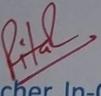


Thadomal Shahani Engineering College
Bandra (W.), Mumbai- 400 050.

© CERTIFICATE ©

Certify that Mr./Miss RAGHAV RATHI
of Computer Department, Semester IV with
Roll No. 2113208 has completed a course of the necessary
experiments in the subject PCE II under my
supervision in the **Thadomal Shahani Engineering College**
Laboratory in the year 2023 - 2024


Teacher In-Charge

Head of the Department

Date 20 October 2023

Principal

CONTENTS

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1.	COVER LETTER AND RESUME	1-6	28 Aug, 23	
2.	SHORT PROPOSAL	7-14	11 Sept, 23	
3.	MEETING DOCUMENTATION	15-21		
3.1	Notice and Agenda	15-17	7 Aug, 23	
3.2	Minutes	18-21	7 Aug, 23	
4.	TECHNICAL PAPER	22-27	7 Oct, 23	
5.	STATEMENT OF PURPOSE	28-31	4 Sep, 23	
6.	INTERPERSONAL SKILLS	32-38		
6.1	Time Management	32-32	21 July, 23	
6.2	Leadership	33-34		
6.3	Conflict Resolution	35-35		
6.4	Start up Skills	36-38		
7.	APTITUDE TEST	39-44	27 Oct, 23	

CONTENTS

EMPLOYMENT SKILLS
[Cover letter]
[Assignment no 1]

K

Draft a cover letter & resume for the
Post of software development engineer 1
in Amazon.

27 August, 2023

Snapnil Jain
The Hiring Manager,
Amazon Office,
Bandra Kurla Complex,
Bandra,
400051

Subject: Job application for Software
Development Engineer 1

Dear Sir

I am writing this to express my
interest towards the software Development
Engineer 1 (SDE 1) position at Amazon
with a strong concepts inside & outside
of my curriculum a passion toward
programming & a versatile approach
towards development and problem

solving, I am eager to step into the corporate world with this position, & contribute towards Amazon's dynamic and innovative environment.

As an engineering student in Thadomal Shahani Engineering College, I have been exposed to a vast & rigorous curriculum that has equipped me with deep understanding of various concepts that are required for a Software Development Engineer which include Data Structures and Algorithms, RDBMS, OOP & other software engineering principles. My dedication to honing my programming skills has been demonstrated through my active GitHub page and my active Hackerrank profile. These experiences not only have enhanced my technical abilities but also helped me understand concepts & improve my problem solving skills & grow as a person.

I am particularly drawn to Amazon for its customer-centric focus, innovative technologies, & commitment to excellence. Your company's culture of continuous learning & cutting-edge development aligns perfectly with my personal

values and aspirations. I am excited about the opportunity to collaborate some of the brightest minds in the industry & contribute to projects that impact millions of users worldwide.

What sets me apart is my eagerness to learn my adaptability to new technologies & my keen interest in problem solving. I am confident with technologies like C++, java, python, DMP, SQL and my proficiency with the data base design & problem solving will make me valuable & an impactful contributor to the SDE team.

Thank you for considering my application. I am excited about the possibility of joining Amazon as a SDE & contributing towards the ongoing success. Please find the attached document as my resume. I would be glad to discuss my goals in more detail via interview. Please feel free to reach out rvr@gmail.com

Sincerely,

Raghav Rathi

Encl : Resume

RESUME

Name : Raghav V. Rathi
Email ID : raghavrathi8855@gmail.com
Mobile No : 8855996970
My all Socials : <https://withrvr.bio.link/>

OBJECTIVE: TO ACHIEVE SELF ACTUALIZATION & PROFESSIONALISM

Aiming to find a position in the website development software engineering field where I could utilize and enhance my technical skills and gain a good experience of working in a challenging environment.

EDUCATION QUALIFICATION :

EDUCATION	YEAR OF PASSING	COLLEGE/SCHOOL	UNIVERSITY/BOARD	Percentage
B.Tech	2025	Thadomal Shahani Engineering College	Mumbai	-
Diploma in Computer Science	2021	Government Polytechnic Amravati	Amravati	97.05%
S.S.C	2018	Bhawarilal Samra English High School	Amravati	88.60 %

EXPERIENCE

- 2 month of internship

TECHNICAL PROFICIENCY

- OS : Windows, Ubuntu
- Have the Knowledge of computer languages
HTML, CSS, JS, React, NextJS, Node, Tailwindcss, Bootstrap,
Python, Django, Java, C, C++, PHP, ... etc
- ... More detailed

SOME OF MY PROJECTS

Name	Why ?	Github Repo	Live Demo
1 Link	Create 1 & Only 1 Link, for my YT Description	link	link
JavaScript Equality Table	To use === always, instead of ==	link	link
Chess Training Game	To improve chess positions, muscle memory	link	link
Twitter API	On Sir demand, to store tweet in files	link	link
Hulu Clone	How to create clone websites	link	link
Github-API	See users Github stats	link	link
Increase Reaction Time	How fast are gamers / aim practice	link	link

QUALITIES

- Ability to Work in Group as well as independently
- Sincere and hard working
- Passion for constant improvement
- Like to learn new things
- Fast Grasping Power

INTERESTS

- Competitive Coding, Math, Typing
- Computer Gaming, Sports (Chess, Football, Badminton, Swimming ...)
- Traveling

PERSONAL INFORMATION

Date of Birth : 18/11/2002
Gender : Male
Nationality : Indian
Marital Status : Single
Permanent Address : Near Ram mandir opposite to Gopal Talkies, Rajapeth, Amravati Maharashtra

DECLARATION

I do hereby declare that the above information is true to the best of my knowledge.

Resume .

Raghav. Rathi raghavrathi8875@gmail.com +918855996970
(<https://www.r.bio.link/>)

Education : → Government polytechnic Amravati (2018 - 21)
→ Thadomal Shahani Engineering College (2021 - 25)

knowledge / skills : → Web - Development (frontend, Backend, full stack), HTML, CSS, Javascript ; PHP
React.js, Next.js, Node.js, Django.
→ App - Development (Flutter, React native)
→ Other languages (C, CPP, Java, Python, Dart)

Project : → 1Link, Twitter tweet Downloader, Connect4
→ Chess Training, Reaction Time, CDF docs,
→ Email Replier using Chat GPT, ..., etc.
(... many more 10+ small project, github.com/wittus).

Experienced : → Web developer Inter (Mindzita) (Aug 21 to Oct 21)
→ Organizer [Connecting Dots forever (Jan 23 to present)]
→ Creator [Tech/code with RVR (2020 - present)]

Speciality : → Eager to learn new things.
→ Adjust in any environment quickly.
→ Work at any location, any mode.
→ Good Communication skills (English, Hindi, ..etc)
→ Any Marketing skills / group person / interactive

Hobbies : → Chess, Video game, Typing, outdoor games
→ Travelling, video/photo creation (YT, insta)
→ Swimming, Cricket, football

PROPOSAL WRITING

[Short Proposal]

[Assignment No. 2]

K

Write a short proposal for a customer support portal for a hard drive manufacturing company

1. Introduction

In today's fast-paced business environment, the bridge between companies and their customers is the nexus of success. With consumer expectations constantly on the rise, product-based companies must harness innovative tools to connect, assist, and engage with their customers effectively. Our proposed Flutter-based product represents a visionary solution meticulously designed to revolutionize customer support and interaction. This cutting-edge product seeks to provide a versatile communication platform, enabling customers to seek assistance, engage in real-time conversations through chat, voice calls, or video calls, and access a comprehensive FAQ system for immediate product-related inquiries.

By embracing this solution, our client will not only bolster customer satisfaction but also establish a stronghold in the competitive market by offering a state-of-the-art communication channel. The Customer Support Portal outlines the benefits, and potential impact of this groundbreaking product, demonstrating its alignment with the evolving landscape of customer service and its capacity to foster stronger, more meaningful relationships between our client and their valued customers.

The product will enhance customer satisfaction and engagement resulting in establishment of the company's name in terms of customer services. The scalable nature of our project will help the system in maintaining performance standards and reliability of the system ultimately providing you more market growth and customer satisfaction. A personalized support section will not only help your company grow financially but it will reduce the significant hours that are wasted when the customer faces a trivial issue that can be solved over call but yet he/she calls up a technician on sight.

2. Purpose

The purpose of this comprehensive Flutter project is to redefine the way the company connects and engages with the customers of their products. The central objective of the project is to craft a dynamic and versatile communication ecosystem that helps customers to seamlessly engage and initiate timely support conversations with the support sector employees of the company, and in

turn get comprehensive insights about their problems with the products that they have purchased with unparalleled ease. Through real-time chat, voice calls, and video calls, we aspire to forge a closer and more personal connection between the company and its diverse customer base, facilitating swift and tailored support experiences. Simultaneously, the meticulously curated FAQ section will serve as an invaluable self-help repository, granting customers immediate access to answers for their product-related queries.

We are dedicated to elevating the overall customer experience for the company's products with the help of this comprehensive platform. We aim to enhance customer satisfaction by providing flexible communication channels that cater to a wide range of preferences and needs. Through personalized interactions, we intend to foster stronger, more enduring customer relationships that transcend the transactional.

Furthermore, our ambition is to solidify the company's reputation as an industry leader in customer-centricity, recognized for its commitment to empowering customers with knowledge, offering responsive support, and facilitating easy and seamless engagement. This project is about forging lasting connections and elevating customer-centricity to new heights. And with Flutter, we have the robust framework needed to achieve these ambitions efficiently and effectively.

3. Features

The comprehensive system that you will be delivered with will have features ranging from generic features to specific user privileges and rich designing, here are some of the general features that the provided system will have:

Minimalist yet Rich UI/UX: A rich UI/UX will help the users and technicians to flow through the project with ease and use their needed modules seamlessly

Multi-user concurrency: The product that you will be delivered will allow maximum 500 users to use it concurrently without causing any problems, which in turn helps in connecting with more customers at the same time and consequently increasing customer satisfaction.

The other features that the provided system will have are user specific, the features for the customers and technicians will have contrast features, meaning that they will have the same features provided but both on the opposite side of the system. The following are the key features we will be providing them:

- **Authentication:** The customers as well as the technicians will be provided with a login or register (if they are using the system for the first time) option.
- **Profile:** The customers and the technicians will have their profile which they can update any time that they want to make any tweaks.
- **Chat with each other:** One of the core features of our system, a chat portal for the technician and the customer, a minimalist chat system which allows them to communicate with each other via text, images and videos.

- Voice call each other: Another one of the core features that allows both sides to get on a voice call for any kind of issue or query the customer has.
- Video call each other: The third core feature that allows both sides to get on a video call to visually showcase the issue and communicate regarding the problem the customers are facing.
- Get control of the customer's system: The technician will have the privilege to get control of the user's desktop/system in scenarios where it is needed.
- Troubleshooting articles: The Customer will be able able to read the trouble shooting articles curated by the technicians and the administrator of the system which might help the type of users who have some moderate to high knowledge of their product and how it works and don't want/need to communicate with a technician at the moment. Or maybe if sometimes the technicians take time to get on a call with the customer, the customer can try using the articles up till then.
- FAQ: The Frequently Asked Questions section curated by the technicians will be available to the user to further look if their problem is a very common one that most users face or their query is very naive so they can check up the FAQ section to get answers on their own without talking to anyone.

The third user of our system will be the administrator who will have the following privileges:

- Get communication history: The admin will be able to get the communication history between the customers and the technicians.
- Update technicians list: Add, delete or update any technician in the list whenever needed.

4. Plan

Our first step will be holding an initial kickoff meeting with key partners, who include reps from the product-based enterprise and members of our cross-functional crew that includes developers, designers, QA engineers, and a PM. Defining the project's scope, objectives, and expectations, the primary goal is clear. Ensuring a seamless project outcome, this collaboration involves a well-organized strategy based on clear project scope definition followed by requirement collection. Gathering insights into the specific needs of the client support and product teams, we undertook interviews and polls. Based on insights gathered through this current process stage, features prioritization includes real-time chat support, voice & video calls, and an easy-to-navigate help section.

Having gained insight into project demands, the design process kicks off. It is the responsibility of our design team to generate wireframes and prototypes representing the UX/UI of the app. Close collaboration with the design team guarantees an alignment between the app's layover and the organization's brand identity while also incorporating visual appeal and user friendliness.

With the Flutter framework, our focus was on cross-platform app development. By choosing the right technologies, we ensure smooth support interactions and well-organized FAQ management during this phase.

Initially coding concentrates on building the basics of the app, starting with real-time chat support and FAQs. User authentication, secure data storage, and real-time communication demand a sturdy backend architecture, and we deliver. Integral to identifying and resolving bugs and issues are regular code reviews and testing.

Of great importance to us is quality assurance within the development process. Performance is optimized by thoroughly testing the application across multiple platforms and devices. High-quality standards demand that problems spotted are dealt with quickly.

With real-time support features now integrated, including chat support and video/voice calls. Ease of use and accessibility can also be achieved through our user-friendly interfaces, we design them for customers and support agents alike.

FAQ management efficiency is essential. Adding, editing, and grouping material are all made simpler with our system.

Created specifically to help both end users and support personnel get the most out of the app, materials exist. To guarantee a flawless user encounter, regular backing and learning opportunities are given.

Prioritized through the entire project lifecycle, compliance with relevant data protection laws, data security, and user privacy are paramount.

Prior to official launch, UAT collaborations ensure app effectiveness validation and user feedback collection.

After development, apps are made available on app stores for users to access. To ensure top performance, continuous monitoring takes place, with prompt action on areas needing enhancement.

User concerns and problems are addressed by a committed team after launch. Enhancing features, improving performance, and keeping apps updated; regular releases ensure.

Policies and compliance take center stage in efforts to ensure full accordance with legal codes, industry models, and ethical principles.

Key to us is promoting the educational testing app within educational institutions. Among educators and students, strategies are developed to encourage adoption.

Evaluation of application effectiveness and impact is something that happens continuously. To support continuous improvement, teacher and student feedback helps identify key performance indicators.

5. Project Implementation

The implementation of Customer Support Portal relies on a combination of technology frameworks that facilitate mobile app development. Flutter serves as the foundation for our application, ensuring a consistent user experience across Android and iOS platforms. Firebase will function as the backend, offering real-time data storage through Firestore and user authentication via Firebase Authentication. These services facilitate data handling and authentication for customer support interactions.

Socket-Based Communication:

To enable real-time chat, voice calls, and video calls, we will harness socket-based communication technology, such as WebSocket or Firebase Realtime Database. These technologies enable low-latency interactions between customers and the employees in the support sector of the company, ensuring easy, fast and reliable communication. This real-time capability transforms customer support into a dynamic and engaging experience, ultimately leading to improved customer satisfaction, customer trust and engagement.

Testing, Security, and Scaling:

Comprehensive testing will cover Flutter and Firebase integrations, with an emphasis on compatibility and performance. Security measures, including Firebase security rules and encryption, safeguard customer data and communications. The plan includes provisions for scalability within Firebase to accommodate a growing user base and future enhancements. By integrating Flutter for cross-platform development, Firebase for backend capabilities, and socket-based technology for real-time interactions, Customer Support Portal will deliver an efficient customer support platform.

6. Schedule

Milestone	Estimated Days
Project Initiation	5
Requirement Gathering	3
Design and Planning	20

Technology selection	2
Development	40
Quality assurance	15
Integration	5
Support resource management	3
User training	3
Documentation	5
Security and Compliance	8
User Acceptance Testing(UAT)	10
Deployment and initial monitoring	5
Ongoing Maintenance and Support	20
Promotion and Adoption	5
Evaluation and Feedback	5

7. List of resources

- Development Environment:

Computer with sufficient processing power and memory for Flutter development. Text editor or Integrated Development Environment (IDE) like Visual Studio Code for coding and testing.

- Flutter Development Tools:

Flutter SDK for cross-platform app development. Dart programming language for building the app's logic. Emulators or physical devices for testing on both Android and iOS platforms.

- Backend Infrastructure:

Server or cloud-based backend to handle real-time chat, voice calls, and video calls. Use Firebase or a similar platform for real-time communication and data storage.

- Communication Technologies:

Implement socket-based communication technology like WebSocket for real-time interactions.

- Utilize WebRTC for secure and efficient voice and video calls.

- UI/UX Design:

Graphic design software or design tools for creating an intuitive and visually appealing user interface.

- Database Management:

Database management system (DBMS) like MongoDB to store customer data and chat history securely.

- Content Management:

Develop a robust content management system for managing FAQs and product-related information.

- Testing and Quality Assurance:

QA testing tools and devices for comprehensive testing of chat, voice, and video call functionalities. Debugging tools and error tracking for identifying and resolving issues.

- Version Control:

Version control systems like Git to track code changes and facilitate collaboration within the development team.

- Analytics and Monitoring:

Data analytics tools to gather insights on user interactions and support ticket trends. Performance monitoring tools for optimizing app functionality.

- Cloud Hosting (Optional):

Cloud hosting platforms like AWS, Azure, or Heroku for scalability to ensure reaching maximum customers and ease of deployment and maintenance.

8. Budget

Item Description	Total Cost (In Rs)
Developer license on company's name	5000
Firebase services	12,000/year
Content writing	8000
Security and compliance	17,000
Maintenance	13,000/year
Cloud hosting(optional)	15,000/year
Total	70,000

9. Conclusion

On the whole, our Flutter project represents a substantial step forward in terms of customer assistance and interaction. Integration of various communication methods allows us to take customer service to previously unreached heights. Issue resolution occurs swiftly, leading to higher levels of customer satisfaction and increased retention.

While an upfront investment must be made, there is a promising return on investment. Operational efficiency gains will come about through decreased support response time. Customer feedback collected through these interactions allows us to perfect our products and services more effectively tailoring them to match their expectations. Customer-centricity and innovation alignment set us apart as market leaders through this project.

Thank you for your time, feel free to contact us at group@gmail.com

MEETING DOCUMENTATION

[Notice and Agenda]

[Assignment No. 31]



Draft to notice and agenda from the
first meeting of the Board of
Directors.

TECHNOLOGY SERVISE LIMITED

16th rd, Thane (W)
Thane - 04

Ph: 987654321 E-mail: main@tach.com

July 21, 2023

NOTICE

This is to inform you that first meeting of the Board of Directors will be held at the Registered office of the company at 9:00 am on July 28, 2023. The agenda is attached.

Sd/-

Secretary

TECHNOLOGY SERVICE LIMITED

16 ac, Thane (w)
Thane - 401

Ph: 9826543210 E-mail: main@tech.com

AGENDA

- 1.01 Casual Wear or formal Wear
- 1.02 Work from home or work from office.
- 1.03 Starting a new R&D Department
- 1.04 Starting an inter hiring program
- 1.05 Any other matter with the permission of the chairperson.
- 1.06 Vote of thanks

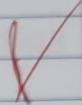
Sd/-

Secretary.

MEETING DOCUMENTATION

[Minutes]

[Assignment No. 32]



Draft the minutes of the first Meeting of the Student concile held at 3:00 pm on July 31, 2023 whose agenda is as follows.

- 1.01 Confirmation of the chairperson
- 1.02 Member decide the event.
- 1.03 Technical members decide the problem statement / domain
- 1.04 Spenser of the event
- 1.05 Mode of event
- 1.06 Topics of workshop.
- 1.07 Date and Location of event
- 1.08 Date of the next Meeting.
- 1.09 Vote of Thanks.

Sd/-

Secretary.

1B

THADOMAL SHAHANI ENGINEERING COLLEGE

July 31, 2023

Minutes of the first meeting of Student Council held in Room No 303 Old Building at 3:00 pm on July 31, 2023

Present : Sakshi Bhandari Chairperson
Samrath Shetty
Hirday Rochani
Pratham Yawb
Disha Bajaj
Tanya Hindija
Sawan Sharma
Nipoon Dombani
Bhavisha Hemwani
Ritika Gupta

In attendance : Akash Nair Secretary .

1.01 Confirmation of the Chairperson Resolved that Sakshi Bhandari to be the chairperson

1.02 Member decide the event It was decided to add event in two parts workshop and then Machathons.

- 1.03 Technical members decide the problem statement / domain It was decided by team to add many different domains like AI, ML, APP and Web development, blockchain, etc.
- 1.04 Sponsor of the event To search for new sponsor and send mail and messages to current connected network.
- 1.05 Mode of event first phase of shortlisting will be online and main showcase and ending event will be online.
- 1.06 Topics of workshop Basic guideline line hackathon explanation and events, also political topic to talk on was not decided.
- 1.07 Date and location of event Decided to conduct in labs at TSEC Bandra September 3, 2023

1.08 Date of the next meeting To conduct in August with in 2 weeks at same location . 31st August , 2023

1.09 Vote of Thanks It was concluded by vote of thanks from the chairperson

Akash Nair

Sakshi Bhavdari

Secretary
August 31, 2023
9:30 am

Chairperson
August 31, 2023
9:30 am

A comparative study on Flutter and React Native

Amiraag Dubey¹, Hiten Dusseja², Om Avhad³, Ritika Gupta⁴, Vrajesh Hegde⁵, Sawan Sharma⁶, Raghav Rathod⁷

Om Shahani Engineering College, Bandra West, Mumbai, Maharashtra 400050
Computer Engineering Department



Abstract - This paper presents a comparative analysis of popular cross-platform mobile app development frameworks, Flutter and React Native. We explore their fundamental features, development experiences, real-world case studies, and considerations such as performance, security, and maintenance. This study aims to provide valuable insights for developers and organizations seeking guidance in choosing the most suitable framework for their mobile app projects.

Keywords: Dart, Flutter, Javascript, React Native, Components, etc.

INTRODUCTION

Mobile development is highly important in today's digital landscape due to the prevalence of mobile devices and the need for seamless user experiences. Flutter and React Native play significant roles as cross-platform frameworks, streamlining development and enabling efficient creation of apps that work across multiple platforms. Flutter's rich UI capabilities and React Native's performance advantages are particularly noteworthy, allowing developers to produce visually appealing, high-performance apps that cater to modern user expectations. These frameworks contribute to businesses' success by accelerating app development, ensuring consistent experiences, and meeting the demands of an ever-evolving digital market.

As businesses and developers strive to create efficient and engaging applications, this comparison helps identify which framework better aligns with project requirements. Evaluating factors like performance, cross-platform capabilities, development speed, and community support ensures that the chosen framework maximizes development efficiency and delivers high-quality user experiences, contributing to the success of app projects in this competitive digital era.

Flutter
Flutter transcends the conventional boundaries of a mere framework, ushering in a paradigm shift in app development methodologies. At its core, Flutter stands as an open-source UI development toolkit, meticulously crafted by Google. Its primary purpose lies in empowering developers to create natively compiled applications that seamlessly traverse mobile, web, and desktop platforms. All this is achieved through a unified codebase. Flutter's distinctiveness shines in its ability to create visually captivating and immersive user interfaces, which consistently replicate the native experience across an array of platforms.

Flutter is based on the Dart programming language. Dart is a language developed by Google, and it serves as the primary

language for building apps using the Flutter framework. The best thing about Flutter is that you can use it to create applications for various platforms, including Android, iOS, Windows, macOS, and Linux. Also, Flutter has an extensive developer community that is working continuously to make it more powerful and feature-rich.

1.2 React Native:

React Native defies conventional categorization, emerging as a revolutionary force that propels app development. At its core, React Native embodies an open-source mobile application framework conceived by Facebook. Its essence lies in empowering developers to engineer applications with a native aura that seamlessly spans iOS, Android, and beyond. React Native empowers you to craft visually compelling and interactive user interfaces that mirror native experiences. React Native as it makes use of React or React JS, which is the fastest-growing JavaScript front-end library.

By leveraging React JS's component-based approach and its underlying principles, React Native enables developers to compose user interfaces using familiar methods. The integration of React JS's core principles into React Native optimizes code reusability, enabling developers to craft a single codebase that functions across multiple platforms, minimizing duplication and enhancing development efficiency.

2. REQUIREMENTS

Requirements in this context refer to the necessary components and tools needed for developing mobile applications with Flutter and React Native.

2.1 Hardware Requirements

The hardware requirements of flutter and react native include:

- CPU: A multi-core, 32 bit or 64 bit processor.
- RAM: 4 GB(required) and 8 GB(suggested)
- Disk Space: The disk space required is to store the SDK and other tools related to the frame-work in the system. A good amount of additional space for the projects is also recommended but the minimum disk space both the frame-works required are as follows:
 - Flutter: 2 GB
 - React Native: 5 GB

2.2 Operating System

The operating system helps with particular tools and libraries that are designed to work on particular operating systems.

Following are the OS requirements for both frameworks:

- Windows: Windows 10 or later (64-bit), x86-64 based.
- MacOS: 10.14 (Mojave) or later.
- Linux: Modern Linux Distribution(64 bit)

2.3 Software prerequisites

The software requirements and prerequisites are the software

components that the system needs in order to develop apps using these frameworks.

Flutter

- Flutter SDK: The SDK(Software development kit) includes Dart programming language compiler and other tools that are required to build an app using Flutter.
- Java Development Kit(JDK): The JDK is required because the flutter app runs on the android device needs the JDK
- Git: Git is required for version control which is a crucial part in development of apps
- Platform-Specific Dependencies: Depending on the platforms the project is targeting (iOS or Android), one might need extra platform specific tools.
- IDE(Integrated Development Environment): An IDE provides an environment where the developer can edit and debug code. Flutter is compatible with various IDEs like Android Studio, Visual Studio Code, XCode(MacOS).
- Emulator(Android or iOS): An emulator is an optional tool that might be required in some cases to run the project on an emulated mobile phone that is actually running on the desktop.

React Native:

- Node.js: Node.js is the primary base requirement for React Native since it relies on Node.js for its build and development.
- npm or yarn: Node Package Manager(NPM) or Yarn are the dependencies that are used to install and manage dependencies. NPM is provided with Node.js and yarn is not inclusive of node.js
- JDK: Just like Flutter, React Native also requires JDK for android development
- IDE: Just like Flutter, React Native also requires an IDE that helps developers to build and use the development environment. Compatible IDEs include Android Studio, XCode
- CocoaPods: CocoaPods is a dependency manager for swift and objective C projects which will be used to build iOS projects using react native
- Emulator: Just like Flutter, React Native might also require emulators in some cases

3. ARCHITECTURE

The architecture of a tech framework encompasses a set of components and layers designed to provide core functionality, abstractions, and interfaces for developers and users. This typically includes a central core, APIs and interfaces for interaction, middleware for handling auxiliary tasks, a data access layer for managing data interactions, and support for extensions, plugins, and configuration. Security, testing tools, documentation, and community support are vital considerations in crafting a robust framework architecture, with the specifics varying based on the framework's domain and design principles.

3.1 Flutter Architecture

This article is intended to provide a high-level overview of the architecture of Flutter, including the core principles and concepts that form its design.

Flutter is a cross-platform UI toolkit that is designed to allow code reuse across operating systems such as iOS and Android, while also allowing applications to interface directly with underlying platform services. The goal is to enable developers to deliver high-performance apps that feel natural on different platforms, embracing differences where they exist while sharing as much code as possible.

During development, Flutter apps run in a VM that offers stateful hot reload of changes without needing a full recompile. For release, Flutter apps are compiled directly to machine code, whether Intel x64 or ARM instructions, or to JavaScript if targeting the web. The framework is open source, with a permissive BSD license, and has a thriving ecosystem of third-party packages that supplement the core library functionality.

This overview is divided into a number of sections:

- The layer model: The pieces from which Flutter is constructed.
- Reactive user interfaces: A core concept for Flutter user interface development.
- An introduction to widgets: The fundamental building blocks of Flutter user interfaces.
- The rendering process: How Flutter turns UI code into pixels.
- An overview of the platform embedders: The code that lets mobile and desktop OSes execute Flutter apps.
- Integrating Flutter with other code: Information about different techniques available to Flutter apps.
- Support for the web: Concluding remarks about the characteristics of Flutter in a browser environment.

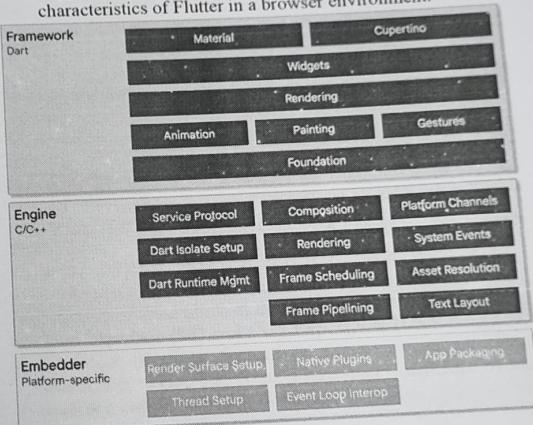


Fig 3.1 Flutter Architecture

3.2 React Native Architecture

The React Native New Architecture marks a significant departure from its predecessor, ushering in a series of noteworthy enhancements. Foremost among these improvements is its commitment to performance optimization, achieved through direct communication between JavaScript and native threads, as well as the implementation of lazy loading for native modules. Memory usage is streamlined through the sharing of a shadow tree between JavaScript and native threads. Additionally, developers benefit from a more type-safe and debug-friendly environment.

At its core, this architecture revolves around key components: the JavaScript Virtual Machine (JSVM) executing bundled JavaScript code, React handling UI components, and Metro bundling JavaScript into a single executable file for the JSVM. The Renderer is responsible for UI rendering, while Native Modules represent native code accessible to JavaScript. The JavaScript Interface (JSI) enables seamless communication between JavaScript and native modules. Native UI, the visible user interface, is constructed based on the shadow tree generated

ng execution.

This architecture is orchestrated as follows: the JSVM executes the JS bundle, generating a shadow tree sent to the renderer for native UI creation. User interactions trigger events sent to the JSVM, updating the shadow tree, which, in turn, prompts the renderer to adjust the native UI. Additional features include the Native Modules for efficiency, Fabric as an improved rendering engine, and Hermes as a more compact JavaScript engine. While still in development, the React Native New Architecture is suitable for production use and recommended for all new React Native applications.

React Native New Architecture

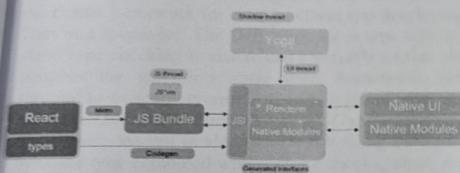


Fig 3.2 React Native Architecture

Performance

As we have already mentioned, Flutter is written in Dart language, which allows compiling a code quicker than JavaScript. Moreover, thanks to this, Flutter displays animations at 60 fps as a standard. As a rule, all Flutter apps have a high performance by default. Still there are some major differences between the two which are discussed below:

1. Compile and run

Compile and run refers to the process of translating source code into executable code and then executing it. The steps for compiling and running code depends on the programming language used.

- Flutter: Flutter uses the Dart programming language and compiles the entire application into native ARM code that runs directly on the device's CPU. This eliminates the JavaScript bridge and allows more direct interaction with the device's hardware, improving performance.

- React Native: React Native uses JavaScript and utilizes a bridge to communicate between JavaScript code and native modules. This bridge adds an additional layer of communication between your app and native components, which can introduce overhead in terms of performance.

4.2 UI rendering

UI rendering is the process of displaying a graphical user interface (UI) on a screen, including elements like text, images, and controls.

- Flutter: Flutter has its own rendering engine and does not rely on the platform's native UI components. This gives you consistent and smooth UI performance across different devices and platforms.
- React Native: React Native leverages the platform's native UI components, which can lead to performance differences across devices and platforms. React Native has greatly improved its UI performance over time, but complex UI animations and interactions can still experience performance variability.

4.3 Hot reload

UI rendering is the process of displaying a graphical user interface (UI) on a screen, including elements like text, images, and controls

- Flutter: Flutter's "hot reload" feature allows developers to immediately see changes to their code reflected in their app without requiring a full recompilation. This has a positive impact on developer productivity and contributes to faster development.
- React Native: React Native also provides a "fast update" feature that achieves similar results. However, the performance of this feature may vary depending on the complexity of your app and your specific use case.

4.4 Native module access

Native module access allows a programming language or framework to interface with platform-specific code, enabling developers to utilize platform features and functionality beyond the language's capabilities.

Flutter: Flutter provides platform-specific channels for direct access to native code. This is useful for tasks that require low-level interaction with the device's hardware.

React Native: React Native allows the use of native modules through a bridge. This allows access to device-specific features, but communication overhead can impact performance, especially for time-sensitive tasks.

4.5 Third party libraries

Third-party libraries are pre-written code modules or packages created by developers outside of a project to provide specific functionality, enhancing development efficiency

Flutter:

Flutter has a more consistent ecosystem, with third-party packages often following similar design philosophies, resulting in more consistent performance.

React Native:

React Native offers a wide selection of third-party libraries due to its long presence in the market. However, the quality and performance of these libraries may vary.

4.6 User Interface (UI) and User Experience (UX)

- UI (User Interface) focuses on design elements for user interaction, while UX (User Experience) considers overall usability, satisfaction, and flow.

Flutter:

- Flutter apps have a consistent UI across platforms because they use their own set of customizable widgets that mimic native components. This can lead to a visually uniform experience.

React Native:

- React Native apps use native components provided by the platform, which can give a more native look and feel. However, achieving pixel-perfect consistency across platforms might require additional effort.

4.7 Tooling and Debugging

"Tooling" refers to the software tools and utilities used in the development process, while "Debugging" is the process of identifying and fixing errors or issues in a program

- Flutter: Flutter provides a rich set of developer tools, including the Flutter DevTools, which offer insights into UI rendering, performance profiling, and more.
- React Native: React Native has solid debugging tools, including the ability to inspect components and use browser-based developer tools for debugging JavaScript code.

Controlling build costs, employing effects and widgets only when necessary, applying lazy methods for lists or grids, creating

showing frames in 16ms are some recommendations made by Flutter. We should be aware that Flutter does not perform better than React Native in terms of performance. However, many developers admit that Flutter has too many other advantages, which make it a better choice.

Language and Documentation

Dart is a client-optimized language for developing fast apps on every platform. Its goal is to offer the most productive programming language for multi-platform development, paired with a flexible execution runtime platform for app frameworks.

I Dart for flutter

Dart is a programming language developed by Google. It's often used for building web, mobile, and desktop applications, with a strong focus on performance and simplicity. It's notably used with the Flutter framework for cross-platform app development.

Dart as a language: The Dart language is type safe; it uses static type checking to ensure that a variable's value always matches the variable's static type. Sometimes, this is referred to as sound typing. Although types are mandatory, type annotations are optional because of type inference. The Dart typing system is also flexible, allowing the use of a dynamic type combined with runtime checks, which can be useful during experimentation or for code that needs to be especially dynamic.

Dart has built-in sound null safety. This means values can't be null unless you say they can be. With sound null safety, Dart can protect you from null exceptions at runtime through static code analysis. Unlike many other null-safe languages, when Dart determines that a variable is non-nullable, that variable can never be null. If you inspect your running code in the debugger, you see that non-nullability is retained at runtime; hence sound null safety.

Dart as a platform: Dart's compiler technology lets you run code in different ways: Native platform: For apps targeting mobile and desktop devices, Dart includes both a Dart VM with just-in-time (JIT) compilation and an ahead-of-time (AOT) compiler for producing machine code. Web platform: For apps targeting the web, Dart can compile for development or production purposes. Its web compiler translates Dart into JavaScript.

Dart Native (machine code JIT and AOT): Dart Native (machine code JIT and AOT) During development, a fast developer cycle is critical for iteration. The Dart VM offers a just-in-time compiler (JIT) with incremental recompilation (enabling hot reload), live metrics collections (powering DevTools), and rich debugging support. When apps are ready to be deployed to production—whether you're publishing to an app store or deploying to a production backend—the Dart ahead-of-time (AOT) compiler can compile to native ARM or x64 machine code. Your AOT-compiled app launches with consistent, short startup time.

- The AOT-compiled code runs inside an efficient Dart runtime that enforces the sound Dart type system and manages memory using fast object allocation and a generational garbage collector.

- The Dart runtime: Regardless of which platform you use or how you compile your code, executing the code requires a Dart runtime. This runtime is responsible for the following critical tasks:

- Managing memory: Dart uses a managed memory model, where unused memory is reclaimed by a garbage collector

(GC).

- Enforcing the Dart type system: Although most type checks in Dart are static (compile-time), some type checks are dynamic (runtime). For example, the Dart runtime enforces dynamic checks by type check and cast operators.
- Managing isolates: The Dart runtime controls the main isolate (where code normally runs) and any other isolates that the app creates.
- On native platforms, the Dart runtime is automatically included inside self-contained executables, and is part of the Dart VM provided by the dart run command.

5.2 Javascript

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It was a general-purpose scripting language to ensure web pages' interoperability across different browsers and devices. JavaScript has continued to grow alongside new browsers like Mozilla Firefox and Google Chrome since then. The latter even started developing the first modern JavaScript engine, called V8, which compiles bytecode into native machine code. Today, JavaScript has plenty of frameworks and libraries to simplify complex projects, such as AngularJS, jQuery, and ReactJS. Originally run on the client-side, the JavaScript implementation has branched out to the server-side after the Node.js development – a cross-platform server environment built on the Google Chrome JavaScript V8 engine.

- Just-in-time compilation: A JavaScript engine is a software component that executes JavaScript code. The first JavaScript engines were mere interpreters, but all relevant modern engines use just-in-time compilation for improved performance. JavaScript engines are typically developed by web browser vendors, and every major browser has one. In a browser, the JavaScript engine runs in concert with the rendering engine via the Document Object Model. The use of JavaScript engines is not limited to browsers. For example, the V8 engine is a core component of the Node.js and Deno runtime systems. Since ECMAScript is the standardized specification of JavaScript, ECMAScript engine is another name for these engines. With the advent of WebAssembly, some engines can also execute this code in the same sandbox as regular JavaScript code.
- Runtime environment: JavaScript typically relies on a run-time environment (e.g., a web browser) to provide objects and methods by which scripts can interact with the environment (e.g., a web page DOM). These environments are single-threaded. JavaScript also relies on the run-time environment to provide the ability to include/import scripts (e.g., HTML <script> elements). This is not a language feature per se, but it is common in most JavaScript implementations. JavaScript processes messages from a queue one at a time. JavaScript calls a function associated with each new message, creating a call stack frame with the function's arguments and local variables. The call stack shrinks and grows based on the function's needs. When the call stack is empty upon function completion, JavaScript proceeds to the next message in the queue. This is called the event loop, described as "run to completion" because each message is fully processed before the next message is considered. However, the language's concurrency model describes the event loop as non-blocking: program input/output is performed using events and callback

functions. This means, for instance, that JavaScript can process a mouse click while waiting for a database query to return information. When using React Native, you're going to be running your JavaScript code in up to three environments. In most cases, React Native will use Hermes, an open-source JavaScript engine optimized for React Native. If Hermes is disabled, React Native will use JavaScriptCore, the JavaScript engine that powers Safari. Note that on iOS, JavaScriptCore does not use JIT due to the absence of writable executable memory in iOS apps. When using Chrome debugging, all JavaScript code runs within Chrome itself, communicating with native code via WebSockets. Chrome uses V8 as its JavaScript engine.

3 Flutter Documentation:

Flutter documentation is said to be one of the most organized code documentations.

Official Documentation Quality: Flutter's official documentation is known for its clarity and organization. It provides a step-by-step guide for beginners, covering installation, creating a new project, and understanding the basic structure. The documentation is available at Flutter's official website, and it includes numerous code samples, explanations, and visuals to aid understanding.

Widget-Centric Approach: Flutter's documentation places a strong emphasis on its widget-centric approach to UI development. Widgets are the building blocks of Flutter applications, and the documentation provides an extensive catalog of widgets, explaining their properties and usage. The "Widget Catalog" section of the documentation is particularly useful for exploring the available widgets.

Hot Reload: Flutter's documentation highlights the Hot Reload feature as a powerful tool for improving the development workflow. With Hot Reload, developers can see changes made to the code in real-time, allowing for quicker iteration and debugging. This feature is discussed in detail in the "Flutter in Focus" video series within the documentation.

Integration of Dart: Flutter uses the Dart programming language. The documentation provides an introduction to Dart, explaining its syntax, concepts, and features. This is helpful for developers who are new to Dart but want to work with Flutter effectively.

Theming and Customization: Flutter's theming and customization capabilities are explained in the documentation's "Building a theme" guide. It covers how to define and apply themes to maintain a consistent look and feel across the app.

Rich Ecosystem: The documentation not only covers the core Flutter framework but also highlights the Flutter ecosystem. This includes information about packages available on pub.dev, which is the official package repository for Flutter. The documentation explains how to search for, use, and contribute to packages.

5.4 React Native Documentation

React Native also has a well-versed documentation.

- Official Documentation Quality:** React Native's official documentation is comprehensive and well-structured. It provides an in-depth guide to building mobile apps using React Native, available at React Native's official website. The documentation covers installation, project setup, and guides for various aspects of app development.
- Component-Based Approach:** React Native's documentation introduces the concept of components, which are reusable building blocks for creating UIs. It

explains how to use core components provided by React Native, such as View, Text, and Image, along with styling and layout techniques.

- Hot Reloading:** React Native's documentation also highlights the Hot Reloading feature. It's discussed in the "Fast Refresh" section of the documentation, explaining how developers can see their changes reflected in the app without a full restart.
- JavaScript Programming:** Since React Native uses JavaScript (and optionally TypeScript), the documentation provides guidance on writing components using React's declarative syntax. It also delves into managing state, handling events, and using hooks for more modern state management.
- Native Modules:** React Native allows interaction with native code through native modules. The documentation explains how to create native modules in languages like Java, Objective-C, and Swift, and how to bridge them with JavaScript code.
- Community Packages:** React Native's documentation discusses the integration of community-contributed packages from npm. These packages enhance the capabilities of your app and cover areas such as navigation, animation, and more. The documentation explains how to install and use these packages.

6.0 Community and Ecosystem

A thriving community and ecosystem are pivotal for a tech framework's success. Communities facilitate collaboration, knowledge sharing, and innovation among developers and users, leading to rapid growth and improvement. A robust ecosystem extends the framework's capabilities through third-party tools and integrations, broadening its appeal and usability. Together, they ensure a framework's longevity, sustainability, and adaptability, making it a trusted and dynamic solution in the ever-evolving tech landscape.

6.1 Community size

A thriving community and ecosystem are pivotal for a tech framework's success. Communities facilitate collaboration, knowledge sharing, and innovation among developers and users, leading to rapid growth and improvement. A robust ecosystem extends the framework's capabilities through third-party tools and integrations, broadening its appeal and usability. Together, they ensure a framework's longevity, sustainability, and adaptability, making it a trusted and dynamic solution in the ever-evolving tech landscape.

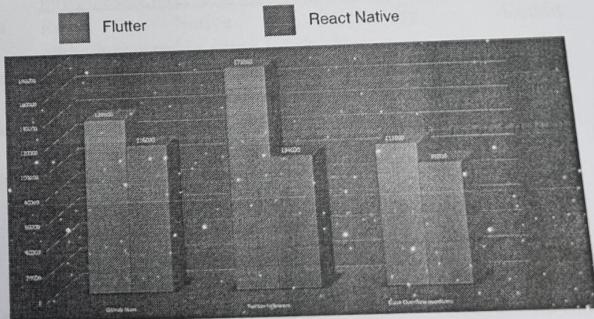


Fig 6.1 Community size

The bar graph shows the percentage of people who are using a filter and a reach native app. The percentage of people who are

using a filter is higher than the percentage of people who are using a react native app.

The data suggests that filters are more popular than react native apps. This could be due to a number of factors, such as the fact that filters are easier to use or that they offer more features. It is also possible that filters are more popular because they are more widely available.

It is important to note that the data does not show the reasons why people are choosing to use filters or react native apps. It is also worth noting that the data is based on a small sample size, so it may not be representative of the population as a whole. Overall, the graph suggests that filters are more popular than each native apps.

7.0 Implementation

To get hands-on experience on how both these frameworks work and perform and how much time does it take to implement them in some projects, we developed a BMI calculator using both the frameworks.

7.1 Flutter Implementation

This section will discuss the implementation of the BMI calculator app using Flutter. We will start by creating a new Flutter project, then we will create the components for the inputs and then apply formulas to calculate results.

Steps we followed to create the flutter project:

- We installed Flutter on our computer, ensuring that our development environment was ready for mobile app development.
- After installation, we verified the Flutter installation's status by running the flutter doctor command in our terminal. This helped us identify any missing dependencies or configuration issues.
- To kickstart our Flutter project, we used the flutter create project_name command, where "project_name" was replaced with the desired name of our project.
- We then navigated to the project directory using the cd project_name command, positioning ourselves within our project's workspace.
- We proceeded to open the project in our preferred code editor, where we would write and manage our app's code.
- To see our app in action, we ran the command flutter run in the terminal, allowing us to test and debug our app on an emulator or physical device.

7.2 React Native Implementation

This section will discuss the implementation of the BMI calculator app using React Native. We will start by creating a new Flutter project, then we will create the components for the inputs and then apply formulas to calculate results.

Steps we followed to create the project:

- We ensured that Node.js and npm were installed on our computer, as they were prerequisites for React Native development.
- To facilitate React Native project creation, we globally installed the React Native CLI using the npm install -g react-native-cli command.
- Using the react-native init project_name command, we initiated the creation of a new React Native project. We replaced "project_name" with the chosen name for our project.
- We navigated into the project directory by running cd project_name in our terminal, positioning ourselves within the project workspace.
- Our chosen code editor was used to open the newly created

project, where we began writing and managing our app's code.

- To view and test our app on an emulator or physical device, we used the npx react-native run-android command for Android and npx react-native run-ios for iOS.

Aspect	Flutter	React Native
File Size(MB)	546	364
File Size on Disk(MB)	546	410
Components	2 TextFields + 3 SizedBox + 1 ElevatedButton + 1 Text	2 TextInputs + 1 TouchableOpacity + 1 Text
Lines of Code	119	146
Apk Size(MB)	103	112
Development time(hrs)	1	1

7.3 Observations from the implementation

8.0 Conclusion

In conclusion, both Flutter and React Native are powerful frameworks that help build native cross platform apps with unique strengths and considerations and almost the same requirements. While both have their merits and very limited demerits, Flutter seems to be the more reliable and more compelling choice amongst the two.

Its utilisation of Dart programming language coupled with the support from Google and the ever growing community makes it an undeniable superior framework with regards to building a native cross platform mobile app. Moreover, over 46% of app developers use Flutter which is a good sign of its growth despite being a newer framework than React Native.

9.0 References

- React Native vs Flutter: <https://hackr.io/blog/react-native-vs-flutter>
- Flutter vs React Native: <https://www.interviewbit.com/blog/flutter-vs-react-native/>
- React Native Community Reddit: <https://www.reddit.com/r/reactnative/>
- Flutter Community Reddit: <https://www.reddit.com/r/flutter/>
- Android developer Community Reddit: <https://www.reddit.com/r/androiddev/>
- Choosing the best framework: <https://www.cleveroad.com/blog/flutter-vs-react-native-choosing-the-best-framework-for-cross-platform-mobile-development/#:~:text=How%20is%20React%20Native%20different%20from%20React%20Native%20in%20the%20operating%20system.>
- React Native documentation: <https://reactnative.dev/docs/getting-started>
- Flutter documentation: <https://docs.flutter.dev/>

STATEMENT OF PURPOSE

[Assignment No. 5]

Raghav Rathi

Fall 2024

University of Toronto

I am writing to express my strong interest in the Master of Science in Applied Computing program at the University of Toronto. It is with great enthusiasm that I submit my application for admission to this prestigious program, as I believe it aligns perfectly with my academic background and career aspirations.

From a young age, I have been captivated by the ever-evolving field of computer science. My undergraduate studies in [Your Undergraduate Major] at [Your Undergraduate University] laid the foundation for my passion and proficiency in programming, algorithms, and software development. During my academic journey, I had the opportunity to engage in various hands-on projects and research endeavors, which deepened my understanding of the practical applications of computer science.

What sets the University of Toronto apart in my eyes is its commitment to fostering innovation and research excellence. The Master of Science in Applied Computing program's interdisciplinary approach, renowned faculty, and cutting-edge facilities make it an ideal environment for me to further develop my skills and contribute to groundbreaking advancements in the field.

One aspect that particularly resonates with me is the program's focus on applied research. I am eager to collaborate with esteemed professors and fellow students on projects that bridge the gap between theory and real-world solutions. I am particularly drawn to research areas such as [Mention Specific Research Area of Interest], and I am excited about the opportunity to explore these topics within the program.

Beyond academics, I am inspired by the vibrant and diverse community that the University of Toronto offers. I am eager to engage with fellow students from various backgrounds, sharing ideas and experiences that will broaden my perspective and enrich my learning journey.

Upon completion of the Master of Science in Applied Computing program at the University of Toronto, I aspire to leverage the knowledge and expertise gained to make a meaningful impact in the tech industry. Whether through contributing to innovative startups, collaborating on research that addresses pressing societal challenges, or mentoring the next generation of computer scientists, I am committed to advancing the field and giving back to the community.

26%

In conclusion, I am truly excited about the prospect of joining the University of Toronto's Master of Science in Applied Computing program. I am confident that this program will provide me with the academic rigor, research opportunities, and professional development needed to achieve my goals. I eagerly anticipate the chance to contribute to the university's esteemed academic community and be a part of its legacy of excellence.

Thank you for considering my application. I look forward to the opportunity to further discuss my candidacy.

STATEMENT OF PURPOSE

I'm excited to apply for the master of science in Applied Computing program at the University of Toronto because I'm passionate about computers especially when it comes to creating video games.

During my undergraduate studies in Thadomal Shahani Engineering College, I discovered my love for game development. I enjoy coding and designing games that entertain and challenge players. Though I have created small projects only, but which have been enjoyed by my friends and family members too. Also point to be noted I have done degree of computers from here where we were thought core subject and web technologies. I got interest in it by exploring different fields.

My particular interest is in video game development, and I'm eager to explore this field further during my studies. I want to understand how to create immersive and captivating games that bring joy to people's lives. But I have keen interest in FPS games and Intellectual or puzzle solving games.

One of the things, I admire about the University of Toronto is its diverse and dynamic community. Putting people from various backgrounds give broad view on different topics and exposure to new tech / fields and will also give the new perspectives on video game development.

After completing this program, my goal is to use my video game development skills to create unique and memorable gaming experience. Whether it's designing for entertainment or education. I'm determined to make a positive impact through the world of gaming.

To sum it up, I'm thrilled about the opportunity to join the University of Toronto's program and explore the exciting world of video game development. I know this program will equip me with the knowledge and skill to achieve my dreams.

Thank you for considering my application.

Sincerely,

Reaghan Rathn

Name : Raghav. V. Rathai
Batch : C34
Roll no : 2113208
Date : 21 July 2023



Time Management

K

1. Yes
2. Yes
3. Yes
4. No
5. Yes
6. No
7. No
8. Yes
9. Yes
10. Yes
11. Yes
12. No
13. Yes
14. No
15. Yes
16. No
17. Yes
18. Yes
19. No
20. Yes
21. No.

Roeghan
C34
2113208

Leadership Questionnaire

Group

Name

Directions: the following items describe aspects of leadership behavior. Respond to each item according to the way you would most likely act if you were the leader of a work group. Circle whether you would most likely behave in the described way always (A), frequently (F), occasionally (O), seldom (S), or never (N). Once the test is completed, go back to number 2 under Implementation.

A	F	O	S	N	1.	I would likely act as the spokesperson of the group.
A	F	O	S	(N)	2.	I would encourage overtime work.
A	(F)	O	S	N	3.	I would allow members complete freedom in their work.
A	F	(O)	S	N	4.	I would encourage the use of uniform procedures.
A	(F)	O	S	N	5.	I would permit the members to use their own judgment in solving problems.
A	F	(O)	S	N	6.	I would stress being ahead of competing groups.
A	F	(O)	S	N	7.	I would speak as representative of the group.
A	F	(O)	S	N	8.	I would needle members for greater effort.
(A)	F	O	S	N	9.	I would try out my ideas in the group.
A	(F)	O	S	N	10.	I would let the members do their work the way they think best.
A	(F)	O	S	N	11.	I would be working hard for a promotion.
A	F	(O)	S	N	12.	I would tolerate postponement and uncertainty.
A	(F)	O	S	N	13.	I would speak for the group if there were visitors present.
A	F	O	(S)	N	14.	I would keep the work moving at a rapid pace.
A	F	(O)	S	N	15.	I would turn the members loose on a job and let them go to it.
A	(F)	O	S	N	16.	I would settle conflicts when they occur in the group.
A	F	O	(S)	N	17.	I would get swamped by details.
A	(F)	O	S	N	18.	I would represent the group at outside meetings.

A	F	O	S	N		19.	I would be reluctant to allow the members any freedom of action.
A	F	O	S	N		20.	I would decide what should be done and how it should be done.
A	F	O	S	N		21.	I would push for increased production.
A	F	O	S	N		22.	I would let some members have authority which I could keep.
A	F	O	S	N		23.	Things would easily turn out as I had predicted.
A	F	O	S	N		24.	I would allow the group a high degree of initiative.
A	F	O	S	N		25.	I would assign group members for to particular tasks.
A	F	O	S	N		26.	I would be willing to make changes.
A	F	O	S	N		27.	I would ask the members to work harder.
A	F	O	S	N		28.	I would trust the group members to exercise good judgement.
A	F	O	S	N		29.	I would schedule the work to be done.
A	F	O	S	N		30.	I would refuse to explain my actions.
A	F	O	S	N		31.	I would persuade others that my ideas are to their advantage.
A	F	O	S	N		32.	I would permit the group to set its own pace.
A	F	O	S	N		33.	I would urge the group to beat its previous record.
A	F	O	S	N		34.	I would act without consulting the group.
A	F	O	S	N		35.	I would ask that group members follow standard rules and regulations.

T _____

P _____

Raghav. V. Rathore
C34
21/3/2008
✓

Opinion Survey on Conflict Management

Please read each statement given below and write a number from 1 to 5 in the space on the left hand side of the statement to indicate your opinion.

- Write 1 if you strongly disagree with the statement.
- Write 2 if you disagree with the statement.
- Write 3 if you half agree and half disagree with the statement.
- Write 4 if you agree with the statement.
- Write 5 if you strongly agree with the statement.

1. Conflicts are inevitable in organizations and nothing can be done about them. 3
2. The best strategy is to avoid conflict situations. 2
3. Conflict is like a problem; we have to find the causes and try to find solutions. 5
4. Conflict can be solved only if one shows one's strength to the other party. 2
5. In a conflict situation both the parties have to give up something in order to reach a situation. 5
6. A third party should be asked to give a solution to a difficult conflict. 3
7. It is better to give concessions to the opponent group to win their confidence. 5
8. The best way to deal with conflicts is to withdraw from the scene for sometime. 5
9. It is better to lie low and live with conflict. 2
10. In a conflict situation one party should leave to avert unpleasantness. 3
11. Conflict management needs an involved process of joint exploration for solution(s). 5
12. In most conflicts one should fight out the solution. 3
13. Compromise is the best strategy in managing a conflict. 3
14. When two parties are deeply in conflict, arbitration by an acceptable outside party may be very helpful. 4
15. Accepting a few demands of the opponent group may help in solving the conflicts. 5
16. If one waits for sometime and does not attempt to solve the problem, the conflicts will get defused and resolved in the due course of time. 5
17. It is foolish to be bothered by conflicts; they are there and we may better live with them. 2
18. If a group interacts with the other group only on the necessary and limited dimensions, conflicts can be managed. 3
19. Conflicts can be solved if the conflicts parties understand each other, and jointly search alternative solutions. 5
20. The more powerful you are, the more effectively you can resolve the conflicts. 5
21. If conflicting parties accept a part of each other's demands, conflict can be resolved. 5
22. Difficult conflicts can be resolved by an impartial arbitration acceptable to both the conflicting group, so that conflicts can be effectively resolved. 3
23. It is better to buy peace for sometime even by acceding to some demands of the conflicting group, so that conflicts can be effectively resolved. 5
24. Waiting for sometime to let the emotions subside helps in resolving the major problems. 5

Scoring key:

Conflict management style
Items

1. Resignation
2. Withdrawal
3. Appeasement
4. Defusion
5. Confrontation
6. Arbitration
7. Compromise
8. Negotiation

10	1, 9, 17
8	2, 10, 18
15	7, 15, 23
15	8, 16, 24
9	4, 12, 20
10	6, 14, 22
13	5, 13, 21
14	3, 11, 19

Pagon
C36
2/13/2008



Entrepreneurial Self-Assessment Survey

This is not a test! This survey is for your personal information.
Please answer each of the following questions as honestly as possible.

Strongly Agree
5

4

Somewhat Agree
3

2

Strongly Disagree
1

1. I am willing to work 50 hours or more per week regularly.
2. My family will support my going into business.
3. I am willing to accept both financial and career risks when necessary.
4. I don't need all the fringe benefits provided by conventional employment.
5. I would like to take full responsibility for the successes and failures of my business.
6. I would experience more financial success by operating my own business.
7. I feel a great deal of pride when I complete a project successfully.
8. I have a high energy level that can be maintained over a long time.
9. I enjoy controlling my own work assignments & making all decisions affecting my work.
10. I believe that I am primarily responsible for my own successes and failures.
11. I have a strong desire to achieve positive results even when it requires a great deal of additional effort.
12. I have a good understanding of how to manage a business.
13. I can function in ambiguous situations.
14. One or both of my parents were entrepreneurs.
15. I believe that my abilities and skills are greater than those of most of my coworkers.
16. People trust me and consider me honest and reliable.
17. I always try to complete every project I start, regardless of obstacles and difficulties.
18. I am willing to do something even when other people laugh or belittle me for doing it.
19. I can make decisions quickly.
20. I have a good network of friends, professionals, and business acquaintances.

TOTAL 60

Total the numbers you placed before the statements and enter the total in the space provided.

Characteristics of an Entrepreneur

The following list describes some common characteristics of an entrepreneur. The number(s) after each characteristic indicates the related statement(s) in the assessment form. This list interprets the form qualitatively to mean that arriving at a conclusive portrait of a typical entrepreneur is very difficult. Therefore, you may score low on the assessment and still succeed as an entrepreneur.

Works Hard (Statements 1 & 8)

Self-employment requires a great deal of time and effort. The entrepreneur must perform a wide variety of time-consuming tasks. 77% of all entrepreneurs report working 50 hours or more per week, and 54% say that they work more than 60 hours per week. Such a time commitment requires that you have a high energy level.

Wants Financial Success (Statement 6)

A primary reason that most entrepreneurs have for going into business is to achieve financial success. If you want to be an entrepreneur, you need to establish a reasonable financial goal that you want to achieve through self-employment. This goal will help you measure how well you are doing in fulfilling your personal needs through an entrepreneurial career.

Has Family Support (Statement 2)

A successful entrepreneur needs family support. If you are married, your spouse must believe in your business because it will require that both of you sacrifice time and money. The stress may create disruptions in family relationships. If you have children, they will need encouragement in understanding your need to spend so much time away from the family. The more positive support you receive from your family, the more you can concentrate on making the business a success.

Is Energetic (Statements 1 & 8)

Self-employment requires long work hours. You will frequently be unable to control the number of hours required to fulfill all the necessary tasks. The entrepreneur must have a high energy level to respond to the job's demands.

Has an Internal "Locus of Control", (Statement 10)

Successful entrepreneurs have an internal *locus of control* or inner sense of responsibility for the outcome of a venture. To be an entrepreneur, you should have a strong sense of being a "victor" who is responsible for your actions. If, however, you often consider yourself a "victim" and blame other people, bad luck, or difficult circumstances for your failures, entrepreneurship might not be the right career move for you.

Takes Risks (Statement 3)

Entrepreneurs are risk takers. They risk their careers, time and money in order to make a success of their businesses. To be successful in self-employment, you should feel comfortable taking reasonable risks.

Sacrifices Employment Benefits (Statement 4)

One of the major realities of self-employment is that you won't receive a regular paycheck. You pay for your own fringe benefits. A nice office, secretarial assistance, equipment and other features of employment you have grown to expect are no longer available unless you provide them for yourself.

Has a Need to Achieve (Statements 7 & 11)

Entrepreneurs have a strong need for achievement. They strive to excel and accomplish objectives that are quite high. You should be willing to set high goals for yourself and enjoy striving to achieve those goals.

Has Business Experience (Statement 12)

An entrepreneur should have extensive business experience to be successful. General management experience is beneficial because an entrepreneur should know something about all types of management. Formal training and education in management also are helpful.

Is Independent (Statements 5 & 9)

Entrepreneurs like to be independent and in control of situations. Many people who become self-employed consider the opportunity to be their own boss as one of the major benefits of self-employment. Although being independent may not be a major concern for you, it is certainly an aspect of self-employment that you need to feel

able with. If you cannot afford to hire other employees when you begin your business, you may at first be a self-employed person.

Self-employed Parent as a Role Model (Statement 14)

Research has shown that entrepreneurs are more likely to have a parent who is self-employed. A parent's education and knowledge about operating a business can contribute to an entrepreneur's success.

Self-confidence (Statements 10, 15, and 18)

The most important characteristic of entrepreneurs is self-confidence. This factor is particularly important when you face challenges and difficulties with your business. You need to believe in yourself. Your belief will help you overcome the problems that inevitably affect all self-employed persons at some point in their careers.

Integrity (Statement 16)

Business owners often cite honesty and integrity as characteristics of entrepreneurs. Customers do not want to deal with business owners who are dishonest and unethical. You should feel positive about your ethical treatment of people and be committed to conducting your business with the utmost integrity.

Determination (Statement 17)

One of the most important characteristics of entrepreneurs is determination. This trait is closely related to self-confidence. The more you believe in yourself, the more likely you are to continue to struggle for success when faced with tremendous obstacles. You need determination in order to overcome the problems that beset every new venture.

Adapt to Change (Statement 13 and 19)

A new business changes rapidly, so an entrepreneur must be able to adapt to change. Two primary skills are required for adaptation to change: the capacity to solve problems, and the ability to make quick decisions. Another skill is the ability to learn from your mistakes.

Has a Good Network of Professionals (Statement 20)

An entrepreneur has a good network of professionals. This network provides access to those who can be consulted for advice, information, and referrals. You should have an extensive network of professionals to whom you can turn for assistance.

Assessment

You have outstanding ability to be an entrepreneur.

You have satisfactory ability to be an entrepreneur.

Self-employment may not be an appropriate career for you.

You should probably avoid entrepreneurship.

Score
80 - 100
70 - 79
60 - 59
- 39

Raghav Rathi
21132008
C34

P 23
—
44

APTITUDE TEST
[Assignment no. 8]

Exercise 1: Grammar:

Instructions for questions 1 - 11: From among the given alternatives choose the one that best completes the sentence.

1. Truth and honesty ____ the best policy.

(1) are (2) is

2. Fish and rice ____ his favourite food.

(1) are (2) is

3. Your car and mine ____ both at the door.

(1) are (2) is

4. Neither Maradona nor the referee ____ guilty.

(1) are (2) is

5. The garage with the car ____ sold.

(1) were (2) was

6. The state of affairs in the city ____ such as to cause disturbance to normal life.

(1) were (2) was

7. Each of the alleged terrorists ____ arrested.

(1) were (2) was

8. Neither of the alibis ____ convincing.

(1) are (2) is

9. The military ____ still under the command of the ousted head. ✓
(1) are (2) is

10. Gymnastics ____ a difficult sport to master. ✓
(1) is (2) are

11. A large number of rioters ____ arrested. ✓
(1) was (2) were

Instructions for questions 12 - 17: Determine whether the given sentences are grammatically correct (option 1) or incorrect (option 2).

12. My father and my brother is in the office. incorrect ✓

13. Each of you have been allotted separate plots of land. incorrect ✓

14. Ten chocolates costs five rupees. incorrect ✓

15. The first few pages of the text has been copied. incorrect ✓

16. Which are your pair of scissors? incorrect ✓

17. Neither he nor his friend know how to play the game. incorrect ✓

Instructions for questions 18 - 24: Identify the correct sentence.

18. (1) Bread and pickle are not a good combination. ✗
(2) Bread and pickle is not a good combination.
(3) Bread and pickle be not a good combination.

19. (1) Both the government and the opposition is interested in governance. ✗
(2) Neither the government nor the opposition are interested in governance.
(3) Neither the government nor the opposition is interested in governance.

20. (1) not only was the CEO but his executives also insulted. ✓
(2) The CEO as well as his executives was insulted.
(3) The CEO as well as his executives are insulted.

21. (1) Neither Raj nor his cousin were invited for the big event. ✗
(2) Neither Raj nor his cousin was invited for the big event.
(3) Neither Raj nor his cousin are invited for the big event.

22. (1) Whose is this dirty pair of socks?
(2) Whose are these dirty pair of socks? ✗

- (3) Whose were these dirty pair of socks?
23. (1) According to him, fifty dollars are not a neat sum.
 (2) According to him, hundred dollars be a neat sum.
 (3) According to him, hundred dollars is a neat sum. ✓
24. (1) None of them was present.
 (2) Neither of them were present.
 (3) Neither she nor her friend were present. ✗

Exercise 2: Analogies:

Instructions for questions 1 to 7: From the following words, identify which word will make a similar analogous relationship as the first pair.

1. PLANTS : BOTANY :: INSECTS :

- (1) Epidemiology (2) Entomology (3) Helminthology (4) Carpology ✓

2. PULP : PAPER :: HEMP :

- (1) Basket (2) Yarn (3) Cotton (4) Rope ✓

3. HORSE : NEIGH :: HYENA :

- (1) Chatter (2) Talk (3) Laugh (4) Howl (5) Roar ✗

4. BREW : BEER :: DISTILL :

- (1) Milk (2) Oil (3) Butter (4) Bread ✗

5. NEEDLE : KNIT :: LOOM :

- (1) Weave (2) Sew (3) Thimble (4) Stitch (5) Darn ✗

6. COHERENT : CONSISTENT :: IRATE :

- (1) Rage (2) Irritated (3) Unreasonable (4) Cantankerous (5) Hostile ✗

7. ELEPHANT : CALF :: FISH :

- (1) Fawn (2) Fry (3) Cub (4) Roe (5) Fillet ✓

Instructions for question 8 to 13: Each of the following questions consists of two capitalized words that have a certain relationship to each other, followed by a certain pair of words. Choose the pair that is RELATED to each other in the same way as the capitalized pair.

8. SATURN : PLANET

- (1) Star : Sun (2) Fig : Apple (3) Moon : Satellite ✗
(4) Europe : Asia (5) Comet : Meteor

9. INDEX: INDICES

- (1) Object : Symbol (2) Male : Female (3) Worker : Tool ✓
(4) Female : Male (5) Singular : Plural

10. WHITE: PEACE

- (1) Object : Symbol (2) Male : Female (3) Singular : Plural ✗
(4) Female : Male (5) Worker : Tool

11. ANVIL: SMITH

- (1) Fire : Mason (2) Cement : Mortar (3) Shoes : Cobbler ✓
(4) Wickets : Cricketer (5) Hammer : Carpenter

12. FOX: VIXEN

- (1) Goose : Gander (2) Drake : Duck (3) Thoroughbred : Stallion ✗
(4) Horse : Colt (5) Sheep : Coyote

13. FISH : PISCES

- (1) Air : Libra (2) Elements : Clue (3) Crab : Cancer ✓
(4) Lion : Capricorn (5) Libra : October

Exercise 3: Jumbled Sentences:

Instructions for questions 1 – 3: Each of the questions below consists of a paragraph in which the first and last sentences are identified. Choose the option that has the most logical order of the intermediate sentences.

1. A. World War II, was a global military conflict, the joining of what had initially been two separate conflicts.
B. The other began in Europe in 1939 with the German invasion of Poland.

- C. This global conflict split the majority of the world's nations into two opposing military alliances: the Allies and the Axis Powers.
D. The first began in Asia in 1937 as the Second Sino-Japanese War.
E. The Allies included USA, UK, France, Russia etc.
F. Whereas Axis powers included mainly Germany, Italy and Japan.

(1) DBCE (2) CEBD (3) BCDE (4) DBEC (5) CBED

2. A. Bal Gangadhar Tilak was an Indian nationalist, social reformer and freedom fighter.
B. "Swaraj is my birth right, and I shall have it!"
C. He is reverently addressed as Lokmanya.
D. He is known as "Father of the Indian unrest."
E. This quote of his is well-remembered in India even today.
F. It means "Beloved of the people" a title that is well deserved.

(1) BCDE (2) DBEC (3) DBCE (4) EBCD (5) CDEB

3. A. 'Chakra' is a Sanskrit term meaning circle or wheel.
B. They are considered to be a point or nexus of metaphysical and/ or biophysical energy of the human body.
C. Theories on chakras fit within systems that link the human body and mind into a single unit, sometimes called the body mind or 'namarupa'.
D. Chakras are commonly described as energy centers in the spine located at major branches of the human nervous system, beginning at the base of the spinal column and moving upwards to the top of the skull.
E. There is a wide literature on chakra models, philosophy, and lore that underpin many philosophical systems and spiritual energy practices, religious observance, and personal discipline.
F. These philosophical theories and models were first codified in Ancient India

(1) DBEC (2) DCEB (3) BCED (4) BECD (5) CBED

Instructions for questions 4 - 7: Each of the questions below consists of a set of labelled sentences. These sentences, when properly sequenced, form a coherent paragraph. Choose the most logical order of sentences from among the options.

4. A. According to this philosophy, acquiring the awareness of Brahmanavidya (direct perception or awareness of Reality) alleviates this deep source of suffering.
B. The Indian philosophy of Advaita Vedanta explains that we suffer as a avidya (ignorance) and maya (misconceived, misinterpreted views of Reality).
C. Only this awareness directly leads us to moksha (liberation).
D. Brahmanavidya, thus is an ancient system of Yoga and Philosophy, which helps eliminate suffering.

(1) ACDB (2) ADBC (3) BDCA (4) BACD (5) BADC

5. A. It describes the lives of the young March sisters.
B. 'Little women' is a beloved classic.
C. It is an ambition she is destined to fulfil.
D. Meg is a young girl who settles into quiet domesticity.
E. However Jo, her sister, longs to be a famous author.

(1) ABCDE (2) BADEC (3) BCDAE (4) BAECD

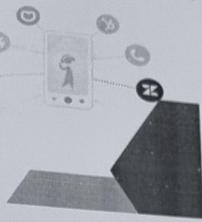
6. A. Unfortunately it is not limited to only the geriatric age group.
B. Children as young as 10 years may have this disease.
C. One of the commonest chronic diseases in old age is Diabetes.
D. However in them, it is usually of type 1.

(1) CBDA (2) CABD (3) CADB (4) BADC

7. A. Rising sea levels swamped the coastal regions.
B. Forests replaced open woodlands and grasslands across the continent.
C. The Ice Age was ebbing.
D. About 12000 years ago, warmer, wetter weather was beginning to take hold.
E. As their habitats disappeared, so did the bison and the mammoth.

(1) CDABE (2) DCABE (3) DCBEA (4) BEDCA (5) ABDCE

CUSTOMER SUPPORT PORTAL USING FLUTTER



PRESENTED BY

- Vrajesh Hegde
- Hiten Dusseja
- Anuraag Dubey
- Om Avhad
- Ritika Gupta
- Sawan Sharma
- Raghav Rathi

B

TOPICS COVERED

- Introduction
- Aim
- Scope
- Why Flutter
- Need
- Plan
- Schedule
- Block Diagram
- List of Resources
- Budget
- Conclusion

Introduction

- Bridge between companies and customers is crucial for success
- Consumer expectations are constantly rising
- Innovative Flutter-based solution for customer support
- Versatile communication platform (chat, voice calls, video calls)
- Immediate access to comprehensive FAQs
- Enhances customer satisfaction and competitive advantage



Q1

Redefining customer company connections and engagement.

Creating a dynamic communication ecosystem for support.

Facilitating real-time chat, voice calls, and video calls.

Offering immediate access to a curated FAQ section.

Elevating overall customer experience and satisfaction.

Building customer-centric industry leader image.



FEATURES

- User-Friendly Interface
- Multi-Channel Communication
- User Authentication and Profiles
- Live Support Agent Queue
- FAQ Section
- Notification System
- Security and Compliance
- Scalability and Maintenance
- Testing and Quality Assurance

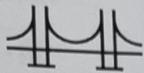


W5

07
4
am

DO OF THE PROJECT

- Enhanced Customer Engagement and Satisfaction
- Improved Accessibility
- Personalized Support
- Knowledge Base and FAQ Integration
- Data-Driven Insights



SCHEDULE

Milestone	Estimated Days
Project Initiation	5
Design and Planning	30
Development	40
Testing and quality assurance	20
Fixing bugs	10
Integration and optimization	20
User training and documentation	5
Implementing security measures	5
Deployment	2

WHY FLUTTER

- Flutter's Advantage
Cross-platform app development, cost-effective, convenient
- Rapid Development
Fast iteration, no UI wait time
- Performance & Scalability
High performance, available for multiple platforms
- Customizable Widgets
Extensive library for customization
- Community & Support
Large developer community
- Optimized User Experience
Smooth, fast, and consistent

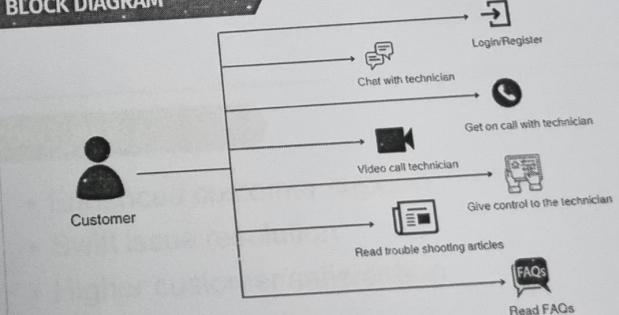


09
4
an

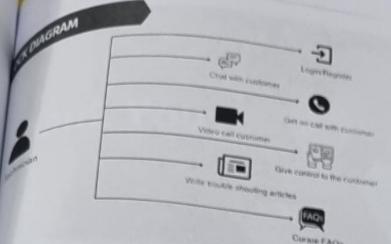
PLAN

- Provide user training and documentation
- Prioritize data security and compliance
- Validate app through User Acceptance Testing (UAT)
- Publish app and monitor performance
- Offer ongoing maintenance and updates

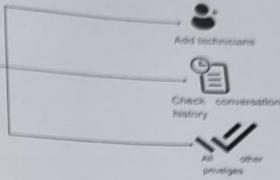
BLOCK DIAGRAM



BLOCK DIAGRAM



BLOCK DIAGRAM



RESOURCES



Flutter development tools



Software testing tools



Communication technologies



Version control tools



UI/UX design tools



Firebase subscription

RESOURCES(Optional)



Analytics and monitoring tools



Cloud tools



Hosting tools

BT

CONCLUSION

- Enhanced customer support
- Swift issue resolution
- Higher customer satisfaction
- Increased retention
- Stronger brand image
- Anticipated return on investment

Item Description	Total Cost (In Rs)
Developer license on company's name	5000
Feature services	12,000/year
Content writing	8000
Security and compliance	17,000
Maintenance	13,000/year
Cloud hosting(optional)	15,000/year
Total	70,000

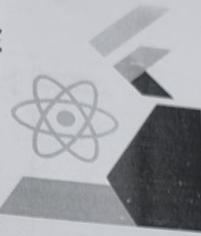
6x

Thank You



09
4
2021

COMPARITIVE STUDY ON FLUTTER AND REACT NATIVE



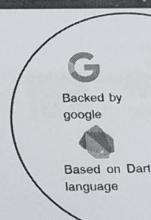
PRESENTED BY

- Vrajesh Hegde
- Hiten Dusseja
- Anuraag Dubey
- Om Avhad
- Ritika Gupta
- Sawan Sharma
- Raghav Rathi

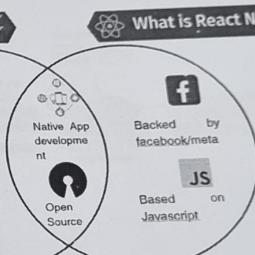
TOPICS COVERED

Introduction
Requirements
Architecture
Performance
Similarities
Differences
Implementation
Conclusion

What is Flutter?



What is React Native?



Flutter



Google Classroom

G Pay

Google Pay

Apps with React Native



FaceBook



Skype



Instagram



Uber Eats

Hardware Requirements

Hardware Requirements	Flutter	React Native
CPU	Multi-core, 32-bit or 64-bit (64-bit recommended)	Multi-core, 32-bit or 64-bit (64-bit recommended)
RAM	4 GB (8 GB recommended)	4 GB (8 GB recommended)
Disk Space	2 GB	5 GB
Operating System	Android, iOS	Android, iOS
Windows	Windows 10 or later (64-bit)	Windows 10.0.16299.0 or higher
MacOS	MacOS 10.14 or later	MacOS 10.12 or later
Linux	Modern Linux Distribution (64-bit)	Modern Linux Distribution

69

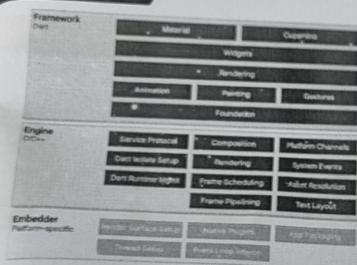
Requirements

	Flutter(required or not)	React Native (required or not)
UI/UX	✓	✗
Performance (CPU / GPU)	✓	✓
Memory Usage	✓	✓
Cross-platform (iOS/Android)	✗	✗

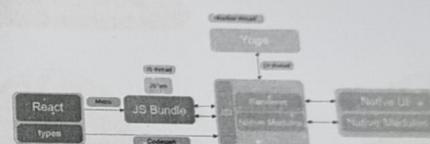
IDEs Available

Development Tooling/IDEs	Flutter	React Native
Official IDE	VS Code	React Native CLI
Other IDEs	Android Studio, Xcode	IntelliJ IDEA, WebStorm

Architecture



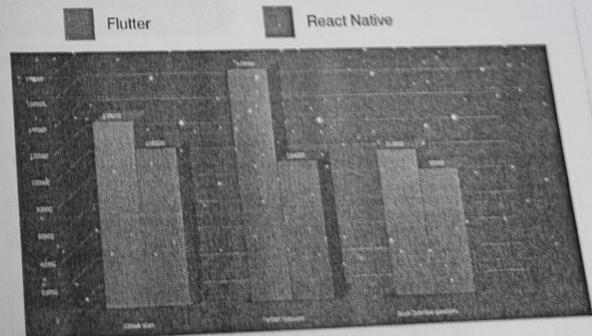
React Native Architecture



Performance

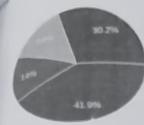
Flutter	React Native
Compile and Run	Compiles to native code
UI rendering	Has its own rendering engine
C hot reload	Very fast
Native module access	Direct access through platform-specific channels
Third-party libraries	More consistent ecosystem
UX	More consistent UI across platforms

Popularity



Similarity(Survey)

In how many new frameworks have you used before?



- Flutter
- React Native
- Both Flutter and React Native
- Neither Flutter nor React Native

Performance(Survey)

In your experience or perception, which framework do you believe offers better performance in terms of development and accomplishments. Flutter or React Native?



Evaluation(Survey)



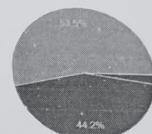
Flutter

- Poor
- Decent
- Excellent



React Native

Community support(Survey)



Flutter

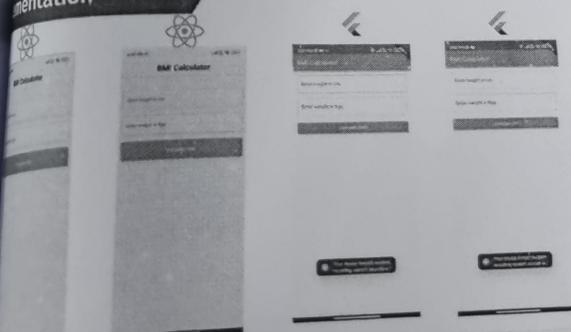
- Poor
- Decent
- Excellent



React Native

- Poor
- Decent
- Excellent

Implementation



Observations(Implementation)

Aspect	Flutter	React Native
File Size(MB)	546	364
File Size on Disk(MB)	546	410
Components	2 TextFields + 3 SizedBox + 1 ElevatedButton + 1 Text	2 TextInputs + 1 TouchableOpacity + 1 Text
Lines of Code	119	146
Apk Size(MB)	103	112
Development time(hrs)	1	1

Particulars matter in life
and business.
Marketing is no exception.
That's why we've
partnered with the experts
in digital marketing to bring
you the best of both worlds.
High Performance. It
means you'll get the best of both
marketing worlds, with proven
strategies and tools from
the experts in digital marketing,
and the expertise of our own
marketing team.



- **Marketing Proficiency:** Gain
access to world-class marketing
strategies and tools.
- **Component Reusability:** It
means you can reuse components
from one project to another.
- **Extreme Customization:** Access
multiple strategies & wide range of
marketing techniques for extensive
personalization.
- **Platform-Specific Design:** If
you're working with a specific platform,
we'll ensure your design is platform-
specific design requirements.

Thank You



CORPORATE ETHICS
[Assignment No. 6]



What are the issues of integrity, ethics and law present in the case study?

Bad quality product being marketed to most of the customers is one of the biggest issue of product integrity. Poor control over different manufacturing units makes the product inconsistent and would result in loss of customers trust in the product.

As mentioned, it was alleged by one of the analyst that the company already had information about some people falling ill after consumption of their products. Emphatically they should have published a statement regarding some people falling ill due to their product and they are looking into the matter. It would then had made the customers aware and they could make their choice to still consume or not.

Loké faced a lot of issues of compliance to law one of the common

allegation on them were of misleading advertising. They even had to eventually abide by the Supreme Court decision and reduce caffeine content by 50%. In the 1970s the Food and Drug Administration (FDA) ruled that one of the ingredients in Coke, Saccharin was potential source of cancer. Coke also got into trouble due to their exclusive school contracts due to them exposing teenagers to harmful soft drinks promoted by the schools.

What options did the soft drink company and the government have?

With preventive measures not being implemented with great execution the first and foremost damage control option the soft drink company had was to recall all their products back avoiding any further escalation. Then initiate an investigation into knowing the root cause of adulterated products reaching the markets. Upon coming to a conclusion they needed to communicate back to their consumers that such events would never be repeated also that the product formula in itself was not the reason for people falling.

being young children falling ill in the various parts of country to stop further harm the government has to temporarily ban the soft drink companies whose cause is responsible for it until the cause is investigated. The plants need to be regulated and ensure they were not faulty. Inspectors needed to ensure they were complying with laws regarding food safety. Stay put on the misleading advertisements made by the soft drink companies while making the general public aware about the harms of soft drink.

If you were the decision maker when decision would you have taken and why?

Upon receiving the early information about some people falling ill after consuming our products I would have initiated an investigation to know if our products have any contamination by getting samples of products from the areas where incidents happened. This would help in getting at the end most of the contamination. A statement

could be put forth about the possible contamination from the customers and not to safeguard our product from the recall of contamination from the markets.

The next task will be to ensure such events do not repeat itself. The staff needs to undergo safety training about hazards that can contaminate the product. A system can be put in place for the workers to notify the higher authorities about any contamination from producers. Also will implement strict quality checking protocols for externally sourced materials.

Further to bring back my lost customs trust will take responsibility and also compensate anybody who has trouble led by our products work alongside the government of the safety and that their regulations are being followed to again enter this market.

COMPARATIVE STUDY ON FLUTTER AND REACT NATIVE

Introduction

- 1.1 What is Flutter
- 1.2 What is React Native
- 1.3 Background of Flutter
- 1.4 Background of React Native
- 1.5 Famous apps created with Flutter
- 1.6 Famous apps created with React Native

FL

Requirements

- 2.1 Hardware requirements for setting up development environment
- 2.2 Software requirements for app development using both frameworks
- 2.3 Integrated Development Environment (IDE) available for Flutter and React Native

Architecture

- 3.1 Flutter Architecture
- 3.2 React Native Architecture

Components

- 4.1 Comparison of components in Flutter and React Native

Performance

Language and Documentation

- 1 Language
- 2 Documentation

Community and Ecosystem

- 1.1 Community size and engagement
- 1.2 Popularity.

Conclusion