VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

Quiz Application

Submitted in partial fulfillment of the requirements for the VII Semester of degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi

Submitted by

SHARANYA J

1RN18IS096

Under the Guidance of

Mr. R Rajkumar

Associate Professor

Department of ISE

Department of Information Science and Engineering

RNS Institute of Technology

Channasandra, Dr. Vishnuvardhan Road, R R Nagar Post

Bengaluru – 560 098

2021-2022

RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road, RR Nagar Post,

Bengaluru – 560 098

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

Certified that the Internship work entitled ***Quiz Application*** has been successfully completed by **Sharanya J(1RN18IS096)** student of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements of 7th semester for the award of degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year **2021-2022**. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

Mr. Rajkumar	**Dr. Suresh L**	**Dr. M K Venkatesha**
Internship Guide	Professor and HoD	Principal, RNSIT
Associate Professor	Assistant Professor	
Department of ISE	Department of ISE	

External Viva

Name of the Examiners	**Signature with date**
1. _____	_____
2. _____	_____

DECLARATION

I, SHARANYA J[1RN18IS096] student of VII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Internship work entitled ***Quiz Application*** has been carried out by me and submitted in partial fulfillment of the requirements for the *VII Semester degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi* during academic year 2021-2022.

Place: Bengaluru

Date :10/01/2022

SHARANYA J

(1RN18IS096)

ABSTRACT

The Quiz Application provides a user friendly, interactive system to take online test in an efficient manner and no wasting time for checking the paper. The main objective of the application is to efficiently evaluate the user thoroughly through a fully automated system that not only saves lot of time but also gives fast results. The system carries out the examination and auto-grading for multiple choice questions which is fed into the system.

The project is built using Flutter, which is an open-source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase. The Flutter framework consists of both a software development kit (SDK) and their widget-based UI library. This library consists of various reusable UI elements, such as sliders, buttons, and text inputs.

This application consists of multiple-choice questions, with selectable choices for each question. The score is internally calculated for each question based on the choice selected. The correct answer is displayed in case an incorrect option is chosen. This is a timed quiz, where a time frame is given for each question and the questions have to be answered in the given time. The score is displayed at the end of the quiz.

ACKNOWLEDGMENT

At the very onset I would like to place our gratefulness to all those people who helped me in making the Internship a successful one.

Coming up, this internship to be a success was not easy. Apart from the sheer effort, the enlightenment of the very experienced teachers also plays a paramount role becauit is they who guided me in the right direction.

First of all, I would like to thank the **Management of RNS Institute of Technology** for providing such a healthy environment for the successful completion of internship work.

In this regard, I express sincere gratitude to our beloved Principal **Dr. M K Venkatesha,** for providing us all the facilities.

We are extremely grateful to our own and beloved Professor and Head of Department of Information science and Engineering, **Dr. Suresh L**, for having accepted to patronize me in the right direction with all her wisdom.

We place our heartfelt thanks to **Dr. Mamatha G** Professor, Department of Information Science and Engineering for having guided internship and all the staff members of the department of Information Science and Engineering for helping at all times.

I thank **Mr. Akshay DR, Director, Enmaz Engineering Services Private Limited**, for providing the opportunity to be a part of the Internship program and having guided me to complete the same successfully.

I also thank our internship coordinator **Dr. R Rajkumar,** Associate Professor, Department of Information Science and Engineering. I would thank my friends for having supported me with all their strength and might. Last but not the least, I thank my parents for supporting and encouraging me throughout. I have made an honest effort in this assignment.

for supporting and encouraging me throughout. I have made an honest effort in this assignment.

SHARANYA J

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ARM	Advanced RISC Machines
BLOC	Business Logic Component
GPU	Graphics Processing Unit
IDE	Integrated Development Environment
iOS	iPhone Operating System
SDK	Software Development Kit
UI	User Interface
VM	Virtual Machine

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Background

The Quiz Application provides a user friendly, interactive system to take online test in an efficient manner and no wasting time for checking the paper. The main objective of the application is to efficiently evaluate the user thoroughly through a fully automated system that not only saves lot of time but also gives fast results. The system carries out the examination and auto-grading for multiple choice questions which is fed into the system.

1.1.1 History

Flutter is an open-source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase. Released in 2017, Flutter allows developers to build mobile applications for both iOS and Android with a single codebase and programming language. This capability makes building iOS and Android apps simpler and faster. The Flutter framework consists of both a software development kit (SDK) and their widget-based UI library. This library consists of various reusable UI elements, such as sliders, buttons, and text inputs.

The first version of Flutter was known by the codename "Sky" an ran on the Android operating system. During the keynote of Google Developer Days in Shanghai in September 2018, Google announced Flutter Release Preview 2, which is the last big release before Flutter 1.0. On December 4th of that year, Flutter 1.0 was released at the Flutter Live event, denoting the first "stable" version of the Framework. On December 11, 2019, Flutter 1.12 was released at the Flutter Interactive event. On March 3, 2021, Google released Flutter 2 during an online Flutter Engage event. On September 8th, 2021, the Dart SDK in version 2.14 and Flutter version 2.5 were released by Google. The update brought improvements to the Android Full-Screen mode and the latest version of Google's Material Design. The major components of Flutter include: Dart platform, Flutter engine, Foundation library, Design specific widgets, Flutter Development tools.

1.2 Existing System

There are loads of hard copied documents being generated. This brings us to the age-old discussion of keeping information in the form databases versus keeping the same on sheets of paper. Keeping the information in the form of hard-copied documents leads to the following problems.

Drawbacks of existing system:

- Lack of space - It becomes a problem in itself to find space to keep the sheets of paper being generated as a result of the ongoing discussion.
- Filing poses a problem - Filing the documents categorically is a time consuming and tedious exercise.
- Filtering is not easy - It becomes hard to filter relevant documents for the irrelevant ones if the count of the same crosses a certain manageable number.
- Reviewing becomes time-consuming - All the process done manually at the centres. The Existing system is paper based, time consuming, monotonous, less flexible and provides a very hectic working schedule. The chance of loss of records is high and also record searching is difficult. Maintenance of the system is also very difficult and takes lot of time
- Result Processing- is slow due to paper work and requirement of staff.

1.3 Proposed System

Quiz Application is developed to overcome the time-consuming problem of manual system. This application will check for correct answers and save the examiner's time. The aim here is to computerize the manual system and help in saving time and data. This application conducts a quiz in the form of multiple-choice questions. After the completion of the quiz the application generates test scores for the quiz taken.

This application consists of multiple-choice questions, with selectable choices for each question. The score is internally calculated for each question based on the choice selected. The correct answer is displayed in case an incorrect option is chosen. This is a timed quiz, where a time frame is given for each question and the questions have to be answered in the given time. The score is displayed at the end of the quiz.

Chapter 2

LITERATURE REVIEW

A Literature review describes the concept of how the concept has emerged, how it has been implemented and what is the current status.

In paper [1], the author discussed the importance of assessment and evaluation in the form of quizzes. Most of e-learning platforms provide supporting system to online evaluation and assessment among their features, and in many cases this supporting system include feature, which is automatically graded quiz based on answer that specified in a question's setting in quiz. Internet has made e-learning an alternative way that facilitates students' learning due to its fast growth.

In education, quizzing and testing have typically been used as evaluative tools to assess student performance. This orientation may change, however, with the increasing use of online quiz and its facilities through tools available at universities. The study presents the development and evaluation of online quizzes and investigates whether they could be useful tools to assist students' learning in university. The time limitation is necessity for making quiz as a tool for assessment and evaluation. By this particular tool, students can make self-assessment about their achievement in learning because the quizzes are automatically corrected and provide as grade for students.

Paper [2] discussed how Integrated Development Environments (IDEs) provide a convenient standalone solution that supports developers during various phases of software development. Integrated Development Environments (IDEs) are very popular among software developers since they provide support for many of their daily development or maintenance tasks. Modern IDEs provide integrated debuggers, automated refactoring, assistance tools like code completion, and even integrated version control. The focus in this paper is on Visual Studio IDE. The author instrumented the previously unexplored Visual Studio IDE and tracked the interactions of developers at an industry partner's software-development department. The captured events are transformed into high-level activities such as development, navigation, IDE configuration, and project management in order to identify how much time developers spend on each activity.

Paper [3] discussed about cross platform development using Flutter. Cross-platform mobile development today is full of compromise. Developers are forced to choose between either building the same app multiple times for multiple operating systems, or to accept a lowest common denominator solution that trades native speed and accuracy for portability. Flutter is an open-source SDK for creating high-performance, high- fidelity mobile apps for iOS and Android. Few important features of flutter are - Just-in-time compilation is a way of executing computer code that involves compilation during execution of a program – at run time – rather than prior to execution.

Flutter's hot reload feature helps you quickly and easily experiment, build UIs, add features, and fix bugs. Hot reload works by injecting updated source code files into the running Dart Virtual Machine (VM). After the VM updates classes with the new versions of fields and functions, the Flutter framework automatically rebuilds the widget tree, allowing you to quickly view the effects of your changes.

Paper [4] discussed about the dilemma of developer - either develop multiple native applications for different operating systems or developing one app that is cross-platform compatible. Many technologies for creating cross-platform applications have emerged over the years, and new technologies are released every year. One such technology is Flutter, which is a mobile application SDK (Software Development Kit). Flutter promises the ability to build native applications on iOS and Android that achieve native performance.

Two visually identical mobile applications were developed. One application was developed using the Flutter SDK and one using the native Android SDK. The applications were then evaluated by users who were asked about their perception of using the applications. Users were also asked if they preferred any of the applications. Users perceived the speed of the native application to be faster than the speed of the Flutter application, but perceived the appearance of both applications to be equal. Users perceived the appearance of the apps to be similar with a satisfaction score of 5.1 for the Flutter app and 5.2 for the native app.

The quality of cross-platform development technologies is constantly improving, and Flutter is an interesting new alternative. It is a technology that makes it easy for developers to create native-looking apps that can be executed on both the Android and iOS platform.

Paper [5] discussed application development using Flutter. In Flutter, every application is written with the help of Dart. Google has developed and maintained a programming language called Dart. It is extensively used inside Google and it has been verified to have the proficiency to develop enormous web applications, such as AdWords. Originally Dart was developed to replace and succeed JavaScript. Thus, it implements most of the important

characteristics of JavaScript's next standard (ES7), such as the keywords "async" and "await". Flutter uses a high-performance rendering engine to render each view component using its own. This provides a chance to build applications that are as high-performance as native applications can be. Flutter targets the top mobile operating systems like Android and iOS, it gives you a solution for GPU rendering and UI, powered by native ARM code.

Chapter 3

ANALYSIS

3.1 Introduction

The Quiz Application provides a user friendly, interactive system to take online test in an efficient manner and no wasting time for checking the paper. The main objective of the application is to efficiently evaluate the user thoroughly through a fully automated system that not only saves lot of time but also gives fast results. The system carries out the examination and auto-grading for multiple choice questions which is fed into the system.

3.1.1 Purpose

This Application provides facility to conduct online examination worldwide. It saves time and displays the results as the test gets over, so no need to wait for the result. It is automatically generated by the server. Administrator has a privilege to create, modify and delete the test papers and its particular questions.

3.1.2 Scope

The Scope of this project is very broad in terms of other manually checking yourself. Few of them are:

- This can be used in educational institutions as well as in corporate world.
- Can be used anywhere any time as it is a web-based application (user Location doesn't matter).
- No restriction that examiner has to be present when the candidate takes the quiz

3.2 Description

3.2.1 User Characteristics

Users of the product must know how to access the functionality of this system and get benefited.

Educational level: Users should be comfortable with the English Language.

Experience: Users should have prior information regarding the online Quizzes.

Skills: Users should have basic knowledge and should be comfortable using general purpose applications on computers.

3.2.2 Assumptions and Dependencies

We assume that the users of our application have a minimal knowledge of computer system and should have an availability of Internet. We are dependent on the sources from where we have gathered data and the data are authenticated.

3.3 Requirements

3.3.1 Hardware Requirements

The processor used is Intel(R) Core (TM) i5-8265U CPU @ 1.60GHz. The capacity of Random-Access Memory (RAM) is 4GB. The capacity of the storage element of disk space is 2.99GB. The monitor used is HDMI monitor. The keys available in the keyboard is 104 keys.

3.3.2 Software Requirements

Operating System: Windows 10

Tools: Flutter

Platform or IDE: Visual Studio Code

Network: Internet connection required

Chapter 4

SYSTEM DESIGN

4.1 Flutter Architecture

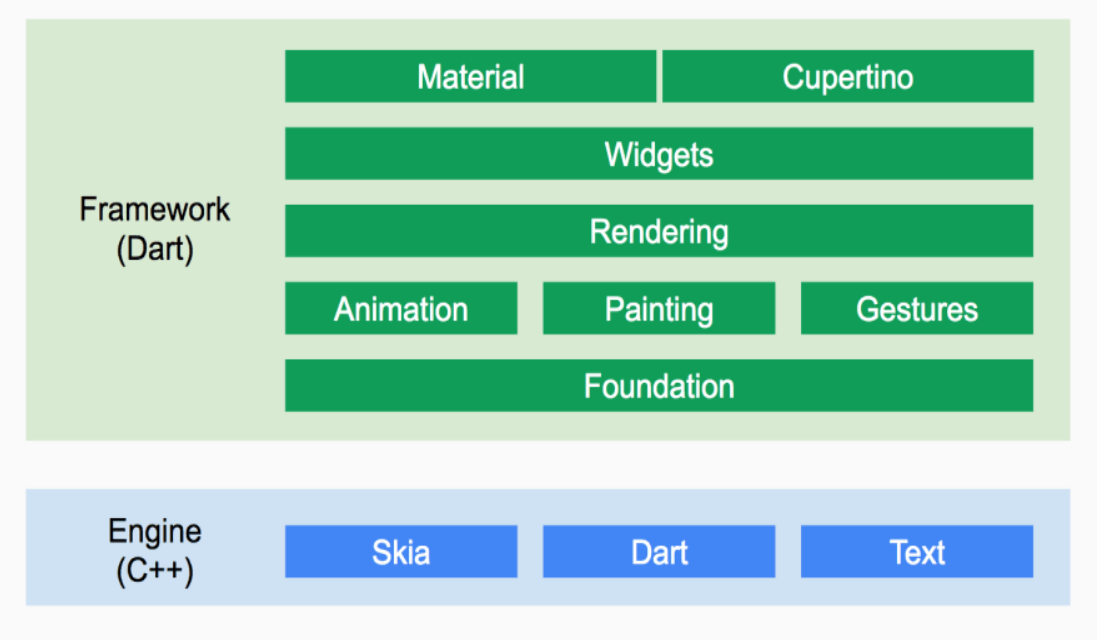


Figure 4.1 Flutter Architecture

Flutter is organized around layers. Each layer is built upon the previous. From the diagram, we can see the low-level part of Flutter is an Engine built in C++. It provides low level rendering support using Google's Skia graphics library. The high-level part of the diagram is the Framework written in Dart.

It provides libraries to handle animation, gestures, rendering, widgets and more. With all this layer the developer can do more with less code by using elements on the top or go down to customize some behavior of its app.

Everything is a widget

In Flutter, everything is a widget nested inside another widget. It comes with beautiful, customizable widgets and we can control the behavior of each widget and also styling becomes easy.

All the widgets of Flutter app form a hierarchy where a widget is a composition of other widgets and each widget inherits properties from its parent.

4.1.1 Basic Widgets

Flutter comes with a suite of powerful basic widgets, of which the following are commonly used.

- Text - The Text widget displays a string of text with single style. The string might break across multiple lines or might all be displayed on the same line depending on the layout constraints.
- Row - A widget that displays its children in a horizontal array. The design of these objects is based on the web's flexbox layout model.
- Column - A widget that displays its children in a vertical array. The design of these objects is based on the web's flexbox layout model.
- Image - A widget that displays an image.
- Icon - This widget acts as a container for storing the Icon in the Flutter.
- Scaffold - This widget provides a framework that allows you to add common material design elements like AppBar, Floating Action Buttons, Drawers, etc.
- Button - This widget allows you to perform some action on click. Flutter does not allow you to use the Button widget directly; instead, it uses a type of buttons like a FlatButton and a RaisedButton**.**

4.1.2 Material Design

Material is an adaptable design system, backed by open-source code, that helps developers easily build high-quality, digital experiences. From design guidelines to developer components, Material can help developers build products faster.

- App structure and navigation – Consists of AppBar, BottomNavigationBar, MaterialApp, Scaffold, TabBar, TabbarView.
- Buttons –Consists of ButtonBar, DropdownButton, FlatButton, FloatingActionButton, IconButton, RaisedButton.
- Input and Selection – Consists of Checkbox, Radio, Slider, Switch.
- Dialogues, alerts and panels – Consists of AlertDialog, BottomSheet, ExpansionPanel.
- Information Display – Consists of Card, Icon, Image, Chip.
- Layout – Consists of Divider, ListTile, Stepper.

4.2 Business Logic Component

Flutter delivers the basic architecture that you can apply to your application and manage its state easily. The architecture that is used in Flutter is called the Business Logic Component (BLOC). Basically, it is an event-state based approach that allows you to trigger events and handle state changes based on them. The BLOC is a good approach that separates your business logic from the user interface and oversees business logic key points by testing. The core ideas that were used for BLOC architecture are simplicity, scalability, and testability, and all these goals were definitely achieved within the BLOC architecture.

Chapter 5

DETAILED DESIGN

5.1 Process Flow Designs

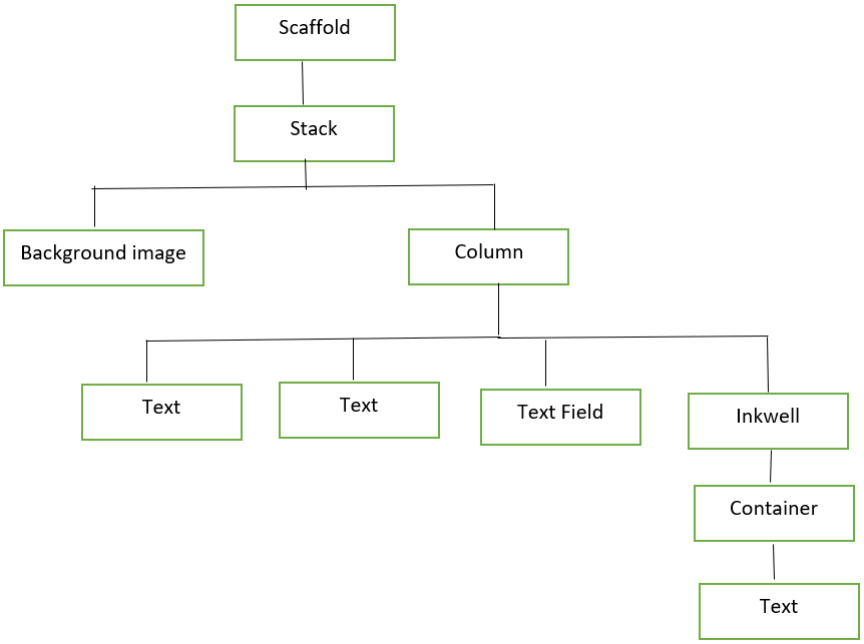


Figure 5.1.1 Widget tree for Welcome Screen

The Figure 5.1.1 shows the Widget tree for the Welcome Screen, which is the first screen of the application. The user is asked to enter the name, post which the Quiz questions are displayed one after the other. The Welcome Screen extends the StatelessWidget. The Scaffold widget contains the Stack widget. The Stack widget stacks together the background image and a Column widget, which in turns contains two text widgets, a TextField widget, and a Container widget of Text.

The Figure 5.1.2 shows the Widget tree for Score screen. The score obtained by the user in the Quiz is displayed in this screen. The Score screen extends the StatelessWidget. The Scaffold widget contains the Stack widget. Background image and a Column widget are stacked together in the Stack widget. The column widget in turn contains two Text widgets to display the scores.

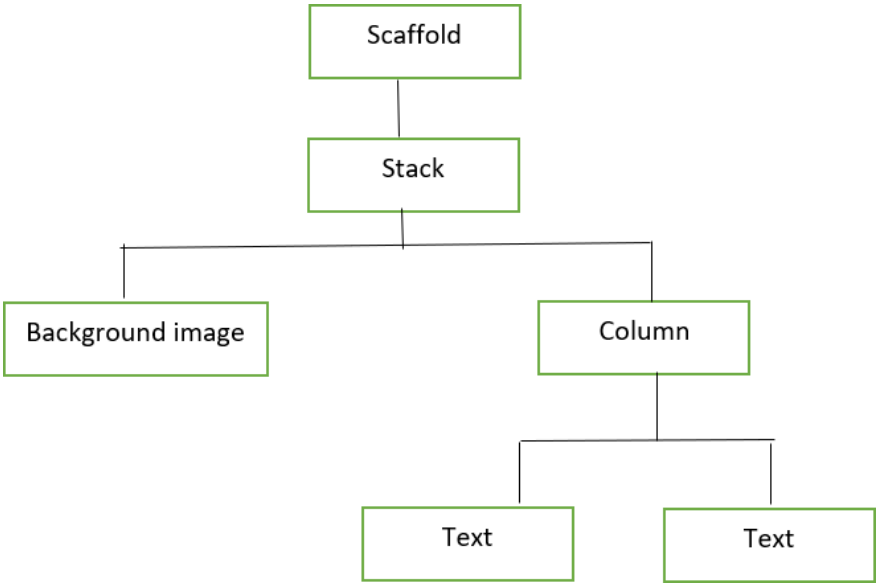


Figure 5.1.2 Widget Tree for Score Screen

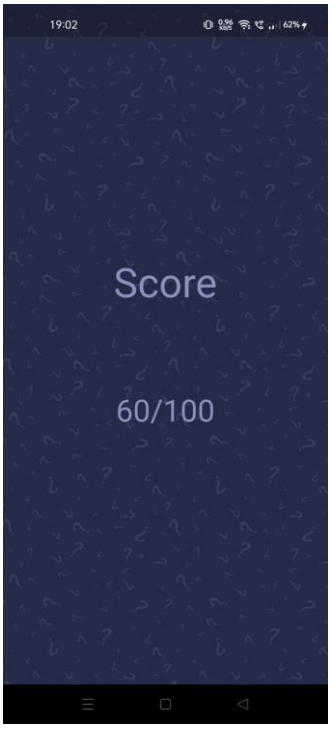


Figure 5.1.3 Screen built using widget tree for Score screen

The Figure 5.1.3 shows the Score screen built using the widget tree, which has a background image and Text fields.

The Questions for the quiz are given in the form of a List, which contains question ID, the question, suitable options for the question and the correct answer for the question.

```
const List sample_data = [
{
  "id": 1,
  "question":
    "Flutter is an open-source UI software development kit created by_",
  "options": ['Apple', 'Google', 'Facebook', 'Microsoft'],
  "answer_index": 1,
 },

{
  "id": 2,
  "question": "When did google release Flutter.",
  "options": ['Jun 2017', 'Jun 2017', 'May 2017', 'May 2018'],
  "answer_index": 2,
 },

{
  "id": 3,
  "question": "Which is the capital city of Karnataka. ",
  "options": ['Mysore', 'Chennai', 'Bangalore', 'Panaji'],
  "answer_index": 2,
 }
```

The Progress bar is included in the Quiz application which acts as a timer. The user has to answer a question in 60 seconds, otherwise the user is awarded zero mark for that question. The Progress bar also includes an option called ’Skip’ to skip a question and move to the next question.

After each question, there is a delay of 3 seconds given before the next question is displayed. Once the last question is displayed, the screen is redirected to the last screen Score screen.

Chapter 6

IMPLEMENTATION

Implementation is the process of defining how the system should be built, ensuring that it is operational and meets quality standards. It is a systematic and structured approach for effectively integrating a software-based service or component into the requirements of end users.

6.1 About Visual Studio Code

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion code refactoring and embedded Git

6.1.1 Features

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++. It is based on the Electron framework which is used to develop Node.js Web applications that run on the blink layout engine Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps . Visual Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, and debuggers, perform static code analysis, and add code linters using the Language Server Protocol

6.1.2 Language Support

Out of the box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace

6.1.3Version Control

Source control is a built-in feature of Visual Studio Code. It has a dedicated tab inside of the menu bar where you can access version control settings and view changes made to the current project. To use the feature you must link Visual Studio Code to any supported version control system (Git, Apache Subversion, Perforce, etc.). This allows you to create repositories as well as make push and pull requests directly from the Visual Studio Code program

6.2 About Flutter

Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, Mac, Windows, Fuchsia,Web, and the web from a single codebase.

6.2.1 History

The first version of Flutter was known by the codename "Sky" and ran on the Android operating system. It was unveiled at the 2015 Dart developer summit[6] with the stated intent of being able to render consistently at 120 frames per second. On December 4th of that year, Flutter 1.0 was released at the Flutter Live event, denoting the first "stable" version of the Framework. On December 11, 2019, Flutter 1.12 was released at the Flutter Interactive event.

On May 6, 2020, the Dart software development kit (SDK) in version 2.8 and the Flutter in version 1.17.0 were released, where support was added to the Metal API, improving performance on iOS devices (approximately 50%), new Material widgets, and new network tracking.

6.2.2 Widgets

Widgets are generally defined in three basic types: Stateful widgets, Stateless widgets, and Inherited widgets. Being the central class hierarchy in the Flutter framework the three basic types of widgets are used in the construction of every Flutter application. Although all the instances of a widget are immutable, the Stateful widget allows the interaction between user and application. By giving access to the method setState, the state can be maintained in separate state objects.

6.3 Pseudo code

6.3.1 main.dart

```
import 'package:flutter/material.dart';
import 'package:get/get_navigation/src/root/get_material_app.dart';
import 'package:quiz_app/screens/welcome/welcome_screen.dart';
void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
// This widget is the root of your application.
@override
  Widget build(BuildContext context) {
  return GetMaterialApp(
  title: 'Quiz App',
  debugShowCheckedModeBanner: false,
  theme: ThemeData.dark(),
  home: WelcomeScreen(),
);
}
```

This is the main file. The execution starts from main.dart file. The class MyApp extends Stateless Widget and the home property calls the Welcome Screen function.

6.3.2 constants.dart

```
import 'package:flutter/material.dart';
const kSecondaryColor = Color(0xFF8B94BC);
const kGreenColor = Color(0xFF6AC259);
const kRedColor = Color(0xFFE92E30);
const kGrayColor = Color(0xFFC1C1C1);
const kBlackColor = Color(0xFF101010);
const kPrimaryGradient = LinearGradient(
colors: [Color(0xFF46A0AE), Color(0xFF00FFCB)],
begin: Alignment.centerLeft,
```

```
end: Alignment.centerRight,
);
const double kDefaultPadding = 20.0
```

This file has a fixed set of colours and properties that are used throughout the application for various screens.

6.3.3 welcome_screen.dart

```
import 'package:flutter/material.dart';
import 'package:flutter_svg/svg.dart';
import 'package:get/get.dart';
import 'package:quiz_app/constants.dart';
import 'package:quiz_app/screens/quiz/quiz_screen.dart';
class WelcomeScreen extends StatelessWidget {
@override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Stack(
        children: [
          SvgPicture.asset("assets/icons/bg.svg", fit: BoxFit.fill),
          SafeArea(
            child: Padding(
              padding: const EdgeInsets.symmetric(horizontal: kDefaultPadding),
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  Spacer(flex: 2), //2/6
                  Text(
                    "Let's Play Quiz,",
                    style: Theme.of(context).textTheme.headline4.copyWith(
                        color: Colors.white, fontWeight: FontWeight.bold),
                  ),
                  Text("Enter your information below"),
                  Spacer(), // 1/6
```

```
TextField(
  decoration: InputDecoration(
    filled: true,
    fillColor: Color(0xFF1C2341),
    hintText: "Full Name",
    border: OutlineInputBorder(
      borderRadius: BorderRadius.all(Radius.circular(12)),
    ),
  ),
),
Spacer(), // 1/6
InkWell(
  onTap: () => Get.to(QuizScreen()),
  child: Container(
    width: double.infinity,
    alignment: Alignment.center,
    padding: EdgeInsets.all(kDefaultPadding * 0.75), // 15
    decoration: BoxDecoration(
      gradient: kPrimaryGradient,
      borderRadius: BorderRadius.all(Radius.circular(12)),
    ),
    child: Text(
      "Lets Start Quiz",
      style: Theme.of(context)
          .textTheme
          .button
          .copyWith(color: Colors.black),
    ),
  ),
),
Spacer(flex: 2), // it will take 2/6 spaces
],
),
),
```

```
    ),
      ],
    ),
  );
 }
}
```

The Welcome Screen is the first screen that is displayed when the application is opened. The Welcome Screen extends the StatelessWidget. The Scaffold widget contains the Stack widget. The Stack widget stacks together the background image and a Column widget, which in turns contains two text widgets, a TextField widget, and a Container widget of Text.

6.3.4 quiz-screen.dart

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:quiz_app/controllers/question_controller.dart';
import 'components/body.dart';
class QuizScreen extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
  QuestionController _controller = Get.put(QuestionController());
  return Scaffold(
   extendBodyBehindAppBar: true,
   appBar: AppBar(
    // Fluttter show the back button automatically
    backgroundColor: Colors.transparent,
    elevation: 0,
    actions: [
     FlatButton(onPressed: _controller.nextQuestion, child: Text("Skip")),
    ],
   ),
   body: Body(),
   );
```

```
  }
  }
```

6.3.5 Questions.dart

```
 class Question {
 final int id, answer;
 final String question;
 final List<String> options;
 Question({this.id, this.question, this.answer, this.options});
 }
 const List sample_data = [
 {
    "id": 1,
    "question":
    "Flutter is an open-source UI software development kit created by ___",
   "options": ['Apple', 'Google', 'Facebook', 'Microsoft'],
   "answer_index": 1,
 },
 {
    "id": 2,
    "question": "When did google release Flutter.",
    "options": ['June 2017', 'June 2017', 'May 2017', 'May 2018'],
    "answer_index": 2,
},
{
   "id": 3,
   "question": "Which is the capital city of Karnataka",
   "options": ['Mysore', 'Chennai', 'Bangalore', 'Panaji'],
   "answer_index": 2,
},
{
 "id": 4,
```

```
    "question": "Who is the current Prime Minister Of India",
    "options": [
      'Indira Gandhi',
      'Narendra Modi',
      'Manmohan Singh',
      'Rahul Gandhi'
    ],
    "answer_index": 1,
  },
];
```

The Questions for the quiz are given in the form of a List, which contains question ID, the question, suitable options for the question and the correct answer for the question. 10 such questions have been included in this application.

6.3.6 question_controller.dart

```
import 'package:flutter/widgets.dart';
import 'package:get/get.dart';
import 'package:get/state_manager.dart';
import 'package:quiz_app/models/Questions.dart';
import 'package:quiz_app/screens/score/score_screen.dart';
// We use get package for our state management
class QuestionController extends GetxController
    with SingleGetTickerProviderMixin {
 // Lets animated our progress bar
 AnimationController _animationController;
 Animation _animation;
  // so that we can access our animation outside
  Animation get animation => this._animation;
  PageController _pageController;
  PageController get pageController => this._pageController;
  List<Question> _questions = sample_data
  map(
      (question) => Question(
```

```
            id: question['id'],
             question: question['question'],
             options: question['options'],
             answer: question['answer_index']),
        )
        .toList();
    List<Question> get questions => this._questions;
    bool _isAnswered = false;
    bool get isAnswered => this._isAnswered;
    int _correctAns;
    int get correctAns => this._correctAns;
    int _selectedAns;
    int get selectedAns => this._selectedAns;
   // for more about obs please check documentation
   RxInt _questionNumber = 1.obs;
   RxInt get questionNumber => this._questionNumber;
   int _numOfCorrectAns = 0;
   int get numOfCorrectAns => this._numOfCorrectAns;
   // called immediately after the widget is allocated memory
  @override
  void onInit() {
   // Our animation duration is 60 s
   // so our plan is to fill the progress bar within 60s
   _animationController =
       AnimationController(duration: Duration(seconds: 60), vsync: this);
   _animation = Tween<double>(begin: 0, end: 1).animate(_animationController)
     ..addListener(() {
       // update like setState
       update();
     });// start our animation
   // Once 60s is completed go to the next qn
   _animationController.forward().whenComplete(nextQuestion);
   _pageController = PageController();
```

```
 super.onInit();
  }// // called just before the Controller is deleted from memory
  @override
  void onClose() {
    super.onClose();
    _animationController.dispose();
    _pageController.dispose();
  }
  void checkAns(Question question, int selectedIndex) {
    // because once user press any option then it will run
    _isAnswered = true;
    _correctAns = question.answer;
    _selectedAns = selectedIndex;
    if (_correctAns == _selectedAns) _numOfCorrectAns++; // It will stop the counter
    _animationController.stop();
    update();
    // Once user select an ans after 3s it will go to the next qn
    Future.delayed(Duration(seconds: 3), () {
    nextQuestion();
    });
  }
void nextQuestion() {
    if (_questionNumber.value != _questions.length) {
      _isAnswered = false;
      _pageController.nextPage(
          duration: Duration(milliseconds: 250), curve: Curves.ease);// Reset the counter
      _animationController.reset();// Then start it again
      // Once timer is finish go to the next qn
      _animationController.forward().whenComplete(nextQuestion);
    } else {
      // Get package provide us simple way to naviigate another page
      Get.to(ScoreScreen());
    }
  }
```

```
void updateTheQnNum(int index) {
_questionNumber.value = index + 1;
 }
 }
```

The Progress bar or the timer is animated and for every question, the timer runs till 60 seconds.

After each question, a delay of 3 seconds is given before moving to the next question. If answer selected by the user is same as the original answer fed into the program, the variable _numOfCorrectAns is incremented. At last, the Score screen function is called to display the scores.

6.3.7 score_screen.dart

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:quiz_app/constants.dart';
import 'package:quiz_app/controllers/question_controller.dart';
import 'package:flutter_svg/svg.dart';
class ScoreScreen extends StatelessWidget {
@override
Widget build(BuildContext context) {
 QuestionController _qnController = Get.put(QuestionController());
 return Scaffold(
  body: Stack(
   fit: StackFit.expand,
   children: [
    SvgPicture.asset("assets/icons/bg.svg", fit: BoxFit.fill),
    Column(
     children: [
      Spacer(flex: 3),
      Text(
       "Score",
       style: Theme.of(context)
          .textTheme
          .headline3
          .copyWith(color: kSecondaryColor),
```

```
            ),
            Spacer(),
            Text(
              "${_qnController.numOfCorrectAns * 10}/${_qnController.questions.length * 10}",
              style: Theme.of(context)
                  .textTheme
                  .headline4
                  .copyWith(color: kSecondaryColor),
            ),
            Spacer(flex: 3),
          ],
        )
      ],
    ),
  );
 }
}
```

The Score screen extends the StatelessWidget. The Scaffold widget contains the Stack widget. Background image and a Column widget are stacked together in the Stack widget. The column widget in turn contains two Text widgets to display the scores.

Chapter 7

TESTING

Testing is the process used to help identify the correctness, completeness, security and quality of the developed computer application. Testing is the process of technical investigation and includes the process of executing a program or application with the intent of finding errors. Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test.

7.1 Unit Testing

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. The unit testing is the process of testing the part of the program to verify whether the program is working correct or not. It usually has one or a few inputs and usually a single output. Unit testing is commonly automated, but may still be performed manually. The objective in unit testing is to isolate a unit and validate its correctness.

Table 7.1 Unit test case for Quiz Input Check

Case_ id	Description	Input data	Expected Ouput	Actual Ouput	Status
1	Opening the Quiz application after entering Name	Name is provided	The Quiz application opens and displays the questions	The Quiz application opens and displays the questions	Pass
2	Validating correct answers	Any option selected	Checks the answer for the respective question and validates it	Checks the answer for the respective question and validates it	Pass

3	Validating wrong answer	Any option selected	Checks the answer for the respective question and validates it	Checks the answer for the respective question and validates it	Pass
4	Skip a question	Select 'Skip' button	Shows the next question	Shows the next question	Pass
5	Exceeded time limit	No option selected	Skips to the next question	Skips to the next question	Pass
6	Closing application	-	Application should be closed without any error.	Application is closed	Pass

7.2 Integration Testing

Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

Table 7.2 Integration Testing for Quiz application

Case _id	Description	Input data	Expected Output	Actual Output	Status
1.	To display the correct answer of the Question	Select the option 3. from the options	Highlight the answer with Green colour	Highlight the answer with Green colour	Pass
2.	To display the wrong answer of the Question	Select the option 2. from the options	Highlight the answer with Red colour	Highlight the answer with Red colour	Pass
3.	Score added for correct answer of a question	Select the correct option for the question	10 marks added to Total score	10 marks added to Total score	Pass

4.	Score for an incorrect answer	Select the incorrect answer	No mark added or deducted	No mark added or deducted	Pass
5.	Score for a Skipped question	Press the 'Skip' button	No mark added or deducted	No mark added or deducted	Pass
6.	Score for a question where time limit exceeds	-	No mark added or deducted	No mark added or deducted	Pass
7.	Display test score	Complete all the questions in the quiz	Total marks displayed	Total marks displayed	Pass

7.3 System Testing

System Testing is a level of the software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. The application is run to check if all the modules (functions) can be executed concurrently, if each return correct results of the operations performed by them, and if the data and index files are left in consistent states by each module.

Table 7.3 System test case for Quiz application

Sl No. of test case:	1
Name of test:	Run test
Item / Feature being tested:	Question Details
Sample Input:	Choices entered in the order: 3 3 2 1
	Screens displayed (except menu screen) in order for: Welcome Page First Question displayed with the options Second Question displayed with the options Third Question displayed with the options Fourth Question displayed with the options Display the Total score for the attempted quiz
Actual output:	Screens are displayed in order
Remarks:	Test succeeded

Chapter 8

RESULTS

All the menu options provided in the application and its operations have been presented in as screen shots.

8.1 Welcome Page

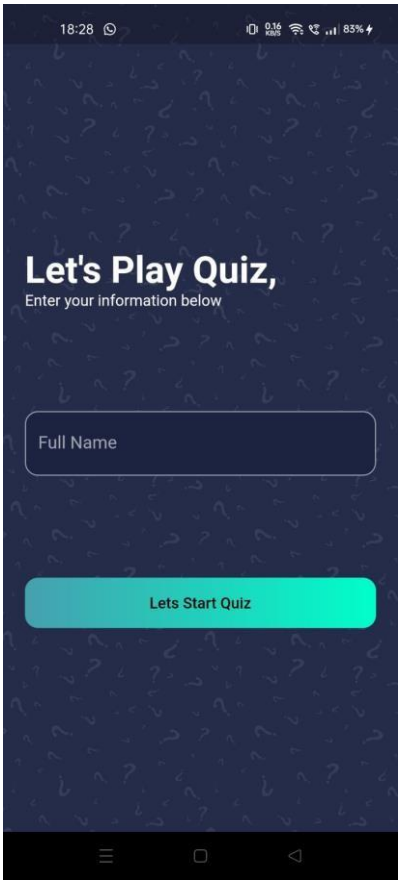


Figure 8.1.1 Starting Welcome Page.

The Figure 8.1.1 shows the Welcome screen of the application. This is the opening screen, where the user has to continue by providing the Name. On entering the application, the quiz questions are displayed.

The Figure 8.1.2 shows the name entered on the Welcome Page, which then leads to the page containing the Quiz questions.

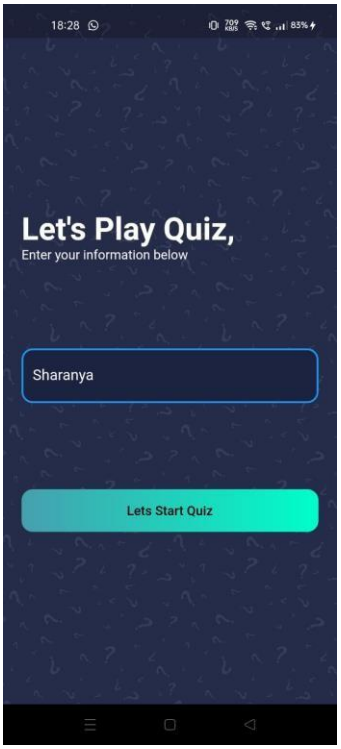
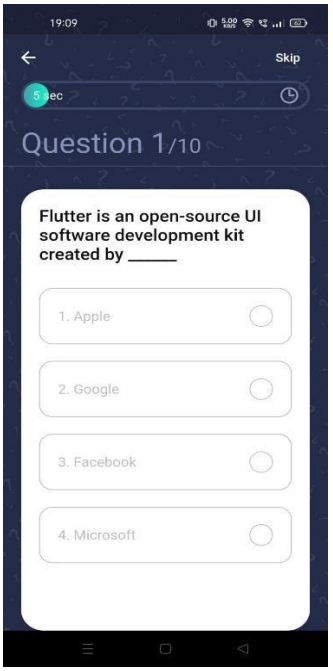


Figure 8.1.2 Welcome Screen with name entered

8.2 Question Page



The Figure 8.2 shows the first question displayed.

The screen displays the question with four options to select from. The timer keeps running for 60 seconds. If any of the given options is chosen, it checks for correct answer. If no option is selected, then after 10 seconds of timer, next question is displayed.

8.3 Correct Answer

The Figure 8.3 displays the screen when a correct option for a question is selected. The selected correct answer is highlighted with Green colour and a tick.

Figure 8.3 Correct answer

8.4 Wrong answer

The Figure 8.4 displays the Wrong Answer screen. When an incorrect option for a question is selected, the selected wrong answer is highlighted with a Red colour and the correct option which should have been selected is highlighted with Green colour. Here, option 2 is the incorrect answer whereas, option 3 is the correct answer for the question.

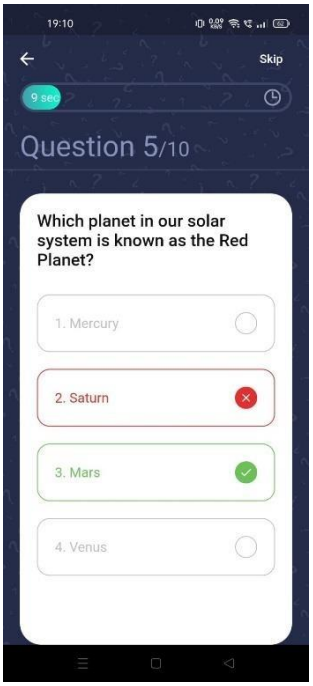


Figure 8.4 Wrong Answer for a Question

8.5 Test Score

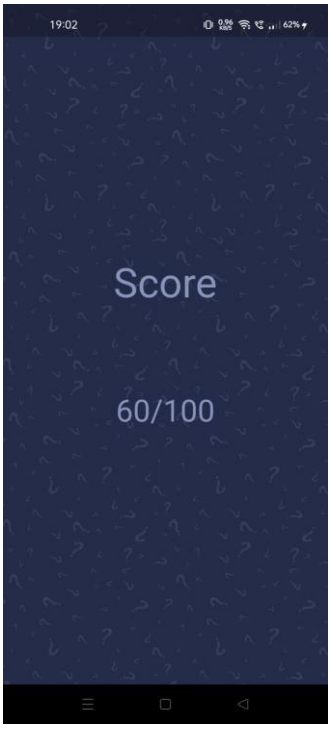


Figure 8.5 Quiz Score

The Figure 8.5 shows the total marks scored in the Quiz. Only six correct answers were selected, hence the score is 60 out of 100. The score in internally calculated and displayed. Here the user got six correct answers out of ten quiz questions, hence the score is 60 out of 100.

Chapter 9

CONCLUSION AND FUTURE ENHANCEMENTS

This internship project has been implemented with making appropriate effort to overcome the time-consuming problem of manual system. It has been successfully designed and implemented using Flutter. This this application will check the correct answer and save the examiner time and carry the examination in an effective manner. This application provides facility to play anywhere and anytime. It saves time as the user need not wait for the results. This system reduces paper work and can be used in events for conducting tests.

Future Enhancements

- In future work the system can be implemented with various quizzes of different topics.
- Concept of negative marking can be introduced.
- It can be possible to make it more user friendly by adding more variety of functions to it.
- A good Internet backup should be automated.
- Explanation of correct or wrong answer can be included
- Feedback from the user about the game

REFERENCES

[1] https://flutter.dev/

[2] https://code.visualstudio.com/

[3] https://stackoverflow.com/

Reference Papers:

Paper [1]: https://ieeexplore.ieee.org/document/7010585/references#references

Paper [2]: https://ieeexplore.ieee.org/document/7476636

Paper [3]: https://ijesc.org/upload/b3930ac14331fd1b425af8cd1c341d41.Cross%20Platform%20Development%20using%20Flutter%20(1).pdf

Paper [4]: https://www.diva-portal.org/smash/get/diva2:1480395/FULLTEXT01.pdf

Paper [5]: https://www.irjmets.com/uploadedfiles/paper/volume2/issue_8_august_2020/3180/1628083124.pdf