: -**

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

**PHP: -**

PHP (recursive acronym for PHP: Hypertext Pre-processor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

**OpenJDK 11**: -

OpenJDK (Open Java Development Kit) is a free and open-source implementation of the Java Platform, Standard Edition (Java SE). It is the result of an effort Sun Microsystems began in 2006. The implementation is licensed under the GNU General Public License (GNU GPL) version 2 with a linking exception. Were it not for the GPL linking exception, components that linked to the Java class library would be subject to the terms of the GPL license. OpenJDK is the official reference implementation of Java SE since version 7.

|  |
| --- |
| 1.5 Feasibility Study. |

A feasibility study is an analysis of how successfully a project can be completed.

It is the initial design stage of any project, which brings together the elements of knowledge.

All activities of feasibility study are directed towards helping answer the question

"Should we proceed with the proposed project idea?"

"Does the proposed system contribute to the overall objectives for which the system was proposed for?"

**Technical feasibility: -**

Technical feasibility is concerned with specifying the equipment’s and the software to satisfy the user requirements.

The aim of technical feasibility is to support the cost of the company to undertake a technical study into: -

The system is very much feasible with its technical aspect as there is not much computing resource required to build the system.

Making changes in the system regarding updating product details can be easily done as the admin will have a complete understanding of the system's content and the tools which are used for developing the system.

**Operational feasibility: -**

In operational feasibility, we attempt to ensure that every user can access the system easily.

The ease to use the system will help to increase the operational importance of the system, as there will be not much computing expertise required to use the system and a person with minimum computing knowledge can use the system very effectively.

The proposed system will really benefit the organisation as the system could be maintained by the admin itself and there will not be requirement for any additional staff for maintaining the system.

The overall response of the system will also increase as there will be a greater number of users affiliated with the system in the near future.

**Economic feasibility: -**

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system.

The proposed system can be developed at a minimum cost and resource.

The system can assure a good beneficial cost to the organisation.

The savings that would arise from the beneficial cost of the system can be used to improve the system's performance in future.

|  |
| --- |
| Chapter 2. System Analysis |

* Event Table
* Entity Relationship Diagram
* Object Diagram
* Use Case Diagram
* Activity Diagram
* State Chart Diagram
* Sequence Diagram

|  |
| --- |
| 2.1 Event Table |

Events are objects or messages used when a software component wants to notify a state change to other components.

An Event model is a software architecture (a set of classes and interfaces) that determines how components occur.

On the event source side: -

• create and describe events

• trigger (or fire) events

• distribute events to interested components

On the event listener side: -

• subscribe to event sources

• react to events when received

• remove the subscription to event sources when desired

Terminology often used refers to: -

• Event Source or Provider: - the sender of events

• Event: - the object sent

• Event Listener or Event Sink or Consumer: - the receiver of events

User: -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Event** | **Trigger** | **Source** | **Response** | **Destination** |
| User Plays Game | Chooses subject after choosing to play | User | Display Score | Score Page |
| User Changes Audio Setting | Click on Settings in Options Page | User | Change Audio Setting | Audio Setting Page |
| User Views Personal Score | Click on Personal Score in Options Page | User | See Personal Scores | Personal Score Page |
| User Views Leader board. | Click on Leader Board in Options Page | User | Display Leader boards | Leader boards Page |
| User Suggests Question | Click on suggest question in Options Page | User | Submitted Snackbar Shown | Suggest Question Page |

Admin: -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Event** | **Trigger** | **Source** | **Response** | **Destination** |
| Admin Logs In | Submit Login Form | Admin | Confirmed as admin | Admin |
| Adds/ Removes/ Updates Question | Add/ Delete/ Update Button | Admin | Added/ Removed/ Updated Question | Admin |
| Add/ Remove/ Update User Details | Add/ Delete/ Update Button | Admin | Added/ Removed/ Updated User Details | Admin |
| Add/ Remove Suggestion | Add/ Delete Suggestion | Admin | Added/ Removed Suggestion | Admin |
| Add/ Remove/ Update Score | Add/ Remove/ Update Score | Admin | Added/ Removed/ Updated User Score | Admin |

|  |
| --- |
| 2.2 Entity Relationship Diagram |

In software engineering, an entity relationship model (ER model) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that leads itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities (things) and the relationships that can exist among them.

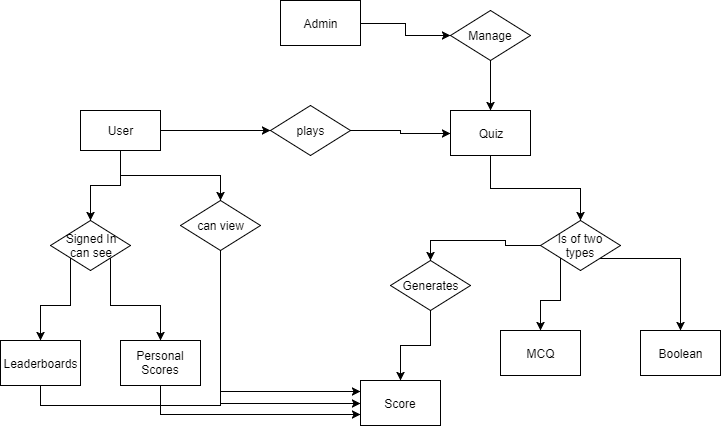
An entity- relationship model is a systematic way of describing and defining a business process. The process is modelled as components (entities) that are linked with each other by relationships that express the dependencies and requirements between them, such as: one building may be divided into zero or more apartments, but one apartment can only be located in one building. Entities may have various properties (attributes) that characterize them. Diagram created to represent these entities, attributes and relationships graphically are called entity -relationship diagrams.

An ER model is typically implemented as a database. In the case of a relational database, which stores data in tables, every row of each table represents one instance of an entity. Some data fields in these tables point to indexes in other tables; such pointers represent the relationship.

Limitations: -

• ER models assume information content that can readily be represented in a relational database. They describe only a relational structure for this information.

• They are inadequate for systems in which the information cannot readily be represented in relational form, such as with semi-structured data



|  |
| --- |
| 2.3 Class Diagram |

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualising, describing and documenting different aspects of a system but also for constructing executable code of the software application

The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages.

The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram.

**Purpose**: -

The purpose of the class diagram is to model the static view of an application. The class diagrams are the only diagrams which can be directly mapped with object oriented languages and thus widely used at the time of construction.

The UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application but class diagram is a bit different. So, it is the most popular UML diagram in the coder community.

So, the purpose of the class diagram can be summarised as: -

* Analysis and design of the static view of an application.
* Describe responsibilities of a system.
* Base for component and deployment diagrams.
* Forward and reverse engineering.

|  |
| --- |
| 2.4 Object Diagram |

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams.

Object diagrams represent an instance of class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment.

Object diagrams are used to render a set of objects and their relationships as an instance.

**Purpose**: -

The purpose of a diagram should be understood clearly to implement it practically. The purpose of object diagrams are similar to class diagrams.

The difference is that a class diagram represents an abstract model consisting of classes and their relationships. But an object diagram represents an instance at a particular moment which is concrete in nature.

It means the object diagram is closer to the actual system behaviour. The purpose is to capture the static view of a system at a particular moment.

So, the purpose of the object diagram can be summarised as: -

* Forward and reverse engineering
* Object relationships of a system
* Static view of an interaction
* Understand object behaviour and their relationship from practical perspective.

|  |
| --- |
| 2.5 Use Case Diagram |

To model a system the most important aspect is to capture the dynamic behaviour. To clarify a bit in details, dynamic behaviour means the behaviour of the system when it is running/operating.

So only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. So, use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

So, to model the entire system numbers of use case diagrams are used.

Purpose: -

The purpose of use case diagram is to capture the Dayna aspect of a system. But this definition is too generic to describe the purpose.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So, when a system is analysed to gather its functionalities use cases are prepared and actors are identified.

Now when the initial task is complete use case diagrams are modelled to present the outside view.

So, in brief the purpose of use case diagram can be as follows: -

• Used to gather requirements of a system.

• Used to get an outside view of a system.

• Identify external and internal factors influencing the system.

• Show the interacting among the requirements are actors.

