A Project Synopsis

Of

**Music Player App using Flutter and Dart Technology**

submitted in partial fulfilment of the requirements

for the award of the degree of

**Bachelor of Technology**

**Computer Science & Engineering**

By

Manish Kumar (191599010)

Pawan Tyagi (191599012)

Pawnesh kumar Pundhir(191599013)

Under the guidance of

**Mr. Manoj Varshney**

Department of Computer Engineering & Applications

Institute of Engineering & Technology



GLA UNIVERSITY

Mathura – 281406, India

**Feb, 2021**



**Department of computer Engineering and Applications**

**GLA University, Mathura**

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,**

**Mathura – 281406**

# DECLARATION

We hereby declare that the work which will be presented in the B.Tech. Project “**Music Player App using Flutter and Dart Technology**” in partial fulfilment of the requirements for the award of the Bachelor of Technology in Computer Science & Engineering of GLA University, Mathura, will be authentic record of our own work to be carried out under the supervision of **Mr. Manoj Varshney**, Assistant Professor, Institute of Engineering & Technology.

The contents of this project Synopsis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

**Sign** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Sign** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Name :** Manish Kumar **Name:** Pawan Tyagi

**University Roll No.:** 191599010  **University Roll No.:** 191599012

**Sign** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Name :** Pawnesh kumar Pundhir

**University Roll No.:** 191599013

**Date:**



**Department of computer Engineering and Applications**

**GLA University, Mathura**

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,**

**Mathura – 281406**

# CERTIFICATE

This is to certify that the above statements made by the candidate are correct and true to the best of my knowledge and belief.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Supervisor**

Mr. Manoj Varshney

Assistant Professor

Institute of Engineering & Technology

GLA University

# Problem Statement

In order to solve the problem of complex functions and large required memory of mobile phone music player on the current market, a new music player of simple, convenient, less required memory as well as user-friendly is developed. Based on the Android technology, using Flutter and Dart.

The new design mainly realizes six core functions including main play interface, playlists, menus, play settings, file browsing and song search. This player has merits of high performance, simple operation, and run independently on the Android mobile devices. At the same time, the player can also browse and access files in mobile phones.

**Objective**

In this project, our objective is to create a simple and interactive application so that it is easier to navigate through the application. Our main focus would be on User experience rather than UI.

**Feasibility Study**

It is feasible to add music player on the Android system from the aspects of economic, technical and social feasibility.

**Economic feasibility**

To design Android mobile phone music player as long as a computer has the Android development and the application development of Android is free. In addition, mobile phone music player is basic needs for public. The information that which functions are necessary form all the consumers , which functions are needed for some people, and which features are seldom to use is easy to understand. And a lot of research is eliminated, thus saved the spending. Therefore, the whole process of development doesn’t need to spend any money that is economic feasibility.

**Technical feasibility**

To design a music player which meets the basic requirements, a deep understand of Dart language, the Flutter framework, Firebase, the Android system architecture, application of framework and other technical knowledge are needed.(framework is the core of the application, and rules that all the programmers participating in the development must abide by). Based on the related technology information and resources for Android on the market, and equipped with technical personnel of technology and the spirit of willing to learn, the technology is feasible.

**Social feasibility**

With the rapid development of the mobile phone market, all kinds of audio and video resources are widely circulated on the Internet. These resources seem ordinary, but have gradually become an indispensable part of people life, which derived the development of all kinds of mobile phone player. But a lot of players devoted to fancy appearance, strong function causing a lot of wasted resources to the user's mobile phone and bringing a lot of inconvenience to the user as multitasking operation is needed. Some functions are useless to ordinary people. Powerful player is a good thing, but a lot of functions are actually useless for most users. Aimed at these problems, developing multiplied audio player which owns the features of simplified functions, common play function, meeting the needs of most users, less required memory and high quality of playing music, maximizes the optimization in performance.

**Future Scope**

This project can be further improved by

1. Implementing Machine Learning Algorithms to recommend songs as per the taste of user.
2. Implementing editing a song in real time like trimming.
3. Automating playlist making of user’s recent songs played.

**Methodology**

### Why Flutter?

Like in every cross-platform mobile technology, having a single code base can save the company’s resources and time by not requiring developers to write and maintain two separate versions (iOS and Android) of an application. From a customer’s point of view, this correlates to reduced expenses in making the product.

Flutter could also be a fast way of demonstrating a proof of concept or developing MVP, even if the finalized production version is eventually built using more traditional, native technologies. In the future, Flutter will also support web-frontend development. .   
Finally, the user interface of Fuchsia OS, a new operating system currently being developed by Google is built with Flutter SDK, so it is likely that this will be one of the technologies for building applications for Fuchsia, although it is hard to find information about it because of the lack of documentation.

### Building the user interface

In Flutter, the user interface consists of widgets. Everything is a widget and the UI is built by nesting these widgets.

Widgets in Flutter can be either stateful or stateless. The main difference with these two types of widgets is that the stateless widget is immutable and is only built once when the instance of this widget is created. The stateful widget on the other hand is mutable and will be rebuilt every time its state changes.

**Dart**

Flutter apps are using the Dart programming language. Dart is developed and maintained by Google. It was developed as a successor to Javascript and incorporates many of the features in the ES7 Javascript standard.

As mentioned before, Dart is compiled ahead of time. This allows for faster execution and avoids the Javascript bridge. The code still needs an interface to communicate with the platform code. This is however much faster than the Javascript bridge. Flutter also do not utilize the platform OEM widgets, reducing the number of times communication with the platform occurs.

Dart can also be compiled using a just in time compiler. This is done during development and allows for rapid development cycles. Flutter take advantage of the just in time compiler using a feature called hot reload. Hot reload allows code changes to be reloaded and injected on the device very fast. This allows for very efficient development of Flutter apps. The downside is that the app will be less efficient during development.

**Widgets**

Flutter doesn’t use the native OEM widgets, but instead provide its own widgets. The widgets are not rendered by the system. Instead, the app itself includes the renderer. Flutter only requires a canvas to render the widgets on the device screen. It also needs access to services (camera, location) and events (timers, touch events) .

This allows Flutter to decide when and how the widgets will be rendered. It also allows the widgets to be more customizable and extensible. The downside of this approach is that the app size increases, as it needs to host both the widgets and renderer within the app.

**Layout**

The layout is implemented differently in Flutter compared to other frameworks. The layout rules are what define the position and size of widgets that are rendered to the screen. The rules are moved from the application scope down to the widgets themselves. Position is controlled using position widgets. The position widgets are not visible on the screen. Its purpose is to control the position of another widget.

**Technologies to be used**

**Hardware**

* **Operating Systems:** Windows 7 or later
* **Processor**: Intel i3 or later
* **Ram**: 6 Gb or more
* **Disk Space:** 2 Gb

**Software**

* Flutter
* Dart Language
* Visual Studio Code.
* Emulator

**Testing Technology**

## Emulators, Simulators, or Real Devices

Real device mobile testing requires the use of the actual smartphone. But emulators and simulators are based on the concept of virtual testing. Virtual testing involves testing the software that provides the same functionality as provided by the real phone.

For testing mobile apps, an emulator is a desktop application that mimics the hardware and OS of the applications that should be tested. A simulator does not mimic the hardware/OS, but rather it mimics the basic behavior of a device. While simulators are usually simpler in function than emulators, they are not as useful as emulators. Real device testing checks the functionality of mobile apps thoroughly and ensures the appropriate working of the apps.

Emulators provide better results compared to simulators as they can be used to test specific situations or cases, and also can mimic multiple devices. Emulators are often used more as they are relatively cheaper – specifically if compared to real devices.

In this project we will be using Emulator as well as real device for testing purpose.

**Contribution of Project**

Our project will help people to just listen to their favourite songs without the hassle of navigating through tons of unused features.

**Scope for extension into a major project**

1. We can implement some advanced features like Machine Learning Algorithms to make the app smart and recommend users the songs of their liking.
2. Improve UI of the app to make it look more pleasing and attractive to users.
3. Enhance the User experience of the app to ease the way of interactive with the app.

**Conclusion**

Through the development of music player on Android platform, we get a clear understanding of overall process of the system. The core part of the music player is mainly composed of main interface, playlists, menus, play Settings, file browsing and song search. Grasping the development of the six parts, the music player has had the preliminary scale. Based on the function of the six categories, add some other small features.

Music player system realized the basic function of player: play, pause, and stop, up/down a, volume adjustment, lyrics display, play mode, song search, file browser, playlists query, and other functions.

This project will help us to understand the basics as well as some advanced concepts of application development.