

PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology (Deemed to be University)

Submitted in partial fulfillment of the requirements for
the award of Bachelor of Engineering Degree in
Computer Science and Engineering

By

D S DARSHAN
REG.NO. 39120019



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

**SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

NOVEMBER 2021



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by
NAAC



(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI– 600119
www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFAIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **D S DARSHAN** (Reg. No: **39120019**) who carried out the project entitled “**TO DO APPLICATION USING FLUTTER**” under my supervision from June 2021 to November 2021.

Internal Guide

DR.R. SUBHASHINI M.E., Ph.D.,

Internal Examiner External Examiner

External Examiner

DECLARATION

I, **D S DARSHAN** hereby declare that the project report entitled **TO DO APPLICATION USING FLUTTER** done by me under the guidance of **DR.R. SUBHASHINI** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE: 12/11/2021

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE:

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **DR. T. SASIKALA M.E., PH.D., Dean**, School of Computing, **DR.R. SUBHASHINI M.E., Ph.D.**, Head of the Department of Information technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **DR.R. SUBHASHINI**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **DEPARTMENT OF INFORMATION TECHNOLOGY** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE



Certificate

FLUTTER FRAMEWORK TRAINING

awarded to

Mr. Darshan DS

*(Reg.No: 39120019) from "Sathyabama Institute of Science and Technology Information"
for being an excellent and outstanding developer, interactive and responsible to
finishing the Flutter training and mini project "Notes & Todo Application"*

PRAVEEN KUMAR P, CEO, ALGOXFUSION
PVT. LTD.,

Name / Title of Presenter

09th November 2021

Issued Date

TABLE OF CONTENTS		
Chapter no.	Title	Page no.
	ABSTRACT	i
	LIST OF FIGURES	ii
1.	INTRODUCTION	1
	1.1 Getting Started	3
	1.2 Flutter as a framework	3
	1.3 Faster development	3
	1.4 Expressive and flexible UI	4
	1.5 Widgets	6
2.	SYSTEM REQUIREMENTS	7

	2.1 flutter for windows	7
	2.2 android studio	8
	2.3 java kit	9
	2.4 setup	11
	2.5 flutter advantages	14
	2.6 dart	17
	2.7 android app development	19
	2.8 google play store option	22
	2.9 open source platform	23

	2.10 con of android app	24
3.	AIM AND SCOPE OF TASK REMINDER	27
	3.1 Aim of task reminder	27
	3.2 Scope of task reminder	27
	3.3 Literature Review	27
	3.4 Proposed System	28
	3.5 Flow Chart	29
4.	ARCHITECTURE	30

	4.1 Components	30
	4.2 Main Dart	31
	4.3 Home screen	32
	4.4 Widgets	33
	4.5 Screenshots	35
5.	RESULT	41
	5.1 Result	41
	5.2 Summary	41
	5.3 Conclusion	41
	5.4 Appendix	42
	REFERENCE	58

LIST OF FIGURES		
FIGURE NO.	FIGURE NAME	PAGE NO.
1	Flutter zip file	7
2	Android studio installation package	9
3	Java installation	10
4	Adding path variable	11
5	Editing path variable	12
6	Checking flutter doctor	13

7	App flowchart	29
8	Welcome page	35
9	Signup page	36
10	Home page	37
11	Profile page	38
12	Notes page	39
13	Personal page	40

ABSTRACT

Reminder apps help you organize your to-do-list into something more manageable. There are even tools out there that allow you to list tasks according to their priority. Task reminders are an easy yet revolutionary app as they can even coordinate and function with our everyday technologies such as PCs, Mobile phones, and even smart watch which makes it as efficient and adaptive towards the growing technology as possible.

Cross-platform mobile application development is the pressing priority in today's world and generation. Developers are enforced to either construct the same application numerous times for various OS (operating systems) or accept a low-quality similar solution that trades native speed and accuracy for portability. Flutter is an open-source SDK for developing high-performance and more reliable mobile applications for operating systems like iOS and Android. Significant features of the Flutter are Just-in-time compilation which executes the computer code that encompasses compiling during program execution at run time rather than preceding execution. More frequently, this comprises of bytecode translation lesser known as source code to machine code, which is unswervingly executed. AOT compilation (Ahead-of-time compilation) compiles a high-level programming language such as C or C++, or an intermediary representation such as Java bytecode or NET Framework Common Intermediate Language (CIL) code into native system-dependent machine code so that the subsequent binary file can execute natively. Flutter has a feature called as hot reload which helps you easily experiment, build UIs, add features, and fix bugs. Hot reload works by inserting 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 reconstructs the widget tree, permitting you to rapidly view the special effects of your changes. 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 1

1 INTRODUCTION

Mobile applications are having a progressively more significant role in our day to day lives. Ever since November 2016, there is more network traffic made by mobile devices (48.19%) compared to desktops or laptops (47%). To dispense it to most of the users, a mobile application needs to familiarize itself with two independent platforms which are Android and iOS. These two platforms share immense dissimilarities which often necessitate different skill sets for developing. For example Java or Kotlin for Android and Object-C or Swift for iOS. Hence, developers and companies usually struggle to deal with the complex nature involved in developing cross-platform applications. On March 15, Facebook introduced an open-source cross-platform JavaScript framework called as React Native which strives to solve the aforesaid problem. In addition to its effective developing process, developers use React framework extensively because of its non-complex and easy nature. Further, Google announced another mobile SDK named Flutter in the latter half of 2016. Inspired by React Native, Flutter application can also run equally on both platforms, consequently decreasing the cost and complexity of application creation across iOS and Android. Flutter is fully built from scratch and at the time of scripting this study (Aug 2017), only Google uses it for commercial project. Cross-platform frameworks that show resemblance to React Native and Flutter, are discussed and implemented by various companies numerous times formerly. Still, neither of them suffices to satisfy the requirement of industrial development. In spite of the ineffective precursors, React Native and Flutter which are backed up by Facebook and Google, draw attention and people are optimistic about its prospect

1.1 Getting Started:

To become acquainted with the application development and with using, FLUTTER we recommend beginning with the:

Flutter is a free and open-source mobile UI framework created by Google and released in May 2017. Flutter allows you to create native mobile applications with only one code. It means that you can use one programming language and one codebase to create two different apps (IOS and Android).

1.2 Flutter as a Framework:

Using Flutter SDK one can build high-performance, high-fidelity apps for iOS, Android, Web, and Desktop from a single codebase. This enables developers to deliver high-performance apps that feel natural on different platforms. Programmers with little experience with object-oriented languages can build apps using Flutter since the apps are written in Dart, which looks familiar if you've used a language like Java or JavaScript.

1.3 Faster Development:

Flutter's hot reload feature helps us 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. Hot reload loads code changes into the VM and restores the widget tree, preserving the app state.

1.4 Expressive and Flexible UI:

Google Flutter focuses on native end-user experiences and offers plenty of designs that are expressive and flexible along with rich widget sets and vast animation libraries. The extensive layered architecture of the framework allows customization and incredible rendering speed.

1.4.1 HARDWARE REQUIREMENTS

RAM Processor:

1. Pentium processor of 400MHz or higher
2. Memory: Minimum 4GB primary memory
3. Hard Disk: Minimum 8 GB hard disk space
4. Monitor: preferably colour monitor(16-bit colour) or above
5. Compact disk drive
6. A mouse and a keyboard

1.4.2 SOFTWARE REQUIREMENTS:

1. Operating system: Windows operating system / Mac Os
2. Language: Dart
3. Database: Firestone
4. Application used: Android Studio
- 5 Flutter

OVERVIEW OF SOFTWARE REQUIREMENTS

INSTALLATION IN MACOS

To install Flutter on MacOS, you will have to follow the following steps:

Step 1: Go to URL, <https://flutter.dev/docs/get-started/install/macos> and download latest Flutter SDK. As of April 2019, the version is 1.2.1 and the file is flutter_macos_v1.2.1stable.zip.

Step 2: Unzip the zip archive in a folder, say /path/to/flutter

Step 3: Update the system path to include flutter bin directory (in ~/.bashrc file

Step 4: Enable the updated path in the current session using below command and then verify it as well. Flutter provides a tool, flutter doctor to check that all the requirement of flutter development is met. It is similar to the Windows counterpart.

Step 5: Install latest XCode, if reported by flutter doctor

Step 6: Install latest Android SDK, if reported by flutter doctor

Step 7: Install latest Android Studio, if reported by flutter doctor

Step 8: Start an android emulator or connect a real android device to the system to develop android application.

Step 9: Open iOS simulator or connect a real iPhone device to the system to develop iOS application.

Step 10: Install Flutter and Dart plugin for Android Studio. It provides the startup template to create a new Flutter application, option to run and debug Flutter application in the Android studio itself, etc., • Open Android Studio. • Click Preferences > Plugins. •

Select the Flutter plugin and click Install. • Click Yes when prompted to install the Dart plugin. • Restart Android studio.

1.5 WIDGETS

The core concept of the Flutter framework is In Flutter, Everything is a widget. Widgets are basically user interface components used to create the user interface of the application. In Flutter, the application is itself a widget. The application is the top-level widget and its UI is build using one or more children (widgets), which again build using its children widgets. This composability feature helps us to create a user interface of any complexity.

CHAPTER 2

2 SYSTEM REQUIREMENTS

Please begin by making sure that you are using supported hardware and devices for this release.

2.1 FLUTTER FOR WINDOWS:

Get the Flutter SDK

Download the following installation bundle to get the latest stable release of the Flutter SDK:

Fig 1: Flutter zip file



flutter_windows_2.5.3-stable.zip

For other release channels, and older builds, see the [SDK releases page](#).

Extract the zip file and place the contained flutter in the desired installation location for the Flutter SDK.

2.2 ANDROID STUDIO:

Install Android Studio

Setting up Android Studio takes just a few clicks.

First, be sure you download the latest version of Android Studio.

Windows:

To install Android Studio on Windows, proceed as follows:

If you downloaded an .exe file (recommended), double-click to launch it.

If you downloaded a .zip file, unpack the ZIP, copy the android-studio folder into your Program Files folder, and then open the android-studio > bin folder and launch studio64.exe (for 64-bit machines) or studio.exe (for 32-bit machines). Follow the setup wizard in Android Studio and install any SDK packages that it recommends. Android Studio offers a complete, integrated IDE experience for Flutter.

- Android Studio, version 3.0 or later Alternatively, you can also use IntelliJ:
- IntelliJ IDEA Community, version 2017.1 or later
- IntelliJ IDEA Ultimate, version 2017.1 or later

Fig 2: Android studio Installation package

Platform	Android Studio package	Size	SHA-256 checksum
Windows (64-bit)	android-studio-2020.3.1.25-windows.exe Recommended	914 MiB	decbe76f8b25a40c5ce8ea3a212dd23d3b3172e26d63e297291f8a555abed4e0
	android-studio-2020.3.1.25-windows.zip No .exe installer	922 MiB	37ba6592e8f11efe96a4a48ae82847452ca7d42d4eba3e5690f63eeb896734d1
Mac (64-bit)	android-studio-2020.3.1.25-mac.dmg	950 MiB	caa2a4a6adb5ff94e0fbb9ffec798d5b24319070d7d231684ea9a458b1420ee
Mac (64-bit, ARM)	android-studio-2020.3.1.25-mac_arm.zip	948 MiB	05e190b6f33f4f924d69772fe149a9d697125a3be1f0b72d805b947fa4e102d4
Linux (64-bit)	android-studio-2020.3.1.25-linux.tar.gz	935 MiB	cfc96e4981ad17e27ba91681fada06553fb118960c95f49232f72f30c2e2f781

2.3 JAVA KIT:

You can download Java 8 from the Oracle's Java official website.

Scroll down and you will see various download options for various platforms. You will see two option, 32 and 64 bit versions for download. You can choose either of them based on your system.

Since the world is moving towards 64bit architecture, I use the 64bit version. File ending with i586.exe is the 32bit version whereas the file ending with x64 is the 64bit version. Click on the version you want to download which is 64 bit version which in my case is jdk-8u241-windows-x64.exe

Go ahead, accept the licence agreement and download the file.

You will see the installation begins.

Fig 3: Java Installation

Java SE Development Kit 8u311

Java SE subscribers will receive JDK 8 updates until at least December of 2030.

The Oracle JDK 8 license changed in April 2019

The Oracle Technology Network License Agreement for Oracle Java SE is substantially different from prior Oracle JDK 8 licenses. This license permits certain uses, such as personal use and development use, at no cost -- but other uses authorized under prior Oracle JDK licenses may no longer be available. Please review the terms carefully before downloading and using this product. FAQs are available [here](#).

Commercial license and support are available for a low cost with Java SE Subscription.

JDK 8 software is licensed under the [Oracle Technology Network License Agreement for Oracle Java SE](#).

JDK 8u311 checksum

Linux macOS Solaris **Windows**

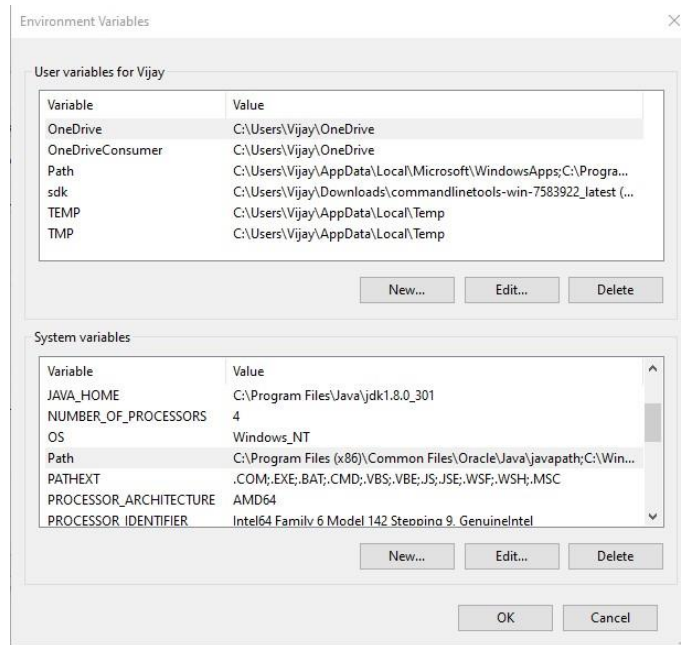
Product/file description	File size	Download
x86 Installer	157.37 MB	jdk-8u311-windows-i586.exe
x64 Installer	170.57 MB	jdk-8u311-windows-x64.exe

2.4 SETUP:

If you wish to run commands in the regular Windows console, take these steps to add Flutter to the PATH environment variable:

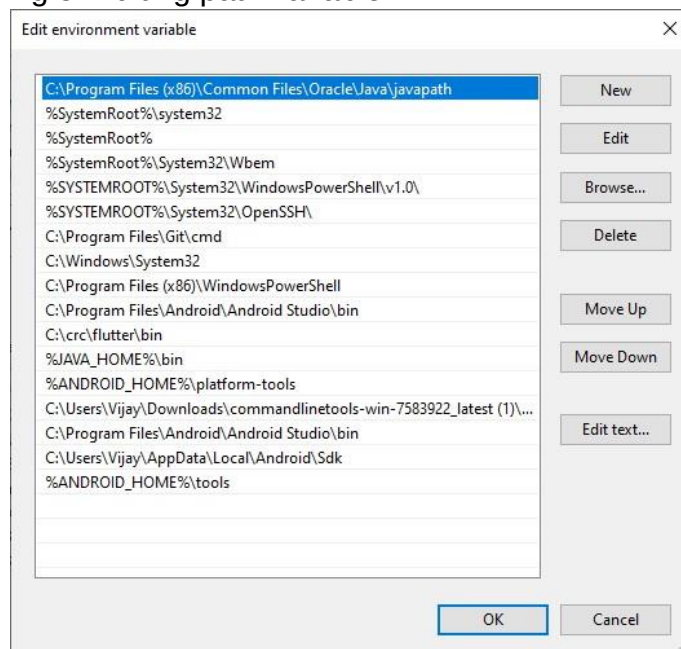
1. From the Start search bar, enter 'env' and select Edit environment variables for your account.
2. Under User variables check if there is an entry called Path:

Fig 4: Adding Path variable



Then edit the path.

Fig 5: Editing path variable



Run flutter doctor:

From a console window that has the Flutter directory in the path, run the following command to see if there are any platform dependencies you need to complete the setup:

C:\src\flutter>flutter doctor

This command checks your environment and displays a report of the status of your Flutter installation.

Fig 6: checking flutter doctor

```
Command Prompt
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Vijay>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 2.5.3, on Microsoft Windows [Version 10.0.19043.1288], locale en-IN)
[✓] Android toolchain - develop for Android devices (Android SDK version 31.0.0)
[✓] Chrome - develop for the web
[✓] Android Studio (version 2020.3)
[✓] Connected device (2 available)

• No issues found!

C:\Users\Vijay>
```

Now your system is ready to build the app.

2.5 FLUTTER ADVANTAGES:

Flutter has some unique features compared both to native development solutions and other cross-platform technologies. That's what makes it so appealing to product owners, tech leads, and developers. Here are the major **benefits of Flutter**:

2.5.1 Efficient development process

As a cross-platform technology, Flutter enables developing an app for both operating systems using the same codebase. But that's not the only reason why using Flutter makes the development process faster and more efficient. It also includes hot restart and a well-regarded "hot reload" function, which allows you to see updates in real time without restarting the app.

This significantly speeds up the development process. React Native and Xamarin have similar functions, but they are slower. With these features, Flutter lets you save time and resources.

2.5.2 Widgets and compatibility

Widgets are the building blocks on which the whole app is based. There are ready-made as well as customizable ones - everything in Flutter can be built from widgets. Since widgets are part of the app and not the specific platform, the end product will most likely have fewer compatibility issues on different OS versions.

2.5.3 High performance

Flutter applications perform at a level comparable with native apps and are winning over other cross-platform technologies. That's mostly because Flutter is the only mobile SDK that doesn't use a bridge (JavaScript or webview) for communication between the app and the platform. As a result, you get a faster-starting app with beautiful, fast animations and fewer performance problems.

2.5.3 Going beyond mobile

It's possible to develop Flutter products that work on the web, desktop, and even TV, which many developers are already doing. These features are still in different stages of advancement, but Google definitely has the capacity to create a stable release. Considering how important IoT is for the company, they will most likely keep developing Flutter in this direction.

2.5.4 Internationalization and accessibility

Internationalization, or preparing different language and region versions of your app, usually happens after the app itself is created and can produce multiple discrepancies. Flutter provides tools that make this process a breeze and incorporate it directly into

development. Flutter also supports accessibility, enabling large fonts, screen readers, and enhanced contrast, all of this automated from within the platform.

2.5.5 Open source and an engaged community

As an open-source platform, Flutter is free to use and has a growing community contributing to its outstanding documentation and assisting with issues that developers may encounter. There are also many YouTube videos available for those who want to start learning Flutter or improve their skills in this Google's mobile UI framework.

2.5.6 Faster Time-to-Market

Thanks to all the features that make developing easier in Flutter, you can release your app quicker and ship new features and upgrades to iOS and Android simultaneously.

Flutter is still an immature framework

It hasn't been around for long, which is why it's still not entirely stable. A number of more or less problematic issues remain, along with a lack of more advanced features that leverage the capabilities of operating systems. Many of these features are not yet supported, with plenty of libraries being in the pre-alpha stage and showing limitations when comparing them to native counterparts (i.e. Google Maps).

2.5.7 Dart is also pretty immature

When comparing it to Swift and Kotlin, it's basically like taking a step back - it has either fewer features or the existing ones are not exactly well-refined.

Flutter apps are quite large and “heavy” to start with

They occupy a lot of space and take longer to download or update.

The look & feel is not 100% the same as with native solutions

Basically, Flutter doesn't create native components. It somewhat replicates Android's Material Design and iOS-specific components with its Cupertino library, but it's not exactly the same. It's visible especially with different system versions where text fields or buttons vary from one another, yet stay the same in Flutter.

There are no single “guidelines” when it comes to developing Flutter apps which can be problematic when building more complex software.

The framework, as well as Dart language, are changing rapidly at times, which can make maintaining the code difficult in the long run.

Plus, given the track record of infamous [“projects killed by Google”](#), the future of Flutter might still be uncertain.

2.6 DART:

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”. Nonetheless, to attract developers that are not acquainted with JavaScript, Dart has a Java-like syntax. Flutter application renews the view tree on every new frame even when few other systems use reactive views. This behavior leads to a drawback that many objects, which might survive for a singular frame, will be created. As Dart is a modern programming language, it is optimized to handle this scenario in memory level with the help of “Generational Garbage Collection”

2.6.1 ADVANTAGES OF DART:

1. The first advantage is that it is easy to learn. Any JavaScript programmer can quickly relearn how to write code in Dart. To do this, they only need to familiarize themselves with the basic principles of this programming language.
2. The second thing is about its availability of documentation. Since Google is developing the interpreter for Dart, all the features of the language are described in detail. This allows you to quickly get answers to almost any questions that may arise during the training process, or directly while writing code.
3. The third advantage is its high performance. Programs written in Dart tend to run faster than programs created in JavaScript.
4. Dart is very stable and it can be used to build production quality real-time applications. It is an object-oriented programming language with support for inheritance, interfaces and optional typing features.
5. It uses AOT and JIT compilation – Dart has the unique capability to handle both Ahead of time and Just in time compiling. In AOT, the Dart code can be directly converted into native machine code. While in the mode of JIT, it can be compiled for exceptionally fast development cycles and game-changing workflow.
6. If you want to start writing your first Dart program without any installation or configuration, there is DartPad for you. The DartPad interface is plain and simple. Write your code and click 'Run' to execute the code. DartPad introduces users to the world of Dart programming. The support for libraries in DartPad is restricted to the basic level.

2.6.2 DISADVANTAGES OF DART:

1. The first drawback is that Dart has very limited resources online and it's hard to find solutions to problems. The reason for it is lack of a larger and cohesive developer communities, which will support you.
2. Dart is fairly new to the programmers and rarely used in the market. So if you will look for a job on this position, it will be harder to find a company with a vacancy of Dart programmer, because it hasn't yet spread widely.
3. Dart is currently in development. That's exciting, and it's not something "wrong" with Dart, but if you start developing in Dart now, there's a certain chance that the API will change, or that things won't be documented fully or correctly, and the amount of information on the web is less than, for example, the amount you can find about jQuery.
4. The absence of native support. For a beautiful demonstration of Dart's capabilities, developers must bring it to users' browsers. It turns out that to achieve at least 80% success, Dart developers must provide native support for it in all popular browsers.

To sum up, Dart is an interesting programming language with features to facilitate Web, mobile and command line apps. Its major advantages are its stability and ease of learning. In the case of Web applications, the combination with AngularDart makes it a very powerful tool. Generally, we think that despite its novelty, this programming language is worth your attention.

2.7 ANDROID APP DEVELOPMENT:

2.7.1 HOW IT WAS

Android Inc was founded in 2003 by a team of developers consisting of Andy Rubin, Chris White, Nick Sears, and Rich Miner. In 2005, Android Inc was bought by Google. Currently, Android monthly has about 200 million active users, as well as 2.6 million Android applications uploaded onto Google Play Store.

Google is interested in reaching as wide a user base as possible. Given the above, today for a company of any size there is a need to create Android applications for further development and strengthening its market position.

2.7.2 WHAT IS ANDROID APP DEVELOPMENT?

Android is an open-source mobile operating system. It has a simplified process of development, thanks to which it has a huge user base around the world. Where and when Android-based products are used? To achieve personal and corporate goals as applications are able to meet customer goals and increase business value.

2.7.3 PROS OF ANDROID APP DEVELOPMENT:

Graphics support

Any need for graphic design can be implemented through the Android app engineering, including graphics for gaming applications. In turn, the support of the demanded 2D and 3D-graphics contributes to the additional growth of traffic in the application.

Cost-effectiveness

Making a decision to enter the market of mobile applications, it is necessary first of all to choose a cost-effective platform. Android SDK - developer tools - is free of charge, except for a one-time registration fee. But the process of creation and testing itself takes place on computers and smartphones - sounds easy, doesn't it? The huge Android user base only proves the high return on development with low investment.

Android apps can compete with iOS in several categories, both in terms of the number of downloads and the number and volume of in-app purchases. Still, the difference in the paying capacity of Android and iOS users is an important point to keep in mind. At the same time, Android advertising budgets are lower, which means that ads can be shown to more users for the same amount of money as an iOS app. This is possible because of the large number of Android users.

A high level of security

The Android platform is highly secured. It has built-in features to protect the application from malware and viruses, which meets the security requirements of many organizations.

Integration options

Android provides the most flexible application integration into any system, be it a multimedia platform or a data management tool. The ability to seamlessly integrate also improves application performance.

Android forks

Android is an open-source system, which means that an unlimited number of alternative versions of this operating system are created. Each version - fork - is created for a different device, which means that, in fact, due to the forks, we do not get a pure Android.

Among the most common Android operating systems (forks) are the following: Oxygen OS, Fire OS, Jovi OS, MIUI, Funtouch OS, LineageOS, Flyme OS.

They differ from each other by a large number of specific features. For example, Fire OS was created by Amazon to run Amazon's Fire TV and tablets. The focus of the Fire OS is on content consumption, a customizable user interface, and connecting with Amazon services.

Common advantages of Android operating systems also include their customization, rich notification panel and fast updates. Thanks to them, Android has many out-of-the-box features and a huge community.

Broad market presence

Google, along with Samsung, Xiaomi and others, is constantly working to improve Android, making it easier for professional Android apps to be distributed geographically and demographically. They are user-friendly, easily adaptable to wearable gadgets, media streaming devices, Android-based smart TV, and compatible with technologies such as VR, AR, and IoT. Thanks to its ubiquity, the Android market share remains at the level of 70-80%.

The strong point of Android is that it is available on devices from many manufacturers, which, in addition, are more affordable, and therefore more widely distributed, compared to iOS devices. Given all this, you can see the deep penetration of your Android app

into the market.

By creating an Android application, you make it available on a large number of electronic devices of different manufacturers and spread all over the world (as opposed to iPhone). Android products can reach customers through multiple channels. In addition to traditional channels, such as the Google Play Store, AppGratis or Amazon Appstore, you can also create separate distribution and sales channels. On the one hand, getting through your own channels implies more custom work on setting up a marketing strategy. On the other hand, there may be less competition along the way.

Fast development process

The development cycle only takes a few hours. Isn't this an opportunity for a business to transform its idea into a product within an agreed and acceptable time frame? The software development speed is one of the convincing advantages of Android app programming. But there are also some nuances that need to be considered. They will be revealed later in the text.

Versatility

Android application development is in the Java language, which makes it possible to target it on various platforms, such as Ubuntu and Symbian. This is one of the reasons why use Android app engineering over others, as it ensures a wise allocation of resources.

2.8 Google Play Store options

It takes a few hours for Android apps and games to be approved in the Google Play store (in the App Store, it takes several days). Developers can make changes within a few

hours, which makes Google Play especially useful when it comes to new applications. As a rule, after uploading, they have to be adjusted several times according to user requests. But in Google Play, applications can even be tested through

personal electronic devices before marketing the final version.

Another advantage of Android apps is that in Google Play developers can release products as MVP and provide access to their testing to certain groups of users. But not only the developers, but also the devices themselves can control the testing process, determining the percentage of users whose versions of the application need to be updated. This approach will serve to correct possible errors, as a result, users will get a ready-to-use application.

If your app is on Google Play Store, it also gets into the Google app indexing library. The number of application views leads to an increase in the number of visits to the site. Thus, one action - making an app available on the Google Play Store - solves the other marketing tasks: increasing website ranking in Google, website traffic growth, client number growth.

2.9 Open-source platform

Android is an open-source OS, this means the following:

- Developed on Java native Android apps, can be ported to the Ubuntu, Chrome OS, Blackberry, Symbian;
- Android Software Development Kit is available for free download;
- The source code can be shared with community members, thereby contributing to the improvement of the quality of Android as a platform;
- There is a room for innovation, creativity, additional features, which makes Android more flexible than iOS;
- Android app development saves money thanks to a fast and easy development process;

- Free and simple integration with other Google products, namely Google Calendar, Google Maps, Google documents, which will improve the user experience of your application;
- The pull of subject-matter experts is vast - each Java developer is able to build Android-based apps.

2.10 Cons of Android Mobile App Development:

Android app development is progressing rapidly, but this does not mean, however, that the process is pitfalls-free. Below we will take a look at the main hiccups faced by Android apps developers, and how to solve them.

2.10.1 No one-size-fits-all UI

One of the biggest obstacles to overcome is an absence of a consistent user interface that could provide an equally easy user experience across all devices. Google has yet to come up with a set of rules to guide mobile app developers. In turn, developers also do not adhere to any specific rules. There is the point why one of the cons of Android app development is the inability of mobile applications to provide all devices with an equally convenient UX.

How to deal with this? To ensure UI consistency on all devices, forward-thinking developers use a responsive template and test the application incrementally on different platforms.

2.10.2 API incompatibility

Often, to make an application more functional, many developers resort to integrating many third-party APIs. However, not all APIs are suitable for every operating system, which causes compatibility issues.

2.10.3 Security issues

On the one hand, that's a good thing that Android is an open-source OS, as developers can customize a platform depending on tasks and specifications. On the other hand, the openness of the code makes it vulnerable to all sorts of security threats. This is the developers' responsibility to provide the development process and the app itself with a sufficient level of security through encryption.

2.10.4 Low visibility

The Google Play store has a huge number of applications for users to choose from. At the same time, free ones are preferred over paid ones. This means, you need to come up with a marketing strategy on how to monetize the application to get ahead of the competition now and then.

2.10.5 Copyright issues

Recently developed Android apps are not thoroughly tested by Google. This allows developers to release applications as quickly as possible. At the same time, there is a high risk that a large number of applications will be similar to each other in terms of functionality, design, and concept. Patents and copyrights are often violated. By working with experienced developers, you can effectively address this issue and have a grasp of how to create a truly unique application without violating patent rules.

2.10.6 Low user purchasing power

Developing an app or game for a broad or growing market is the right choice. That being said, it's important to keep in mind that most users of Android devices are located in countries with low purchasing power and, quite possibly, there is no credit card. It is unlikely that such an audience would be able to buy a paid app. The optimal solution for them is a free app with ads inside.

2.10.7 The development process may be slowed down

Android app development can be slower compared to iOS due to Android fragmentation and system release cycles. In the case of Android, the application needs to be compatible with all platforms and devices. All of them come with different performance levels, resolutions, and screen sizes. If testing goes wrong, there is a risk of leaving a piece of the market without your application. You should constantly test the application until you are sure that it works correctly across all devices and versions.

CHAPTER 3

AIM AND SCOPE OF TASK REMINDER

3.1 AIM:

Task Reminder apps are tools for your smartphones that remind you of what you need to do. Reminder apps help you organize your to-do-list into something more manageable. There are even tools out there that allow you to list tasks according to their priority. Task reminders are an easy yet revolutionary app as they can even coordinate and function with our everyday technologies such as PCs, Mobile phones, and even smart watch which makes it as efficient and adaptive towards the growing technology as possible

3.2 SCOPE:

The future utility of task reminder is nearly endless as modern technologies are adding more and more complex qualities such as tracking location which adds a variety of new prospects to this app as we can further add a specific location with the time and date. Task reminder applications not only stop as a common civilian application but is also used in various public sectors and private sectors such as military, hospitals as well as civil aviation department , though the actual application used in such departments might be more complex, they perform using the same theory and application.

3.3 LITERATURE REVIEW

Application Development Using Android, Muhammad Zubair Asghar, International

Journal of Advanced Computer Science and Applications, Vol. 7, No. 5, 2016. The objective behind making this application was to make it easy for teachers and other staffs to conduct quiz for students during the work from home time. So, while surveying as to on which platform or rather operating system the project must be implemented, we selected android for the following reasons: • Android is an open-source platform • Supports multifunction • Provides rich tools to make an interactive application

Downloading the software required for making the application is absolutely free. Along with this, we surveyed the popularity of the operating system. Market share of android which was a mere 2.8% in 2009(initial stage), boosted to 48% till August 2011 which is almost half the share of the total market. Our basic aim is to make the application reach as many people as possible and this goal is achieved by implementing the application on android. Share of worldwide 2011 Q2 Smartphone sales to end users by O.S, according to Gartner. 4 The Internet provides a capability so powerful and general that it can be used for almost any purpose that depends on information, and it is accessible by every individual who connects to one of its networks. It supports human communication via e-mail, “chat rooms,” newsgroups, and audio and video transmission and allows people to work collaboratively at many [7] different locations. It supports access to digital information by many applications, including the World Wide Web. The Internet has proved to be a spawning ground for a large and growing number of “e-businesses” (including subsidiaries of traditional “brick-and-mortar” companies) that carry out most of their sales and services over the Internet. Many experts believe that the Internet will dramatically transform business as well as society.

3.4 Proposed System

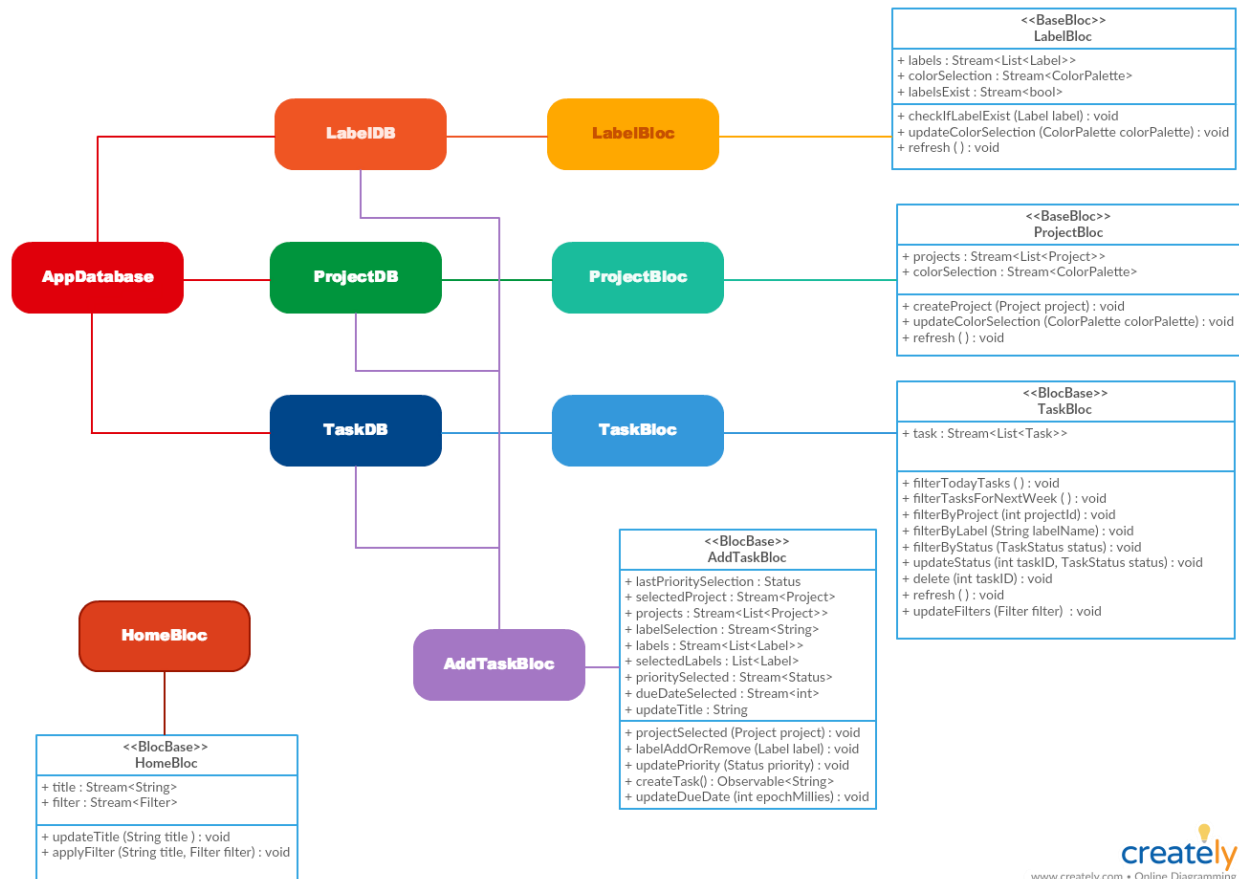
The proposed system is an application for the Android /iOS platform mobiles that will help in improving knowledge and accuracy. Android is a [8] Linux-based operating system designed primarily for touch screen mobile devices such as smartphones and tablet computers, developed by Google in conjunction with the Open Handset Alliance.

The proposed system will be developed for Android and iOS mobiles only because of the market share of Android and iOS is more than other operating systems.

3.5 FLOW CHART:

The flowchart is a means of visually presenting the flow of data through an information processing system, the operations performed within the system and the sequence in which they are performed. In this lesson, we shall concern ourselves with the program flowchart, which describes what operations (and in what sequence) are required to solve a given problem. The program flowchart can be likened to the blueprint of a building. As we know a designer draws a blueprint before starting construction on a building. Similarly, a programmer prefers to draw a flowchart prior to writing a computer program. Here in this flowchart first the user needs to register or login If ID is invalid then he can't get the home page.

Fig 7: app flowchart



CHAPTER 4

ARCHITECTURE

4.1 COMPONENTS:

- android - Auto generated source code to create android application
- ios - Auto generated source code to create ios application

- lib - Main folder containing Dart code written using flutter framework
- lib/main.dart - Entry point of the Flutter application
- test - Folder containing Dart code to test the flutter application
- test/widget_test.dart - Sample code
- gitignore - Git version control file
- metadata - auto generated by the flutter tools
- packages - auto generated to track the flutter packages • iml - project file used by Android studio
- pubspec.yaml - Used by Pub, Flutter package manager
- pubspec.lock - Auto generated by the Flutter package manager, Pub
- README.md - Project description file written in Markdown format

4.2 MAIN DART:

```
import 'package:taskly/app/theme/app_theme.dart';
```

```
"import 'package:flutter/material.dart';  
import 'package:flutter/services.dart';  
import 'package:get/get.dart';  
import 'package:get_storage/get_storage.dart';
```

```

import 'app/routes/app_pages.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  SystemChrome.setPreferredOrientations(
    [DeviceOrientation.portraitUp,
    DeviceOrientation.portraitDown]); await GetStorage.init();
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      theme: appThemeData[AppTheme.YellowLight],
      debugShowCheckedModeBanner: false,
      title: "Taskly",  initialRoute:
      AppPages.INITIAL,  getPages:
      AppPages.routes,
    );
  }
}

```

4.3 HOME SCREEN:

Fig 8: Home screen



4.4 WIDGETS:

- Widgets are high level objects used to describe any part of an application. It can be but is not limited to UI elements such as buttons ; layout (alignment, padding) ;data (a theme, configurations)
- Widgets can be everything. In flutter, almost everything you'll code will be inside a Widget. Even things like "redux reducers" are widgets in flutter.

- The core concept of the Flutter framework is In Flutter, Everything is a widget.
Widgets
- are basically user interface components used to create the user interface of the application.
- In Flutter, the application is itself a widget. The application is the top- level widget and its UI is build using one or more children (widgets), which again build using its children
- widgets. This composability feature helps us to create a user interface of any complexity.

This small bit of code shows an important concept in Flutter — everything is a widget. Our whole app is a widget, which contains the `MaterialApp` widget. `Scaffold` is a widget which helps us quickly create a proper Material layout without needing to worry about manual styling. `AppBar` is a widget which accepts a title, and creates a bar at the top of the screen, which is commonly seen in apps. On Android it will align the text to the left, and on iOS it will align it to the centre.

Since we have made big changes to the app, this time a hot reload will not work. Instead we need to do a full restart of the app. On the command line, type `R` -- note it is uppercase, unlike for a hot reload. You will see a simple app with a title bar.

4.4.1 STATELESS AND STATEFULL WIDGETS:

To make our app look more like a todo app, let's show some tasks. You may have noticed that our simple app above is a `StatelessWidget`. This means that it cannot be changed dynamically. For our todo app, this is not good, because todo items get added and removed all the time. However, a `StatelessWidget` can have children which are dynamic, which are `StatefulWidget`s. Let's break out our stateful functionality (the todo list container) from the overall app container.

To make a stateful widget, we need two classes — one for the widget itself, and a second which creates the state. This setup allows the state to be easily saved, and enables the use of features such as hot reloading.

4.5 SCREENSHOTS:

4.5.1 WELCOME PAGE:

When we first launch the app. We will see this Opening page it's the welcoming page.

Fig 8: welcome page



Taskly

Professionals' life is a Mess

Get Organized



Start

4.5.2 SIGNUP PAGE:

When we first launch the app. We will see this Opening page which shows the sign up options. Here we need to fill name, email and password .which we can use for logging in next time.

Fig 9: signup page



First Name



Second Name



Email



Password



Confirm Password

SignUp



5G

Email

Password

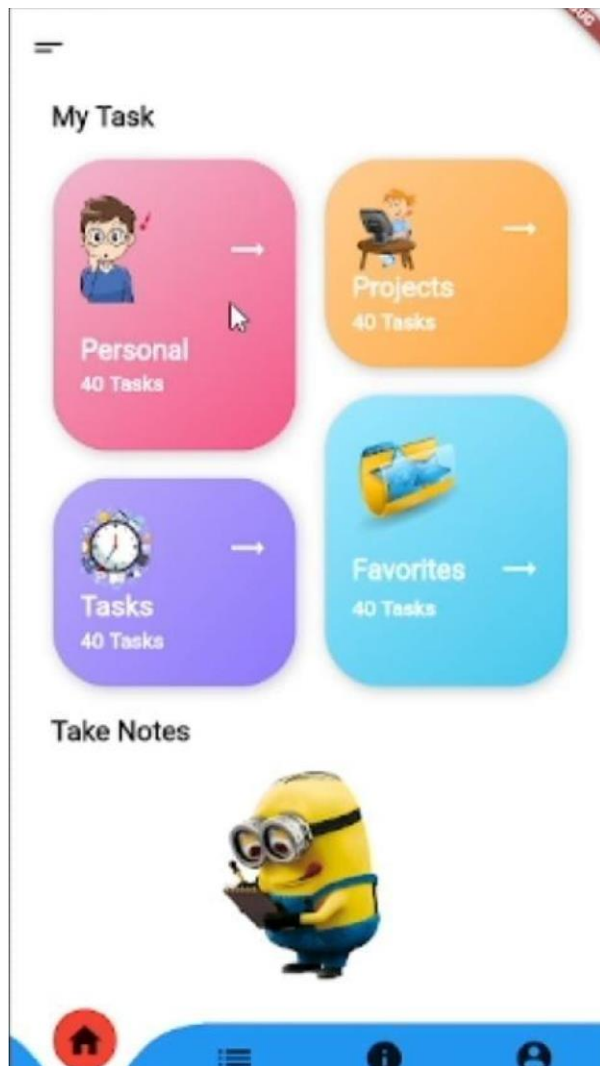
Login

Don't have an account? [SignUp](#)

4.5.3 HOME PAGE:

This is the main page where you can add different tasks and timings and save it.

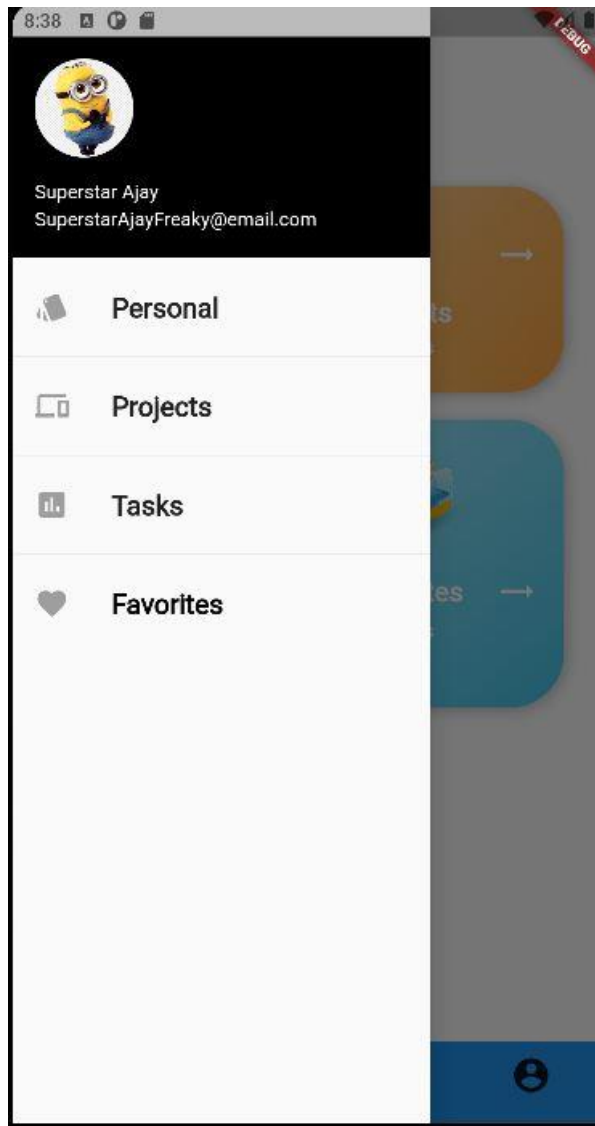
Fig 10:Home page



4.5.4 PROFILE PAGE:

Profile page contains of account settings.

Fig 11: Profile page



4.5.6 PERSONAL PAGE:

This page is where the user can create their personal lists .

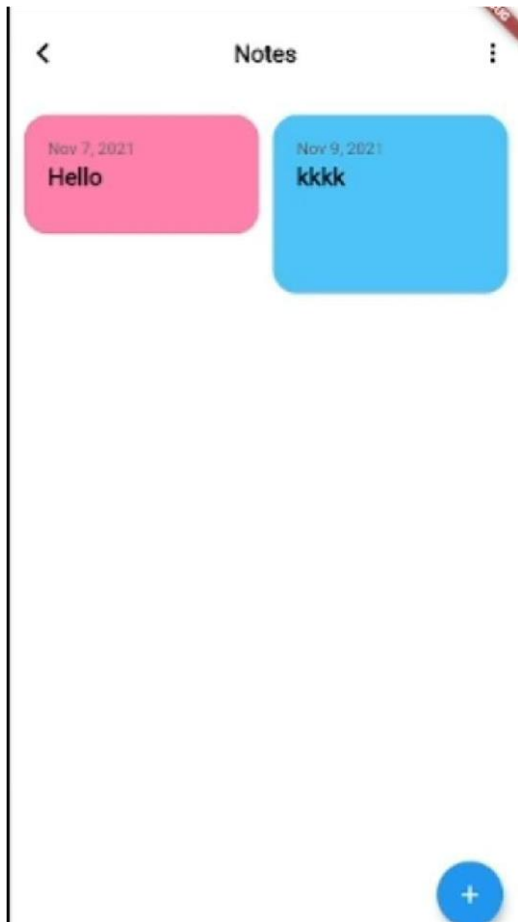
Fig 12:personal page



4.5.7 NOTES PAGE:

This page is where the user of the app can Organize Their notes with CURD operations Which means Create,Update,read and delete.

Fig 12: Notes page



CHAPTER 5

RESULT AND SUMMARY

5.1 RESULT

This Chapter describes discuss the objectives of the system stipulated in earlier chapters, limitation of the system conclusion and recommendation of the System

5.2 SUMMARY

The project is done with the help of Android Studio and Smart Phone. The code is written in Android Studio using Dart programming language. The application is running in the smart phone by connecting to the system at the first time before installation.

5.3 CONCLUSION

As we are leaving in a Fast growing and competitive world we need our life to be more systematic and disciplined in order to be successful. This application helps our life to be the same by making our remind our duties or tasks on time. The app helps in increasing your productivity and keeping track of your to be done tasks.

You can easily add repeated alarm for your routine activities, or add an alarm to any specific task so you won't miss any of them. In future , the application can be developed in schools and colleges and make sure every staffs and students do their duties or tasks on time.

5.4 APPENDIX

5.4.1 SOURCE CODE:

```
import 'package:flutter/material.dart';  
import 'package:flutter/services.dart';  
import 'package:get/get.dart';  
import 'package:get_storage/get_storage.dart';  
import 'package:taskly/app/theme/app_theme.dart';
```

```
import 'app/routes/app_pages.dart';
```

```
void main() async {
```

```
  WidgetsFlutterBinding.ensureInitialized();  
  SystemChrome.setPreferredOrientations([  
    DeviceOrientation.portraitUp,  
    DeviceOrientation.portraitDown  
  ]);  
  await GetStorage.init();  
  runApp(MyApp());  
}
```

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return GetMaterialApp(  
      theme: appThemeData[AppTheme.YellowLight],  
      debugShowCheckedModeBanner: false,  
      title: "Taskly",  
      initialRoute: AppPages.INITIAL,  
      getPages: AppPages.routes,  
    );  
  }  
}
```

```
import 'package:get/get.dart';  
import 'package:taskly/app/modules/home/bindings/home_binding.dart';  
import 'package:taskly/app/modules/home/views/home_view.dart';  
import 'package:taskly/app/modules/profile/bindings/profile_binding.dart';  
import 'package:taskly/app/modules/profile/views/profile_view.dart';  
import 'package:taskly/app/modules/settings/bindings/settings_binding.dart';
```

```
import 'package:taskly/app/modules/settings/views/settings_view.dart';
import 'package:taskly/app/modules/splash_screen/bindings/splash_scre
n_binding.dart';
import 'package:taskly/app/modules/splash_screen/views/splash_screen_v
iew.dart';
import 'package:taskly/app/modules/welcome/bindings/welcome_binding.da
rt';
import 'package:taskly/app/modules/welcome/views/welcome_view.dart';

part 'app_routes.dart';

class AppPages {
  static const INITIAL = Routes.SPLASH_SCREEN;
```

```
static final routes = [
  GetPage(
    name: _Paths.HOME,
    page: () => HomeView(),
    binding: HomeBinding(),
  ),
  GetPage(
    name: _Paths.SPLASH_SCREEN,
    page: () => SplashScreenView(),
    binding: SplashScreenBinding(),
  ),
  GetPage(
    name: _Paths.PROFILE,
    page: () => ProfileView(),
    binding: ProfileBinding(),
  ),
  GetPage(
    name: _Paths.SETTINGS,
    page: () => SettingsView(),
    binding: SettingsBinding(),
  ),
]
```

```

    GetPage(
      name: _Paths.WELCOME,
      page: () => WelcomeView(),
      binding: WelcomeBinding(),
    ),
  ],
];
}

part of 'app_pages.dart';
// DO NOT EDIT. This is code generated via package:get_cli/get_cli.dar
t

abstract class Routes {
  static const HOME = _Paths.HOME;
  static const SPLASH_SCREEN = _Paths.SPLASH_SCREEN;
  static const PROFILE = _Paths.PROFILE;
  static const SETTINGS = _Paths.SETTINGS;
  static const WELCOME = _Paths.WELCOME;
}

abstract class _Paths {
  static const HOME = '/home';
  static const SPLASH_SCREEN = '/splash-screen';
  static const PROFILE = '/profile';

```

```

  static const SETTINGS = '/settings';
  static const WELCOME = '/welcome';
}

import 'package:flutter/material.dart';
import 'package:flutter_svg/flutter_svg.dart';
import 'package:get/get.dart';
import 'package:taskly/app/global_widgets/input_text_feild.dart';
import 'package:taskly/app/global_widgets/proceed_button.dart';
import 'package:taskly/app/modules/home/controllers/home_controller.da

```

```
rt';
import 'package:taskly/app/theme/text_theme.dart';

class BottomSheetContent extends GetView<HomeController> {
  final Function onSubmit;
  final String buttonText;

  BottomSheetContent({this.onSubmit, this.buttonText});

  @override
  Widget build(BuildContext context) {
    return Container(
      padding: EdgeInsets.symmetric(horizontal: 20, vertical: 20),
      height: Get.height * 0.6,
      decoration: BoxDecoration(
        borderRadius: BorderRadius.only(
          topLeft: Radius.circular(20),
          topRight: Radius.circular(20),
        ),
        color: Theme.of(context).scaffoldBackgroundColor,
      ),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.center,
        mainAxisAlignment: MainAxisAlignment.center,
        mainAxisSize: MainAxisSize.min,
        children: [
          Container(
            height: 60,
            width: 60,
            decoration: BoxDecoration(
              color: Theme.of(context).primaryColorLight,
              borderRadius: BorderRadius.circular(10),
            ),
            child: GetBuilder<HomeController>(
              id: 'dropDownIcon',
```

```

init: HomeController(),
builder: (controller) {
    return DropdownButton(
        iconSize: 0,
        elevation: 6,
        value: controller.selectedIcon,
        underline: Container(color: Colors.transparent),
        hint: Padding(
            padding: EdgeInsets.only(left: 10, top: 10),
            child: SvgPicture.asset(
                controller.icons.first,
                width: 40,
                height: 40,
            ),
        ),
        items: controller.icons.map(
            (String value) {
                return DropdownMenuItem<String>(
                    value: value,
                    child: Padding(
                        padding: EdgeInsets.only(left: 10, top: 10),
                        child: Container(
                            width: 40,
                            child:
                                SvgPicture.asset(value, width: 40, hei
ght: 40),
                        ),
                    ),
                );
            },
        ).toList(),
        onChanged: controller.changeIcon,
        dropdownColor: Theme.of(context).primaryColorLight,
    );
},
),
),
),

```

```

        SizedBox(height: Get.height * 0.03),
        InputTextFormField(
          hintText: 'Title',
          tController: controller.titleController,
          textFeildColor: Theme.of(context).primaryColorLight,
          contentTextColor: Theme.of(context).primaryColorDark,
        ),
        SizedBox(height: 25),

```

```

        InputTextFormField(
          hintText: 'Description',
          tController: controller.descController,
          textFeildColor: Theme.of(context).primaryColorLight,
          contentTextColor: Theme.of(context).primaryColorDark,
        ),
        SizedBox(height: Get.height * 0.03),
        Row(
          children: [
            InkWell(
              onTap: () {
                controller.selectDate(context);
              },
              child: InputTextFormField(
                hintText: 'Date',
                tController: controller.dateController,
                textFeildColor: Theme.of(context).primaryColorLight,
                contentTextColor: Theme.of(context).primaryColorDark
              ),
            ),
            SizedBox(width: Get.width * 0.05),
            Text(
              '@',
              style: kSub2HeadTextStyle.copyWith(
                color: Theme.of(context).primaryColorDark, fontSiz

```

```

e: 24),
    ),
    SizedBox(width: Get.width * 0.05),
    InkWell(
      onTap: () {
        controller.selectTime(context);
      },
      child: InputTextFormField(
        hintText: 'Time',
        tController: controller.timeController,
        textFeildColor: Theme.of(context).primaryColorLight,
        contentTextColor: Theme.of(context).primaryColorDark
      ),
    ),
    width: Get.width * 0.36,
    isEnabled: false,
  ),
),

```

```

    ],
  ),
  SizedBox(height: Get.height * 0.015),
  Container(
    width: Get.width * 0.435,
    child: Row(
      children: [
        Text(
          'Repeat Daily',
          style: kSub2HeadTextStyle.copyWith(
            color: Theme.of(context).primaryColorDark,
            fontSize: 16,
          ),
        ),
      ],
    ),
    GetBuilder<HomeController>(
      id: 'isRepeat',
      init: HomeController(),
      initState: (_) {},
      builder: (_) {

```


REFERENCE

- <https://projectworlds.in/flutter-medicine-reminder-app-project-using-android-studio/>
- <https://flutterawesome.com/android-water-reminder-app-made-with-flutter-using-android-apis/>
- [https://ijesc.org/upload/b3930ac14331fd1b425af8cd1c341d41.Cross%20Platform%20Development%20using%20Flutter%20\(1\).pdf](https://ijesc.org/upload/b3930ac14331fd1b425af8cd1c341d41.Cross%20Platform%20Development%20using%20Flutter%20(1).pdf)
- <https://medium.com/the-web-tub/making-a-todo-app-with-flutter-5c63dab88190>