IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ADVANCEMENTS IN HEALTH MONITORING THROUGH IOT USING FLUTTER AND SPRING BOOT: AN OVERVIEW

[1]Nandini Prasad KS, [2]Adnan Khan, [3]Deep Kumar P Khanra, [4]Gagandeep Bhat and [5]Kedarnath Doddamani
[1]Professor, Department of Information Science and Engineering,
Dayananda Sagar Academy of Technology and Management, Bengaluru, India-560082
[2,3,4,5,]Student, Department of Information Science and Engineering,
Dayananda Sagar Academy of Technology and Management, Bengaluru, India-560082

Abstract**—**Over the span of the past decade, a significant body of research has been dedicated to the progressive evolution of healthcare services through cutting-edge technological advancements. Particularly noteworthy in this transformative landscape is the Internet of Things (IoT), showcasing its potential by seamlessly connecting a diverse array of medical devices, sensors, and healthcare professionals. This interconnected ecosystem has not only facilitated the delivery of high-quality medical services in remote settings but has also ushered in commendable enhancements in patient safety, instigated cost-effectiveness within the healthcare paradigm, broadened access to medical services, and elevated operational efficiency across the expansive healthcare sector. This contemporaneous study serves as a comprehensive and up-to-the minute compendium, offering a panoramic overview of the manifold applications of IoT-based technologies in healthcare, commonly denoted as HIoT. The study diligently traces the evolutionary trajectory of HIoT applications, engaging in a meticulous analysis of enabling technologies, the diverse spectrum of healthcare services impacted, and innovative solutions meticulously crafted to confront the diverse challenges prevalent in the healthcare domain. Furthermore, the study delves into a nuanced exploration of potential challenges and issues inherent in the HIoT system, encompassing the complex interplay of technological, ethical, and regulatory considerations. In summation, this expansive and thorough study stands as an invaluable reservoir of knowledge, providing not just insights but a comprehensive understanding of the multifaceted realms of HIoT applications. It is meticulously designed to not only inform but also guide and inspire future researchers, cultivating a deeper understanding for those eager to contribute to the ongoing advancement of this dynamic, rapidly evolving, and critically important field at the intersection of healthcare and technology. Beyond the tangible benefits mentioned earlier, HIoT has opened avenues for personalized medicine, real-time health monitoring, and predictive analytics.

Index Terms**—**IoT, Flutter, SpringBoot, Healthcare, HIoT, sensors, real time, health monitoring .

I. INTRODUCTION

The increasing demand and intense competition in industries have prompted a shift from mass production to meet the requirements of mass personalization. Industry 4.0 technologies play a crucial role in fulfilling these demands, where smart products, within the ever-expanding realm of personalization, offer tailored responses to individual customers. Despite this trend, conventional appliances are still widely used, creating a need for changes with minimal disruption to established production lines. The Internet of Things (IoT) emerges as a potent tool for real-time data collection, providing valuable insights and enabling cost-effective predictive maintenance, thereby ushering in a data-driven approach for added value.

An Internet of Things (IoT) platform comprises a collection of technologically empowered entities. These encompass tangible smart devices such as sensors, actuators, smart tags alongside interconnected software services and systems. These components collaborate seamlessly to facilitate integrated functionality. This integration of sensors, lightweight computation, and various wireless technologies empowers individuals to engage more deeply with their surroundings. IoT platforms, designed to manage and control diverse devices, vary in architecture and design due to their wide range of applications.

These platforms consist of millions of intelligent devices and software services collaborating harmoniously to provide dynamic sensing and sophisticated reasoning. The creation and upkeep of software elements, mobile apps, and foundational services are essential aspects of this orchestration within an IoT platform follow typical information systems project lifecycles, while hardware components require a separate lifecycle to ensure seamless communication with software components.

II. RELATED WORKS

Bikash Pradhan et.al did a comprehensive study on IoT Based Applications in Healthcare Devices[1]. The significance of this study lies in its exploration of the integration of Distributed Computing Environment (DCE) and Radio Frequency Identification (RFID) with Zigbee protocols, marking a crucial advancement in healthcare operations. This fusion not only ensures the maintenance of information integrity but also empowers healthcare providers to efficiently locate and monitor equipment. The utilization of Zigbee protocols, coupled with RFID, enables rapid tracking, thereby elevating operational efficiency. However, it is essential to acknowledge potential concerns, such as unauthorized access and the potential compromise of sensitive patient information associated. In conclusion, this integrated approach represents a noteworthy leap forward in enhancing overall healthcare processes.

Mahdi Fahmideh and Didar Zowghi surveyed on exploration of IoT platform development[3].Their findings explore the synergy of Edge and Cloud Integration with MQTT in IoT applications. This integration is instrumental in elevating data processing capabilities, enabling real-time analysis and fostering agile decision-making. The lightweight messaging of MQTT ensures reliable communication among IoT devices, despite acknowledged challenges like heightened complexity, potential latency, and security concerns related to message interception. In summary, the integrated approach proves effective for optimizing real-time data processing; however, the study emphasizes the importance of addressing associated challenges. Future research should focus on mitigating these challenges to further enhance the practical implementation of this integrated approach.

Jashandeep Singh et.al integrated Flutter and Firebase in making interoperable platforms[10]. The authors focused on Flutter development and how programmers can engage in Dart programming, compiling code ahead of time to cater to native platform architectures. This methodology guarantees swift execution and offers a rich array of widgets for crafting diverse interface elements. It also highlights the notable strength of Flutter's tripartite testing system, showcasing its efficacy in reducing space usage and accelerating initial app launches. However, a crucial consideration surfaces on the potential challenge of prolonged build times, especially noticeable in more extensive applications. . In conclusion, this study underscores the dual nature of Flutter's testing system, emphasizing its strengths while recognizing the need for optimization.

S. R. Modugu and H. Farhat suggest that Spring Boot and REST APIs can be used in implementing IoT applications[9]. Spring Boot which is a Java-based open-source framework proves to be a formidable asset. Its efficiency becomes evident in the seamless integration of IoT device functionalities, exemplified by the facile creation of REST endpoints tailored for sensors. A notable forte lies in its adept handling of databases, while security measures, primarily token-based, fortify the framework. However, caution is warranted, as exclusive reliance on tokens for sensor data security may expose potential vulnerabilities. In summary, Spring Boot emerges as a robust companion in Java-centric microservices development, proficiently accommodating the inclusion of IoT features like sensor oriented REST endpoint creation.

Mihita Tinta created a mobile application by integrating Flutter and Spring Boot where the goal was to help the user using alternatives to access the application, instead of using pins or passwords[11].This combination of Flutter, Spring Boot, and Rest API in this build up underscores Flutter's pivotal role as an open-source UI for cross-platform app development. Meanwhile, Spring Boot adeptly manages the routing of requests from mobile backends to APIs, enhancing overall efficiency. The amalgamation of Flutter with Rest APIs proves particularly advantageous in accessing the application, with Flutter dependencies to share the accessible data. This further allows implementation WebAuthn which is a strong credential method to safeguard the data shared. It's imperative to acknowledge, however, that mastering Flutter may necessitate a gradual learning curve. In summary, the collaborative fusion of Flutter, Spring Boot, and Rest API presents a dynamic framework for streamlined cross platform app development. .

III. METHODOLOGY

In the past few years, there has been a swift expansion in the healthcare industry, becoming a noteworthy source of revenue and employment. Advancements in technology have transformed disease diagnosis and health monitoring, replacing traditional physical analyses with miniaturized devices like smartwatches. The integration of communication services and technologies like IoT has enhanced the accessibility of healthcare facilities, offering advantages such as higher accuracy, lower costs, and improved predictive capabilities. The pervasive acceptance of IoT is driven by enhanced understanding of software and applications, alongside progress in mobile and computer technologies, convenient availability of wireless technology, and the expanding digital economy.

The integration of the Internet of Things (IoT) framework in healthcare represents a fusion of IoT technology and cloud computing within the medical domain. This framework sets forth guidelines for the transmission of patient information from diverse sensors and medical apparatus to a healthcare network. The topology of Healthcare IoT (HIoT) has been described in **figure.1**. A fundamental HIoT system comprises three primary components: the publisher, representing connected sensors and medical devices recording vital patient information; the broker, responsible for processing and storing data in the cloud; and the subscriber, facilitating continuous monitoring and access to patient information through devices like smartphones or computers. The publisher processes data and provides feedback in case of physiological anomalies, contributing to real-time patient monitoring. The HIoT integrates discrete components into a hybrid grid tailored to specific purposes within the healthcare network. While the topology varies based on healthcare demands and applications, the success of an IoT system in healthcare hinges on its alignment with medical rules and diagnostic procedures for different diseases. Designing an IoT-based healthcare system requires careful consideration of associated activities related to the targeted health application.

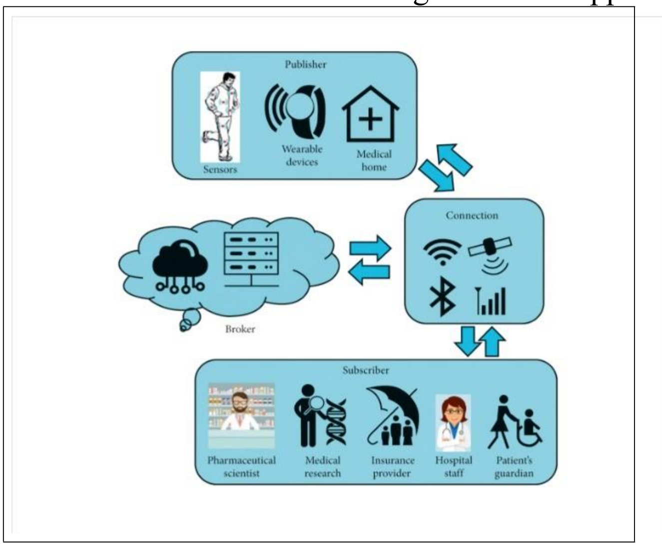

figure.1 **hiot**

A. FLUTTER

1) **Mobile Application Development in IOT***:* Mobile applications play a crucial role in the Internet of Things (IoT) ecosystem by providing user interfaces for IoT platforms. Here are some key points highlighting their importance: 1.Seamless Connectivity: Mobile apps act as a bridge between IoT devices and users, enabling seamless connectivity. They leverage wireless technologies like WiFi, Bluetooth, or cellular networks to establish communication with IoT devices, making it easier for users to access and manage their IoT ecosystem. 2.Data Visualization: Mobile apps facilitate the visualization of data collected from IoT devices. They present data in a visually appealing and understandable format, such as graphs, charts, or dashboards. This allows users to gain insights from the collected data and make informed decisions based on the information provided. 3.Customization and Personalization: Mobile apps allow users to customize and personalize their IoT experience. They often provide options to configure settings, create automation rules, and tailor the app interface to suit individual preferences. This flexibility empowers users to adapt the IoT ecosystem to their specific needs and requirements. 4.Remote Accessibility: Mobile apps enable users to access and control their IoT devices from anywhere, as long as they have an internet connection. This remote accessibility enhances convenience and provides flexibility, allowing users to manage their IoT ecosystem even when they are away from their physical location. 5.Integration and Interoperability: Mobile apps serve as a central hub for integrating and managing multiple IoT devices from different manufacturers. They provide a unified interface that can communicate with diverse devices, protocols, and standards, ensuring interoperability and seamless integration within the IoT ecosystem.

2) **Cross platform application development using Flutter**: The primary objective behind developing this app is to facilitate easy and efficient education for all students by establishing seamless connections and enabling the effortless sharing of study materials. For the construction of this app, we have chosen to utilize Flutter, the latest framework making significant strides in the realm of mobile app development. In this context, we will delve into an exploration of the Flutter framework, understanding its merits, demerits, and the diverse methods employed for testing Flutter applications. Flutter includes both a Software Development Kit (SDK) and a UI library based on widgets. The UI library comprises a variety of reusable UI elements, including sliders, buttons, and text inputs. Developers engaged in the creation of mobile applications with Flutter utilize Dart as the programming language. Dart, akin to JavaScript in syntax, is a typed object programming language designed specifically for frontend development. The widespread adoption of IoT is propelled by an improved comprehension of software and applications, coupled with advancements in mobile and computer methodologies, easy accessibility of wireless capabilities, and the growing digital economy.

3) **Flutter over other technologies**: The achievement of high performance and productivity in Flutter is realized through the application of several techniques. Unlike many other prevalent mobile platforms, Flutter does not incorporate JavaScript; instead, it employs Dart as the programming language. Dart compiles to binary code, ensuring that the Flutter app runs with the native performance efficiency of Swift, Java or Kotlin. Notably, Flutter does not rely on native UI components. While this may seem unconventional initially, the absence of a communication layer between components, as they are implemented within Flutter itself, proves advantageous.

B. **SPRING BOOT**

1) **Analysis of significance of Spring Boot**: The comparative evaluation of Spring Boot against existing mechanism is delineated as follows: Spring Boot, in contrast to other frameworks utilized for constructing RESTful Web services, distinguishes itself through its intrinsic features, effectively mitigating project cluttering. The incorporation of Spring Security within the framework establishes an additional layer of protection that surpasses the security measures typically associated with JSON Web Token (JWT) implementations, such as those employed in Node.js. Leveraging the robust foundation of a Java framework, Spring Boot exhibits a distinct advantage in terms of its multithreading capabilities. This attribute is particularly noteworthy for the development of backends tasked with accommodating voluminous datasets, a crucial requirement in the contemporary landscape. In this regard, Spring Boot emerges as a preeminent choice, setting it apart from alternative frameworks.

The technology distinctions utilized in Spring Boot and their consequences are outlined below: Spring Boot relies on the Java language within the Spring framework, distinguishing itself from alternative frameworks such as NodeJS, Django, and PHP. However, for lightweight and highly scalable solutions, NodeJS may offer a more suitable choice. Leveraging Java, a multithreaded language, provides Spring Boot with a distinct advantage in this context compared to NodeJS.

Highlights of Spring Boot are:

i. **AutoConfiguration** - Automatic configuration of certain classes is facilitated if their presence is detected in the classpath. For instance, Spring Boot will generate an in memory database and a readily deployable JDBC Template by recognizing the inclusion of JDBC Template and H2.jar in your classpath. Writing the aforementioned code is unnecessary when implementing the JDBC Template in a DAO layer.

ii. **Starter** POMs - The starter POMs alleviate the burden by detecting and including standard dependencies in the project. To construct a basic Spring MVC-powered REST application and execute it in a built-in container, a mere seven dependencies are required. All of these dependencies can be integrated with the spring-boot-starter-web dependency into your pom.xml through its bundles.

iii. **Spring Boot CLI** - To construct web applications based on Spring using the Groovy programming language, leverage the CLI—an interface accessible through command-line operations within the framework. The compatibility between Spring Boot and Groovy is evident, as Groovy’s straightforwardness is advantageous for Java developers, while Spring developers find ease in the simplicity offered by Spring Boot.

iv. **Actuator** - Employing the actuator affords the ability to observe the real-time status of a functioning Spring Boot application. This application furnishes an array of measurements and detailed information regarding the operational aspects of applications in a production environment. Leveraging an actuator allows for the identification of specific beans and configurations utilized within an application context, decision-making based on automatic configuration, scrutiny of existing environmental variables, system characteristics, and command-line inputs., among other functionalities.

2) **Springboot In IOT**: Exploration has been given to leveraging Java technology in crafting our service solutions. In the course of service development, the adoption of the Spring Boot framework has been embraced. Originating from the Pivotal Team, this Java-based, opensource framework streamlines the creation of microservices. Specifically designed for spring-based applications, Spring Boot yields multiple benefits in IoT service development. The utilization of Spring Boot in IoT services presents numerous advantages. Notably, it simplifies the development of Spring applications by autonomously managing intricate XML configurations through autoconfiguration. With inherent support for various libraries, Spring Boot substantially diminishes development time, enabling independently executable applications. Its opensource nature, coupled with minimal configuration requirements, enhances its effectiveness. The annotation based coding paradigm provided by Spring Boot simplifies interactions with databases. Implementing functionalities essential for IoT devices, such as creating REST endpoints for sensors to POST values and facilitating endpoints for data retrieval, becomes straightforward. Furthermore, the Spring Boot framework integrates robust security mechanisms, encompassing tokenization and authorization through Spring Security, ensuring the security of sensor data.

3) **Integration of SpringBoot and Flutter**: The task at hand involves delving into the intricacies of merging Flutter and Spring Boot to create a seamless end-to-end solution, highlighting the challenges encountered in this integration journey. The exploration begins by comprehensively understanding the complexities that arise when marrying Flutter, an open-source UI development kit, with Spring Boot, a Java-based framework renowned for building robust RESTful APIs. The inherent goal is to establish an integrated solution that seamlessly connects the front-end capabilities of Flutter with the powerful back-end functionality provided by Spring Boot.

One notable challenge lies in orchestrating effective communication between the Flutter mobile application and the Spring Boot Rest API. This involves navigating the nuances of making HTTP calls, handling data exchange, and ensuring a harmonious interaction between the distinct components of the solution.

WebAuthn, a state-of-the-art technology for user authentication, adds a layer of complexity and innovation to the integration. Challenges emerge in the implementation of WebAuthn, particularly in configuring and synchronizing the Fido Google Play services library with the Flutter application. The secure generation and storage of strong credentials, encompassing private and public keys, become critical components in fortifying the authentication process.

There also needs to be a thorough understanding of the intricacies of asynchronous programming in Dart, the language used in Flutter development. Managing asynchronous calls, especially in the context of waiting for server responses and handling potential errors, becomes a challenge that developers need to address effectively. The need for synchronization and smooth execution in an asynchronous environment is a key consideration.

Debugging and troubleshooting, intrinsic to any integration effort, emerge as ongoing challenges. The article acknowledges the importance of providing insights and strategies for developers to identify and resolve issues on both the Flutter and Spring Boot sides. This includes debugging HTTP calls, understanding error messages, and ensuring the secure transmission of data.

IV. PROBLEMS IDENTIFIED

In recent years, the health industry has undergone significant technological advancements, notably in intelligent sensors, cloud computing, and communication technologies, leading to better healthcare. The rise of the Internet of Things (IoT) has particularly revolutionized user accessibility. However, alongside these strides, notable challenges and future research areas have surfaced. These include managing escalating servicing costs, addressing power consumption, standardizing healthcare products, ensuring data privacy, optimizing patient identification, integrating self-configuration in IoT devices and enabling continuous monitoring in long-term healthcare. Effectively tackling these challenges is crucial for the sustainable evolution of IoT in healthcare.

V. PROPOSED SYSTEM

The innovative system presents an effective solution to the challenges inherent in traditional educational assessment methods, aiming to redefine the landscape of knowledge evaluation in response to the dynamic needs of modern education. The core features of our system include the integration of a diverse array of wearable sensors—such as tilt, blood pressure, breathing, electrocardiogram, electromyogram, and electroencephalogram—to provide comprehensive health monitoring. Leveraging Flutter's capabilities, the platform establishes a wireless sensor network, enabling real-time data transmission and collaboration among integrated sensors.

Utilizing inexpensive, low-power, and versatile sensor nodes, the system supports various monitoring requirements while ensuring the platform's effectiveness and adaptability in the wearable healthcare IoT domain. With a user-friendly interface rooted in the Flutter framework, the platform offers easy navigation and interaction for both end users and healthcare practitioners. Real-time data synchronization mechanisms enable seamless integration and transmission of health data between wearable devices and the Flutter-based application, ensuring access to accurate data for tracking and analysis.

Robust data security and privacy measures, including encryption protocols to safeguard sensitive health information. PostgreSQL facilitates streamlined real-time data management, enhancing overall accuracy and reliability. The Flutter framework contributes to a unified user experience by offering consistent functionalities and design elements across platforms, optimizing performance for responsive and lag-free experiences in real-time health monitoring applications.

Figure 2 illustrates the dashboard layout of the proposed system which comprises of distinct sections for user interaction and information display. The Filters section empowers users to refine data through criteria like date range, patient demographics, and condition. Dashboard provides key metrics such as patient satisfaction scores and cost per procedure. Phanca offers detailed insights into specific healthcare metrics, while Curse (2003-34) depicts trends in healthcare data evolution. The dashboard features user-friendly elements like Add/Remove Code for metric customization, Create Compound for combining metrics, and Add/Remove Cards for card management. Current Value showcases the real-time metric value, and Live Data indicates whether the displayed data is dynamic or static.

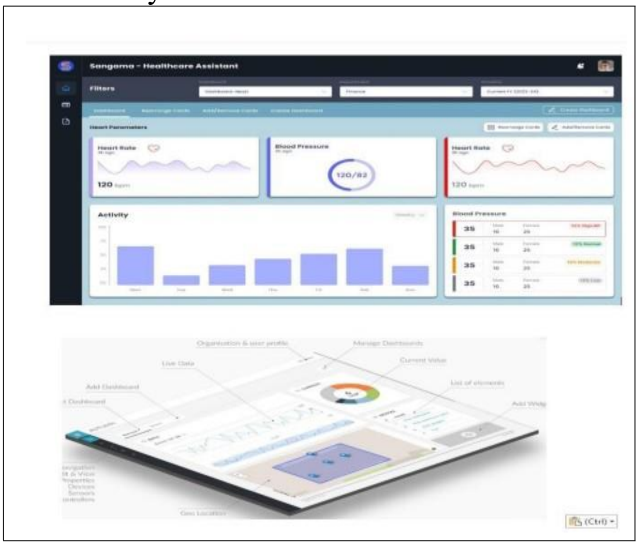

figure.2 **dashboard**

In summary, the proposed system, leveraging Flutter for the wearable healthcare IoT platform, prioritizes user friendly interfaces, real-time data synchronization, seamless integration, and robust security measures to create an effective, efficient, and accessible health monitoring solution across multiple devices.

VI. CONCLUSION

The comprehensive exploration of the Healthcare Internet of Things (HIoT) system within this review has unfolded a nuanced understanding of its architecture, components, and the intricate communication network binding these elements. Concurrently, the narrative extends to illuminate the current landscape of healthcare services, revealing the transformative impact of IoT-based technologies. Through the adept application of these technological paradigms, healthcare professionals have been empowered to conduct remote monitoring and diagnosis of various health conditions, measure diverse health parameters, and extend diagnostic capabilities to previously inaccessible locations. This paradigm shift signifies a pivotal transition from a hospital-centric approach to a more patient-centric healthcare system, marking a profound evolution in the industry.

Furthermore, the review not only expounds upon the manifold applications of the HIoT system and recent trends but also meticulously outlines the formidable challenges entwined with its design, manufacturing, and operational phases. These challenges, intricately discussed, are poised to serve as a crucible for catalyzing future advancements and directing focused research endeavours in the years to come. In tandem, the review offers a comprehensive and contemporary repository of knowledge on HIoT devices, serving as a valuable resource for readers poised not just to embark on their research journeys but also to actively contribute to the ongoing advancements within this dynamic field.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to Mr. Arun Kumar Kumathalli of BayMap Systems for providing us with the invaluable opportunity to contribute to this research project. His unwavering support, technical expertise, and insightful guidance have been instrumental in shaping the trajectory of this endeavour. His commitment to fostering a collaborative and conducive research environment has been truly commendable. We are grateful for his contributions and the positive impact he has had on our academic and professional journey.

REFERENCES

[1] Bikash Pradhan, Saugat Bhattacharyya and Kunal Pal, IoT-Based Applications in Healthcare Devices, 19 Mar 2021, Volume 2021, 6632599,https://doi.org/10.1155/2021/6632599

[2] Leonardo Babun, Kyle Denney, Z. Berkay Celik, Patrick McDaniel and A. Selcuk Uluagac,A survey on IoT platforms: Communication, security, and privacy perspectives, 19 June 2021, Computer Networks Volume 192, 108040,https://doi.org/10.1016/j.comnet.2021.108040

[3] Mahdi Fahmideh and Didar Zowghi, An exploration of IoT platform development, January 2020, Information Systems Volume 87, 101409,https://doi.org/10.1016/j.is.2019.06.005

[4] International Research Journal of Modernization in Engineering Technology and Science (Peer-Reviewed, Open Access, Fully Refereed International Journal) Volume:04/Issue:04/April-2022 Impact Factor- 6.752

[5] Kuzmin, N., Ignatiev, K., Grafov, D. (2019). "Experience of developing a mobile application using Flutter. In Lecture notes in electrical engineering" (pp. 571–575). https://doi.org/10.1007/978-981-15-1465-4$_5$6

[6] Olsson, M. (2020). "A Comparison of Performance and Looks Between Flutter and Native Applications: When to prefer Flutter over native in mobile application development". DIVA. https://www.divaportal.org/smash/record.jsf?dswid=-2838pid=diva2

[7] S. Boukhary and E. Colmenares, "A Clean Approach to Flutter Development through the Flutter Clean Architecture Package," 2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2019, pp. 1115-1120, doi: 10.1109/CSCI49370.2019.00211.

[8] M. Mythily, A. Samson Arun Raj and I. Thanakumar Joseph, "An Analysis of the Significance of Spring Boot in The Market," 2022 International Conference on Inventive Computation Technologies (ICICT), Nepal, 2022, pp. 12771281, doi: 10.1109/ICICT54344.2022.9850910.

[9] S. R. Modugu and H. Farhat, "Implementation of the Internet of Things Application Based on Spring Boot Microservices and REST Architecture," Advances in Intelligent Systems and Computing, pp. 20–31, 2020, doi: 10.1007/978-3-030-63322-6$_3$

[10] Jashandeep Singh, Swapnil Srivastva, Dipanshu Raj, Shubhampreet Singh , Mir Junaid Rasool, "FLUTTER AND FIREBASE MAKING CROSS-PLATFORM APPLICATION DEVELOPMENT HASSLE-FREE", International Research Journal of Modernization in Engineering Technology and Science (Peer-Reviewed, Open Access, Fully Refereed International Journal) Volume:04/Issue:04/April2022, Impact Factor- 6.752.

[11] Spring Boot and Flutter integration without passwords Medium · Mihaita Tinta, Oct 17 2022