A Design of Patient Registration Apps using Flutter, Laravel and, Vue JS

Ketut Agus Seputra[1], Luh Joni Erawati Dewi[2]

{agus.seputra@undiksha.ac.id[1], joni.erawati@undiksha.ac.id[2]}

Universitas Pendidikan Ganesha, Singaraja, Bali[1,2]

Abstract. The creation of mobile applications needs a more systematic and complicated technical methodology. Currently, Android and iOS are the two most popular smartphone platforms. As a result, presenting programs that can be utilized on both platforms is one of the priorities for developers. Furthermore, developers must keep in mind that not all target customers own and use smartphones. Consequently, the adoption of mobile applications to the market needs a website as a backup service and operational support for the mobile application. The development was carried out utilizing a full-stack development method with laravel, vue.js, and flutter, depending on the demands and considerations of the team's efficiency and cost. Combining these three frameworks has shown to be effective in developing SPA Websites and Mobile Apps for patient registration.

Keywords: Mobile Apps, SPA Web, Flutter, Laravel, VueJS.

1 Introduction

Mobile device technology is evolving in terms of the number of users, the number of applications, and the performance of those applications. Mobile app development is an essential aspect of every company's business, social, or health strategy to stay competitive. Observing and following current trends is essential, especially in an effort to increase business value. Chatbots and voice assistants, Augmented Reality, the Internet of Things, Cellular Wallets, Biometric Authentication, and Telemedicine are some of the mobile app development trends. Smartphones have the advantages of computing, ease of use, and being connected to user behaviors. Mobile apps will continue to dominate digital interactions in the future. The mobile application does more than just display web content. It also merges hardware and sensors into one seamless unit. Sensors in mobile apps can provide a new approach to systems that collect data about users' habits[1]. Quality mobile application development necessitates a systematic engineering strategy, cost estimation planning, human resources, quality control, and product positioning, among other things[2]. Experts argue that various factors must be addressed when developing a mobile application, including the app's features, user-friendliness, user interface design, application functionality, and app size[2].

IConVET 2021, November 27, Singaraja, Indonesia

DOI 10.4108/eai.27-11-2021.2315532

A website is always required for mobile-based information systems as a channel of information on the services supplied by the application to potential users. Websites are also used to increase Search Engine Optimization for services or features of an application on search engines. Furthermore, mobile app developers should be mindful that not everyone has and uses a smartphone. As a result, application developers must also provide alternative services accessible via the website. Another factor to consider is that mobile applications target specific people based on the target market. Therefore, just the functionality necessary by the end-user is included in the mobile application. It also has to do with reducing the size of applications and making optimum use of resources on smartphones. More extensive features that require numerous inputs, business logic, and document management, such as administrators, should be supplied solely on the website, following the concept of designing client-server mobile applications based on the fat and thin client paradigm[3]. Thus, a mobile development framework, backend, website, and other supporting information technology infrastructure are required for constructing a mobile-based information system. Within a small team, development can be carried out utilizing a full-stack development method employing laravel, vue.js, and flutter, depending on the demands and considerations of the team's efficiency and cost. This study proposed a new approach to create a mobile patient registration system and website based on the Patient Identifier Cross-Reference Manager as a starting point for constructing data sharing-based medical records. Sharing medical records across healthcare providers is critical, especially in the face of the Covid-19. Many research on the development of medical record information systems in Indonesia has been conducted, ranging from the medical record system in a health service [4][5] to communication between health care systems [6][7] and how patient data can be incorporated into a data center [8].

2 Related Works

Many mobile application development studies have been conducted to test the performance and effectiveness of using mobile applications. However, the development of mobile applications requires a lot of planning. First, it must be determined who the target users are. The target user is very influential, especially in determining the user's platform or operating system. Currently, two operating systems are commonly used, namely Android and iOS. Application development must be adaptive to both operating systems. Development with a cross-platform approach can be a solution to build syntax on Android and iOS[9]. Cross-platform, commonly known as WORA (Write, Once, Run, Anywhere), allows a code to run on Android and iOS. Several classifications of cross-platform approaches can be used, namely Web Apps, Hybrid Apps, Interpreted Apps, and Widget-Based Apps [10]. Many companies have used cross-platform schemes to develop applications on two platforms [11]. Two frameworks are widely used today, namely, Flutter widget-based and React Native Interpreted Apps. React Native and Flutter have proven to be cross-platform frameworks that have the efficiency and ease of developing high-performance mobile applications [12][13]. This certainly supports a product to be distributed more quickly to the market. Nevertheless, it must be admitted that flutter has a brighter future. Developed by Google with support for standardization and high security [2][9][10], flutter can revolutionize React Native design and provide consistency in terms of syntax neatness and SDK. Widget-based Flutter rendered on a dedicated engine improves app performance [12][14]. Like the native framework, flutter can also use the hardware embedded in the smartphone.

Basically, mobile application development is not significantly different from desktop application development. Modern mobile application development is developed for specific purposes and targets, including cross-platform development and data virtualization, energy efficiency and real-time computing, mobile back end, progressive web apps, cloud-platform access and control apps, and new 5G mobile apps[3]. Currently, application development has used distributed systems techniques with a client-server approach on several heterogeneous computer systems. Therefore, mobile applications usually require the service of one to several servers which are commonly referred to as backends. The backend has an important role to run business process functionality. Several approaches can be chosen to develop the backend, namely Package LAMP(Linux, Apache, MySQL, PHP), XAMPP(Linux, Apache, MariaDB, SQLLite, PHP, Perl), server-side Node.js, and MEAN(MongoDB, Express.js, Angular.js, Node.js)[3]. To facilitate communication, services are usually separated into several microservices. Nowadays, microservices architecture has become the industry standard, and RESTFul is a commonly used model for microservices due to its lightweight, scalability, and compatibility with HTTP protocols [15][16].

The Laravel framework provides standard requirements for building Backend and RESTFul infrastructures. Laravel is developed with the PHP programming language providing simplicity and design flexibility[17]. In addition, Laravel is developed with a modern architecture that integrates with PHP 7 features to make the performance even higher. It is supported by the Object-relational mapping (ORM) feature that makes data manipulation lighter and easier. Routing management and integrated unit testing are also features that simplify API development [18]. The Laravel framework can be used to create a backend and API that can be utilized by mobile apps and other third-party applications.

There has been a need for sophisticated and complex web applications to replace the functionality of desktop applications in recent years. As a result, when developing a website the amount of data that will be managed and presented on the page must be considered. Website is very important to maintain uniformity of information spread across several pages. This is also related to user experiences, or how users can access the web quickly and easily. Multi-Page Applications (MPA) and Single Page Applications (SPA) are two ways for developing high-complexity web applications (SPA). MPA is designed for application development with larger systems and many sorts of services, allowing users to interact in a variety of ways. SPA is a new approach to application development that focuses on simpler and lighter interactions [19]. As a javascript framework, Vue is ideal for transforming a website front-end into a high-performance, sophisticated, data-rich application that is simple to design [20]. Furthermore, some laravel packages, such as laravel jetstream and Vue UI, currently allow integration with Vue, making full-stack creation of a SPA-based patient registration system more straightforward [21][22].

3 Method

Integrating the Healthcare Enterprise (IHE) was created to help current SIK communicate and share information more effectively. This is a great opportunity for patients to be recorded in several health systems, with diverse medical histories distributed over multiple clinical domains. The ADDIE (Analysis, Design, Development, Implementation, Evaluation) development

method created the system. This method is very effective for creating a system prototype[9]. The ADDIE approach and tools depicted in **Figure 1** are in a simplified form.

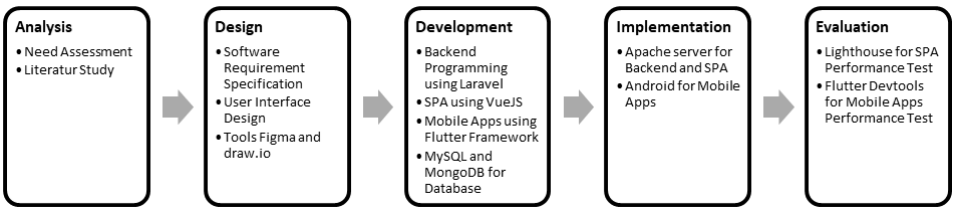

Fig. 1. ADDIE method and software tools.

4 Results and discussion

4.1 Analysis

The literature review in this study used the Patient Identifier Cross-Reference Manager and other documents relevant to Patient Care Coordination as one of the domains in the IHE for the analysis phase. The Patient Identifier Cross-referencing Integration Profile (PIX) governs the exchange of information between many clinical domains within a health service or between several health services[23]. Patient identity information was transmitted from an identity source to the Patient Identifier Cross-reference Manager. The ability to query/respond for a list of cross-referenced patient identifiers or receive an update notification for a list of cross-referenced patient identifiers. As a result, the Patient Identifier Cross-reference Manager was in charge of establishing, maintaining, and disseminating lists of identifiers that are aliases of one another across many Patient Identifier Domains. The actors directly participating in the Patient Identifier Cross-referencing Integration Profile and the transactions that happen between them are depicted in **Figure 2**.

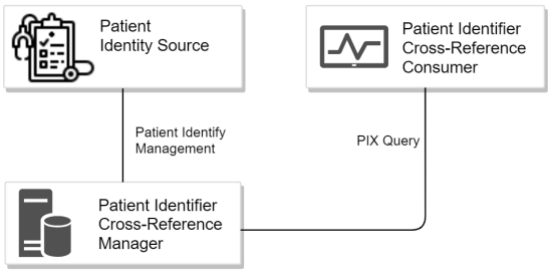

Fig. 2. PIX transaction.

A patient identity source system is a single system that assigns a unique identifier to each instance of a patient-related object and keep track of identity attributes. Although a single Patient Identity Source may issue several identifiers to the same patient and convey this knowledge to the Patient Identifier Cross-reference Manager, a patient identifier is uniquely connected with a particular patient. The NIK was used as a unique identifier to identify the patient in this case. The problem addressed in this study was the creation of a patient registration system as a starting point for identifying patient profiles in PIX. This system was in charge of mapping patient data sources and supplying patient condition data to other systems in the future. According to the literature, a system that can be accessed by patients is required to save the patient's personal data under the following conditions:

1. Patients must be able to register themselves, and healthcare workers must be able to register them.
2. Through the API service method, the system must be able to communicate data with other systems.
3. The system must be able to offer information in a timely, relevant, and accurate manner.

4.2 Design

All user needs to be obtained from the literature were transformed into system requirements. Entities and characteristics relating to patient data were the major documents that must be filled at this stage. At this point, the software requirement specification was developed.

System architecture

The system was developed in three phases: API services with Laravel, Vue JS for SPAs, and Flutter for mobile apps as the last step. Laravel was utilized for client-server application development, with business logic data and the Graphic User Interface separated by writing syntax. Laravel can become a stable server and provide API services that can be utilized by Vue.js and flutter as a client with RESTful features and several other essential components including routing, middleware, MVC (Model View Controller), ORM, and Authentication. When a client sent a RESTful through HTTP Request, the server accepted and forwarded it to the PHP Engine. The initialization actions were performed before the request was delivered to the middleware. Filtering requests, including client authentication, was the responsibility of middleware. The sanctum role in Laravel was used as middleware to limit access to certain routes based on the user's permissions. Through the JSON Web Token (JWT), HTTP requests could be tokenized to identify the user to properly grant access rights. After passing the routes, the MVC controller took over the request. The controller was in charge of handling the execution of business operations and sending requests and data to the Model. Through the ORM component, the model managed data in the database, which in this example were MySQL and MongoDB. The data were not displayed by the view blade in the RESTful SPA schema, hence the view rendering step was skipped. The collected data were encoded in JSON and delivered to the client as an HTTP Response. So that data flow more efficiently and effectively in the SPA.

The SPA is a web application that does not require page reloading when utilized in a browser. When users interact with the application, they will not switch pages by sending requests to the

server. The main advantages of adopting SPA are lower network bandwidth consumption and faster navigation processes[19]. Network bandwidth was reduced because this SPA was widely supported by JavaScript libraries. The navigation process was quicker since the data received from the server is in JSON (JavaScript Object Notation), which could be presented asynchronously using JavaScript. Vue is a framework for creating progressive user interfaces. The core library focuses solely on the view layer, and it's simple to use and combine with other libraries or projects. As a result, when combined with contemporary technology and accompanying libraries, Vue is perfectly capable of powering sophisticated SPA[24]. **Figure 3** depicts the SPA architecture that was constructed.

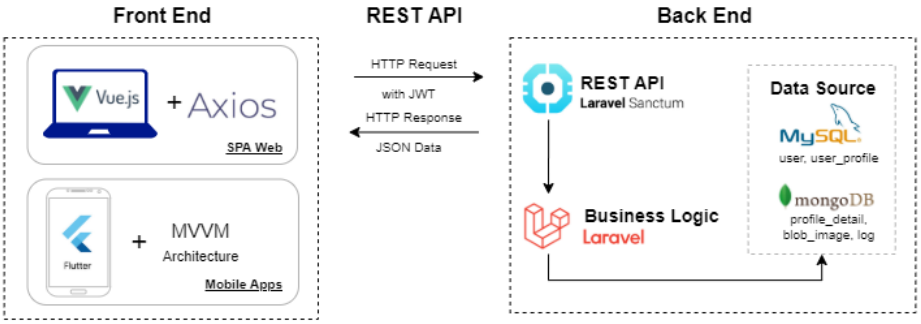

Fig. 3. SPA and mobile apps data work flow.

Use case

Some of the features that must be provided on the system according to the analysis of user needs are as follows:

1. User registration and login are required.
2. The personal info of users should be updated.
3. Upload personal photographs and identification cards.

Figure 4 is a use case diagram that can give a quick overview of the system's functionality and user access permissions.

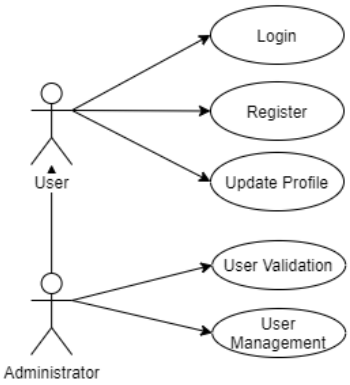

Fig. 4. Use case diagram.

4.2.1 Activity diagram

Users could register on their own through the website or mobile app register menu and complete personal info, photographs, and id cards after successfully registering. The admin was then verified the user data. Other system features were used with user data that had been declared valid. Users who have been labeled invalid could correct their data using the update profile menu. **Figure 5** shows more details about the registration process, starting with user login and ending with validation by the health facility administrator.

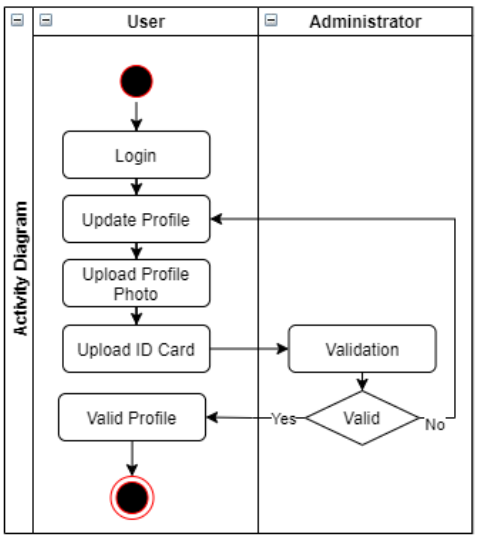

Fig. 5. Activity diagram user registration.

Entity relationship diagram

User data were the major information to identify patients when data were exchanged between domains and systems. Therefore, patient entities and attributes were designed by international standards that correspond to Health Level Seven (HL7). Entity Relationship Diagram (ERD) explained data flow based on system-related objects. **Figure 6** is a Conceptual Data Model user registration system that provides an overview of the logical structure of all data applications, regardless of software.

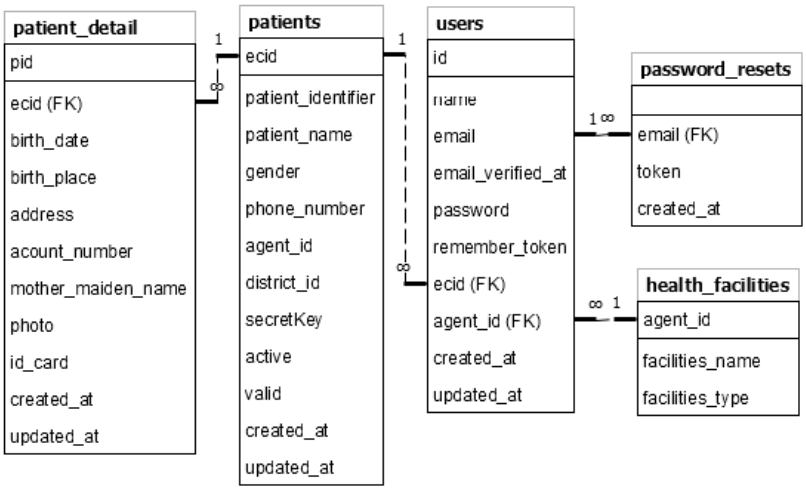

Fig 6. Entity relationship diagram.

4.3 Development

The step of beginning work on an application project according to a predetermined design is known as development. At this point, all tools and related technology were applied. Backend programming and front-end programming were the two phases that made up the development process:

Backend programming

To store user profiles, this phase started with database construction, which used two Database Management Systems (DBMS), notably MySQL and MongoDB for profile detail. The next step was to build the website utilizing the Laravel framework as the registration system's principal business logic. The creation of middleware for communication between client apps and databases was the final stage of backend development. Middleware was built with the Laravel API library and the Auth Sanctum API authentication system.

Front-end programming

This phase began with creating the client-side User Interface (UI). The website was the first to be developed, and it eventually became the major system for both administrators and users. The

SPA-based website was developed using Vue.js and the Tailwind framework as the basic Cascading Style Sheets (CSS). Using the REST API that Laravel had designed to communicate with the database. In the construction of front-end websites, several Vue packages were used, including:

1. The Vue instance was used to set up the vue package for use.
2. Vue Component is a Vue instance used to render the Vue file's look based on the router's commands.
3. The Vue router was used in SPA to render a component within the page dynamically.
4. Vue Axios used HTTP clients for making API queries.
5. Vue-sweetalert2 is a javascript library for displaying a process's popup response message.

The Model–View–ViewModel (MVVM) architecture was used to create the mobile app utilizing the flutter framework. MVVM was used for a variety of reasons. First, developers could collaborate based on their particular areas of expertise. Second, in certain related methods, avoid duplicating code. Finally, testing between the application logic and the user interface was simple[25]. **Figure 7** depicts the MVVM architecture in this case.

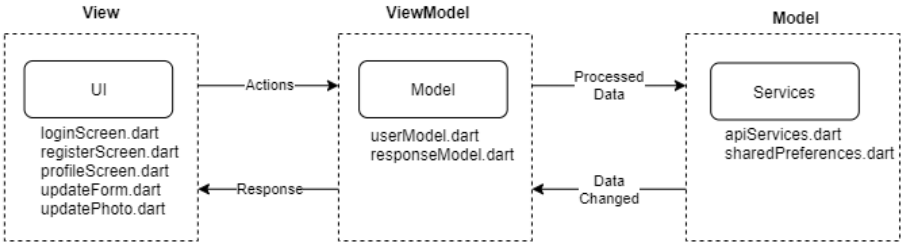

Fig. 7. MVVM architecture.

The model was in charge of data accessibility. The two sorts of data sources were user data collected through the Laravel API and local data in the form of names, emails, photographs, and tokens temporarily kept in the shared preferences package. The data were exchanged via HTTP package with the Laravel API in JSON format. The token obtained after login was held in memory and then inserted in the HTTP request header to enable secure data exchange. This tokenization happened in vue, which follows the same API methodology.

4.4 Implementation

The developed registration system was made up of two platforms: a website and a mobile application. SPA website and Backend installed on Apache web server with minimum PHP version 7.2. Admins use the website created with the SPA scheme utilizing the Laravel framework and Vue JS on the front-end to handle user data, including user validation, as shown in **Figure 8**.

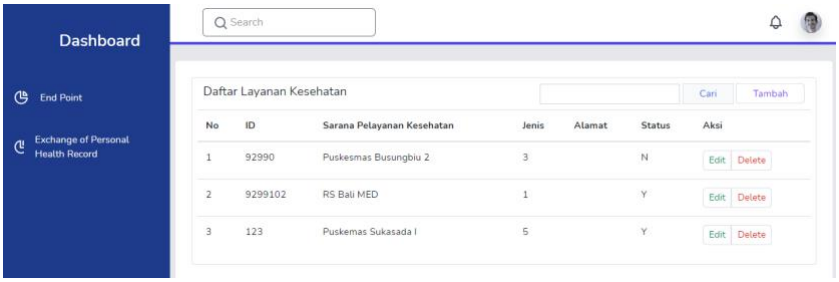

Fig. 8. SPA page.

The mobile application could be used on smartphones with a minimum specification of android 8.0 with 1GB of RAM. **Figure 9** shows the mobile application allowing users to register and update their profiles. In this scenario, the advantage of using a mobile application is that it was simple to register, log in, and update profiles. Notifications could be obtained in real-time in the future.

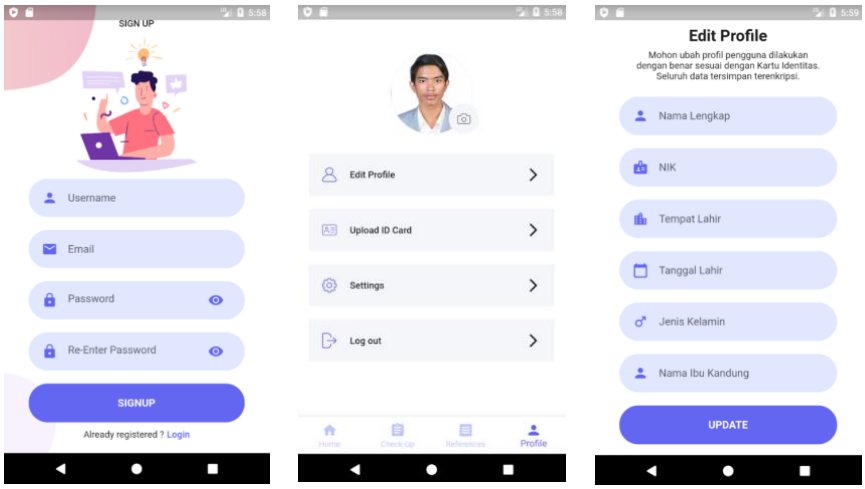

Fig. 9. Mobile apps.

4.4 Evaluation

The concept of Async-Components combined with vue-router had overcome one of the challenges associated with loading a huge page on a traditional website. The minified vue syntax also reduced the size of app.js, allowing Async-Components to load a 5.4 MB page in just 1.7 seconds. The transfer speed and API request size have also been optimized, the user profile could be loaded in only 57.2ms for a single request of 3.1kB. Table 1 shows the results of testing on the SPA website using the lighthouse tools on google chrome. All aspects of the website received a good evaluation based on the indicators for performance, accessibility, best practices, and SEO[26].

Table 1. SPA testing.

Menu	Performance	Accessibility	Best Practices	SEO	Transferred	Resources
User Profile	94	93	93	91	4.2kB	762kB
Profile Edit	92	93	93	91	5kB	762kB
Patients	91	85	100	91	185kB	844kB
Facilities	94	85	100	91	184kB	841kB

Flutter DevTools was used to test mobile applications by measuring how long it took the app to render 60 frames in milliseconds. UI threads and Raster threads were used for performance testing. The time Dart Virtual Machine needs to render a 60 FPS UI was measured in UI. Raster is a measurement of the time. It took the graphics processing unit to generate a 60 FPS UI (GPU). The test results can be seen in Table 2.

Table 2. Mobile apps testing.

Feature	AVG UI (ms)	AVG Raster (ms)	AVG FPS (fps)
Register	42.9	11.2	34
Login	17	7.6	20
Update Profile	17.0	6.1	55
Update ID Card	14.3	3.7	56

Each frame must render in fewer than 16 milliseconds to achieve excellent 60fps performance [27]. When it took more than 16ms, the frame was categorized as jank. The UI threads for the register, login, and profile update functions were all in the junk category, as shown in table 2. As a result, the average rendering frame in each feature did not exceed 60 frames per second. Therefore, some improvements in the flutter code were needed, such as reducing the image file size, animation, and tree widgets.

5 Conclusion

Researchers came to several conclusions based on application development research. First, the REST API has been created using Laravel as a back end works well in Vue JS and Flutter. This provides effectiveness because developers only need one API for system logic and data management. Second, the testing by using lighthouse software showed that the performance of the web SPA designed with Vue JS was a good rating. This also demonstrates that the web could respond more quickly with SPA, which improved the user experience. Third, jank UI threads were identified in the dev tools test that made the mobile application performance assessment was not optimal. As a result, various enhancements to the utilization of tree widgets and the reduction of images in each flutter stage were required to create mobile applications. However, mobile apps black-box testing based on various test scenarios shows that all of the capabilities in the registration system performed properly in general. It indicates that the Laravel, VueJS, and Flutter frameworks could be combined to develop applications more effectively and efficiently in small teams.

References

[1] J. Zhang, C. Calabrese, J. Ding, M. Liu, and B. Zhang, “Advantages and challenges in using mobile apps for field experiments: A systematic review and a case study,” *Mob. Media Commun.*, vol. 6, no. 2, pp. 179–196, 2018.

[2] G. Catolino, P. Salza, C. Gravino, and F. Ferrucci, “A Set of Metrics for the Effort Estimation of Mobile Apps,” *Proc. - 2017 IEEE/ACM 4th Int. Conf. Mob. Softw. Eng. Syst. MOBILESoft 2017*, pp. 194–198, 2017.

[3] A. Luntovskyy, “Advanced software-technological approaches for mobile apps development,” *14th Int. Conf. Adv. Trends Radioelectron. Telecommun. Comput. Eng. TCSET 2018 - Proc.*, vol. 2018-April, pp. 113–118, 2018.

[4] C. Pusparani, B. Priyambadha, and A. Arwan, “Pembangunan Sistem Aplikasi Rekam Medis Elektronik Dan Pendaftaran Pasien Online Berbasis Web Studi Kasus Klinik Medis Elisa Malang,” *Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 2, pp. 1458–1463, 2019.

[5] D. B. Santoso, N. Nuryati, and A. E. Pramono, “Pengembangan Rekam Medis Elektronik Berbasis Software as a Service (SaaS) bagi Dokter Praktik Mandiri,” *J. Kesehat. Vokasional*, vol. 5, no. 3, p. 168, 2020.

[6] A. Winahyu, “Pengembangan Interoperabilitas Informasi Stok Darah berbasis Web Service di Palang Merah Indonesia Daerah Istimewa Yogyakarta dengan Metode Scrum,” vol. 2, no. 1, 2016.

[7] S. M. Marier, “Potensi Interoperabilitas Sistem Informasi Rumah Sakit Untuk Penerapan Standar Pertukaran Data HL7,” *J. Sist. Inf.*, vol. 5341, no. October, pp. 2579–5341, 2018.

[8] B. L. Covid, “SISTEM INFORMASI TERINTEGRASI : SATU DATA COVID-19 INDONESIA Narasi Utama : Integrasi Data.”

[9] K. A. Seputra and G. Sandiasa, “Rancang Bangun Sistem Informasi Satgas Gotong Royong (Si Garong) Desa Adat Berbasis Mobile,” *J. Nas. Pendidik. Tek. Inform.*, vol. 9, no. 3, p. 338, 2020.

[10] K. Shah, H. Sinha, and P. Mishra, “Analysis of Cross-Platform Mobile App Development Tools,” *2019 IEEE 5th Int. Conf. Converg. Technol. I2CT 2019*, pp. 1–7, 2019.

[11] P. D. I. Torino, “Master ’ s Degree Thesis Hybrid Mobile Apps : Development and Testing Challenges,” 2019.

[12] W. Wu, “React Native vs Flutter, cross-platform mobile application frameworks,” *Metrop. Univ.*, no. March, p. 28, 2018.

[13] J. Christoph, D. Rösch, T. Schuster, and L. Waidelich, “Current Progress in Cross-Platform Application Development Evaluation of Frameworks for Mobile Application Development,” *Int. J. Adv. Softw.*, vol. 12, no. August 2021, 2019.

[14] S. Simon and A. Hampus, “Crost-Platform Framework Comparison Flutter & React Native,” Blekinge Institute of Technology, Sweden, 2020.

[15] K. A. Seputra and K. Y. E. Aryanto, “Designing an interoperable social assistance health insurance validation system,” *J. Phys. Conf. Ser.*, vol. 1810, no. 1, 2021.

[16] X. Chen, Z. Ji, Y. Fan, and Y. Zhan, “Restful API Architecture Based on Laravel Framework,” *J. Phys. Conf. Ser.*, vol. 910, no. 1, 2017.

[17] M. Anif, A. Dentha, and H. W. S. Sindung, “Designing internship monitoring system web based with laravel framework,” *2017 IEEE Int. Conf. Commun. Networks Satell. COMNETSAT 2017 - Proc.*, vol. 2018-Janua, pp. 112–117, 2017.

[18] R. Bahana, R. Adinugroho, F. L. Gaol, A. Trisetyarso, B. S. Abbas, and W. Suparta, “Web crawler and back-end for news aggregator system (Noox project),” *2017 IEEE Int. Conf. Cybern. Comput. Intell. Cybern. 2017 - Proc.*, vol. 2017-Novem, pp. 56–61, 2018.

[19] M. Kaluža and B. Vukelić, “Comparison of front-end frameworks for web applications development,” *Zb. Veleučilišta u Rijeci*, vol. 6, no. 1, pp. 261–282, 2018.
[20] V. Javascript, “High Performance Single Page Application with Vue . js,” pp. 1–7.
[21] M. Laaziri, K. Benmoussa, S. Khoulji, K. Mohamed Larbi, and A. El Yamami, “A comparative study of laravel and symfony PHP frameworks,” *Int. J. Electr. Comput. Eng.*, vol. 9, no. 1, p. 704, 2019.
[22] A. Sunardi and Suharjito, “MVC architecture: A comparative study between laravel framework and slim framework in freelancer project monitoring system web based,” *Procedia Comput. Sci.*, vol. 157, pp. 134–141, 2019.
[23] I. PCC Technical Committee, “IHE Patient Care Coordination (PCC) 5 Technical Framework Volume 1 IHE PCC TF-1 Integration Profiles Revision 11.0-Final Text,” vol. 1, 2016.
[24] VueJS, “Introduction,” 2021. [Online]. Available: https://vuejs.org/v2/guide. [Accessed: 05-Sep-2021].
[25] G. Imbugwa, M. Mazara, and S. Distefano, “Developing a Mobile Application Using Open Source Parking Management System on Etherium Smart Contracts.,” *J. Phys. Conf. Ser.*, vol. 1694, no. 1, 2020.
[26] Google Developers, “Lighthouse performance scoring.” [Online]. Available: https://web.dev/performance-scoring/. [Accessed: 01-Sep-2021].
[27] flutter.dev, “Using the Performance view.” [Online]. Available: https://flutter.dev/docs/development/tools/devtools/performance. [Accessed: 01-Sep-2021].