Jurnal Teknik Informatika (JUTIF)
Vol. 5, No. 4, October 2024, pp. 1365-1374
DOI: https://doi.org/10.52436/1.jutif.2024.5.5.2569
p-ISSN: 2723-3863
e-ISSN: 2723-3871

DEVELOPMENT OF MOBILE-BASED FREELANCE SERVICES MARKETPLACE WITH FEATURE-DRIVEN DEVELOPMENT METHODOLOGY

Mochamad Ikhsan Nurdiansyah[*1], Dana Sulistyo Kusumo[2], Arief Ramadhan[3]

[1,2,3]Informatics, School of Computing, Telkom University, Indonesia
Email: [1]ikhsannurdiansyah@student.telkomuniversity.ac.id, [2]danakusumo@telkomuniversity.ac.id, [3]arieframadhan@telkomuniversity.ac.id

(Article received: July 24, 2024; Revision: August 8, 2024; published: October 25, 2024)

Abstract

The rapid advancement of technology demands that humans adapt to the evolving workplace. One of the impacts of technological development is the emergence of freelancers who use online platforms. Online platforms make it easier for freelancers and clients to collaborate. Currently, there are various Freelance Services Marketplace platforms available. However, it is important to note that the developers of these platforms are from outside Indonesia, which poses a risk to the security of people's data and the ease of payment applicable in Indonesia. With the high number of mobile device users in Indonesia,, the development of a mobile-based Freelance Services Marketplace application system that is tailored to local needs and takes into account the data security of Indonesian people, as well as involving local developers, is needed. This research aims to overcome these problems by designing a mobile-based freelance services marketplace application system using the Feature-Driven Development (FDD) method. The selection of FDD as an application development methodology is based on complex feature requirements, good planning management, emphasis on feature quality, and structured. The development process follows the FDD approach starting from feature planning and design, scheduling, implementation, and testing. The application design in this research is based on user needs with a focus on the features needed. This solution is important because it can be used by clients to find freelancers who match their needs and provide opportunities for freelancers to offer their services properly with guaranteed data security and transactions. The main result of this research is that the FDD method can help in the development of a freelance service marketplace application by paying attention to the systematic or structured level, quality, and security of the application.

Keywords: *feature-driven development, freelance, mobile, security, system*

PENGEMBANGAN APLIKASI *FREELANCE SERVICES MARKETPLACE* BERBASIS *MOBILE* DENGAN *METODOLOGI FEATURE-DRIVEN DEVELOPMENT*

Abstrak

Kemajuan teknologi yang begitu pesat menuntut manusia untuk beradaptasi dengan perkembangan dunia kerja. Salah satu dampak dari perkembangan teknologi adalah munculnya pekerja lepas yang menggunakan platform online. Platform online memudahkan para freelancer dan klien untuk berkolaborasi. Saat ini, tersedia berbagai platform Freelance Services Marketplace. Namun, penting untuk dicatat bahwa developer platform tersebut berasal dari luar Indonesia yang mana terdapat resiko dalam keamanan data masyarakat dan kemudahan pembayaran yang berlaku di indonesia. Dengan tingginya jumlah pengguna perangkat mobile di Indonesia, pengembangan sistem aplikasi Freelance Services Marketplace berbasis mobile yang disesuaikan dengan kebutuhan lokal yang memperhatikan keamanan data masyarakat Indonesia dan keterlibatan developer lokal, sangat diperlukan. Penelitian ini bertujuan untuk mengatasi masalah tersebut dengan merancang sebuah sistem aplikasi marketplace jasa freelance berbasis mobile menggunakan metode Feature-Driven Development (FDD). Pemilihan FDD sebagai metodologi pengembangan aplikasi didasarkan pada kebutuhan fitur yang kompleks, manajemen perencanaan yang baik, penekanan pada kualitas fitur, dan terstruktur. Proses pengembangan mengikuti pendekatan FDD yang dimulai dari perencanaan dan perancangan fitur, penjadwalan, implementasi, dan pengujian. Perancangan ini didasarkan pada kebutuhan pengguna dengan fokus pada fitur-fitur yang diperlukan. Solusi ini penting karena dapat digunakan klien menemukan freelancer yang sesuai dengan kebutuhan dan memberikan kesempatan bagi freelancer untuk menawarkan jasa mereka secara baik dengan keamanan data dan transaksi yang terjamin. Hasil utama dari penelitian ini adalah bahwa metode FDD dapat membantu dalam pengembangan aplikasi marketplace jasa freelance dengan memperhatikan tingkat sistematis atau terstruktur, kualitas, dan keamanan aplikasi.

Kata kunci: *feature-driven development, freelance, keamanan, mobile, system*

1. INTRODUCTION

The rapid and unpredictable development of technology in recent years requires humans to adapt to the field of work. The impact of technological advances on the field of work with the emergence of 'freelancers'. Until now, people prefer to work alone or freelancing[1]. This data is supported by data from the Online Labor Index (OLI) in 2017 - 2021 Indonesia is included in the 15 countries with the largest number of freelancers in the world [2]. In reality, the most in-demand fields for freelancers are Music, Writing, Acting, Computer Programming, Graphic Design, Film and Video, etc. [3]. Then freelancers usually use online websites, marketplace apps, and other online sources to get clients or jobs. [1].

The role of online platforms such as the web or apps is very important for freelancers in providing various services to clients in a short time[4]. *Fiverr, Upwork*, and *Fastwork* are online freelancer platforms that provide various freelancer services to the public. *Fiverr* has 2.5 million active users while the *Upwork* platform has 18 million freelancers and 5 million clients [5]. The platform offers potential benefits in terms of flexibility stemming from the ability of workers to decide when they work and the types of tasks they want to perform [6].

The benefits that are considered positive have made this online freelance platform grow rapidly in recent years [6]. Currently, in the Android marketplace, *Play store*, there are already several similar freelance applications available, and most of these applications are owned or originated from outside Indonesia. Therefore, the development of a mobile-based freelance platform application from Indonesia is very important to maintain Indonesia's data security. Indonesia can reflect on China in protecting public data. China has strict regulations on applications outside the country regarding data protection and personal information [7]. Therefore, the existence of developers from Indonesia can provide a security factor for the personal data and information of the Indonesian state. In addition to security factors, there are also other factors related to geographic, demographic, and economic factors [8]. So, the development of freelance service marketplace applications made for the scope of Indonesia must be developed both to maintain the security factor and benefit the geography, demographics, and economy of Indonesia.

Previous research conducted the development of a freelance marketplace application that can facilitate users or clients with a coverage area of Malaysia[9]. Therefore, this research aims to develop a freelance marketplace application that covers the Indonesian region and can facilitate clients and freelancers in Indonesia.

This research aims to develop a mobile-based freelance marketplace application that covers the Indonesian region with a payment system applicable in Indonesia, which can be used by freelancers and clients. FDD (Feature-Driven Development) adopts a practice-based approach to feature[10]. Given that the freelance marketplace application involves two user roles, it naturally requires numerous features. Therefore, FDD is employed with a primary focus on delivering high-quality results throughout the process [11]. The main advantage of FDD is its ability to produce fully functional solutions requiring little to no future maintenance [12]. This ensures that the freelance services marketplace application maintains high-quality features and systems without significant changes needed in the future, emphasizing FDD's prioritization of quality. In contrast, Scrum prioritizes schedules over quality [12]. FDD encompasses five processes: Build Overall Model, Build Feature List, Plan by Feature, Design by Feature, and Build by Feature [13], [14]. With these five steps, the freelance services marketplace application follows a structured process that guarantees the quality of the designed features at every stage with a clear model, structured feature list, and planning. The FDD processes also enforce strict guidelines to identify defects in the system being built and implement coding standards that facilitate the integration of the designed features [15]. Consequently, the coding standards used in the design process simplify the integration of features in the freelance services marketplace application.

The concept of Quality in FDD not only involves testing the code but also includes coding standards, audit measurements, and code metrics [16]. Therefore, by utilizing FDD with coding standards included in the quality concept, it can be implemented to ensure the security of user data, both personal and transactional, within the freelancing application. Thus, the use of the FDD method is highly effective in a systematic and disciplined approach to system design [17], focusing on features and feature quality to ensure the freelance services marketplace application achieves high quality.

The application development is implemented on Android-based mobile devices and the language used is Kotlin. Kotlin has advantages in memory consumption and lighter application size when compared to languages that can build Android applications such as Flutter and React-Native [18]. Kotlin is also considered to be more secure, and compact and another advantage is the possibility of avoiding some of the shortcomings or pitfalls of Java such as Nullability, Mandatory Casts, Long Argument lists, and Data classes [19].

This research will use Kotlin as a programming language for building freelance marketplace applications and tools for building these applications using Android Studio. The selection of the Android studio tool is because the tool is a popular choice as well as an official tool in Android development [20]. After the development is complete, testing will be carried out as a result of this research.

After the application is developed, the research will continue with the evaluation of applications related to aspects of User satisfaction, development efficiency, Code Quality, and security of the freelance services marketplace application.

2. RESEARCH METHODOLOGY

Figure 1 shows the stages of research into the development of a mobile-based freelance services marketplace application.

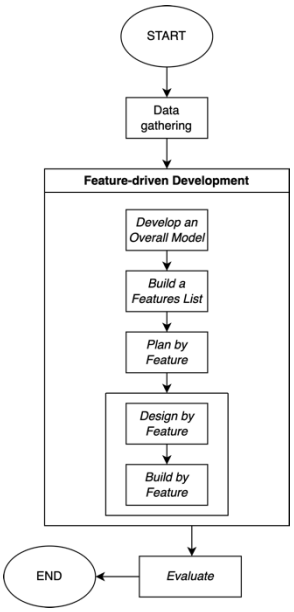


Figure 1. Research Flowchart

2.1. Data Gathering

In this research, the author conducted interviews from February to March 2024 in the city of Bandung, Indonesia. This research involved 10 freelancers aged between 20-22 years and 10 fresh graduates or someone who wants to become a freelancer with the same age range. The interviews conducted in this research were unstructured interviews. The questions asked to the interviewees began with questions related to their demographics. then questions that refer to their daily challenges and related to the ideal features that a freelance services marketplace platform should have that can help them.

Questions that are used to explore what features they expect to find on the freelance services marketplace platform as a benchmark in designing features on the mobile-based freelance services marketplace application and show an analysis of potential areas of freelance work from the interviewers.

Table 1. User Needs Analysis

User needs
18-20 people interviewed have never known a freelance platform
Users need a mobile freelance platform
Users need a freelance platform that can accommodate them and clients so that interactions can run well.
Most users are active in social media as part of promotion. for example, via instagram

Table 1. shows the analysis of the results of interviews related to user needs. This data can be one of the reference features that will be implemented in this research.

2.2 Application of the method

The main stages in this research with the implementation of FDD with the following details:

2.2.1. Develop an Overall Model

At this stage after getting data collection, the author analyzes the determination of what features support the features to be developed. the results of the supporting feature criteria are Login / Register and Dashboard as a link or main area of various features and determination of features that have high priority must be implemented.

Table 2. Feature Request

Number of Persons	Feature Request	Prioritize
16/20	Many users recommend the chat feature for their communication with clients.	High
12/20	User suggests an order feature related to their payment	High
7/20	users suggest skill upload feature and details for clients to know their skill description	Medium
2/20	Users suggest features that are packaged attractively	Low
2/20	User suggests a result upload feature to attach the work to the client.	Low
1/20	User suggests a field type feature	Low

Table 2. is the form of features suggested by the interviewees. the highest result is that the interviewees suggest communication features with a total of 16 out of 20 interviewees and payment features 12 interviewees out of 20. The feature is

then analyzed to determine the priority of features to be built above 50% of the author's interviewees determine for high priority, then < 50% and > 25% of the author's interviewees determine for medium priority and the last is < 25% of the author's interviewees determine to be low priority. This data will be a reference in forming a list of features that will be developed into an android mobile-based freelance services marketplace application.

Figure 2 Shows the design of the lo-fi design of the freelance service marketplace application. The analysis of the design of the lo-fi is based on the list of features that will be on the results of the interview. Lo-fi is implemented based on the results of user experience. *Figma* is used as a tool in making the design of both lo-fi and hi-fi in the next stage.

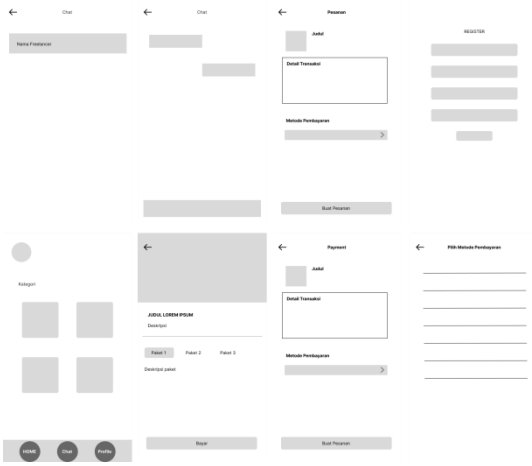


Figure 2. Lo-fi User Interface

The application to be developed has 3 main actors. the first actor is an admin who will be implemented into a web using the *Laravel* language that can control the list of users and transactions and create APIs for mobile application purposes. the second is a client, which is a user who will order services to freelancers. and the last is freelancers, which are people who will offer their expertise services. both client and freelancer actors will be developed with the *Kotlin* language *android studio* application. the limit of android usage is *android 11* and above. Related to payment, the author uses the third party *Midtrans* in managing money transfers and adheres to the policy in the fee system in each transaction.

2.2.2. Build a Features List

At this stage, designing or compiling a list of features that will be realized in the freelance services marketplace application and which have been combined and adjusted to the relationship between features or combining with supporting features.

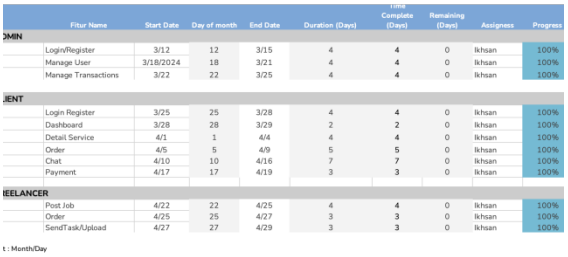
Table 3. List of Feature

Actor	Fitur Name
Admin	Login/Register
	Manage User
	Manage Transactions
Client	Login/Register
	Dashboard
	Service Detail
	Order
	Payment
	Chat
Freelancer	Login/Register
	Post Job
	Get Order
	Send Task

Table 3. displays a list of features sorted according to their dependency on each other. The list of features includes the three actors in the freelance services marketplace application.

2.2.3. Plan by Features

At this stage, the implementation schedule is meticulously set, and the progress of each feature implementation is closely monitored using a Gantt chart. The Gantt chart includes columns for start and end dates, day of the month, duration, time completed, remaining days, assignees, and progress percentage. The comprehensive tracking via the Gantt chart, underscores the effective management and systematic approach to ensuring that all functionalities are developed on schedule, highlighting the project's organized workflow and efficiency. Figure 3 shows the Gannt chart that has been designed to plan each feature in detail and structured project management.



	Fitur Name	Start Date	Day of month	End Date	Duration (Days)	Time Complete (Days)	Remaining (Days)	Assignee	Progress
OMIN									
	Login/Register	3/12	12	3/15	4	4	0	Ikhsan	100%
	Manage User	3/18/2024	18	3/21	4	4	0	Ikhsan	100%
	Manage Transactions	3/22	22	3/25	4	4	0	Ikhsan	100%
IENT									
	Login Register	3/25	25	3/28	4	4	0	Ikhsan	100%
	Dashboard	3/28	28	3/29	2	2	0	Ikhsan	100%
	Detail Service	4/1	1	4/4	4	4	0	Ikhsan	100%
	Order	4/5	5	4/9	5	5	0	Ikhsan	100%
	Chat	4/10	10	4/16	7	7	0	Ikhsan	100%
	Payment	4/17	17	4/19	3	3	0	Ikhsan	100%
EELANCER									
	Post Job	4/22	22	4/25	4	4	0	Ikhsan	100%
	Order	4/25	25	4/27	3	3	0	Ikhsan	100%
	SendTask/Upload	4/27	27	4/29	3	3	0	Ikhsan	100%

t : Month/Day

Figure 3. Gannt Chart

2.2.4. Design by Features

At this stage designing a model design of the freelance service marketplace application. The design here is in the form of a class diagram design from each stack both Backend and Frontend. The model design that is built provides an overview of the database and functions used in coding and provides an overview of the pattern architecture. Figure 4 shows the results of designing the Backend

class diagram and Figure 5 shows the results of designing the frontend class diagram.

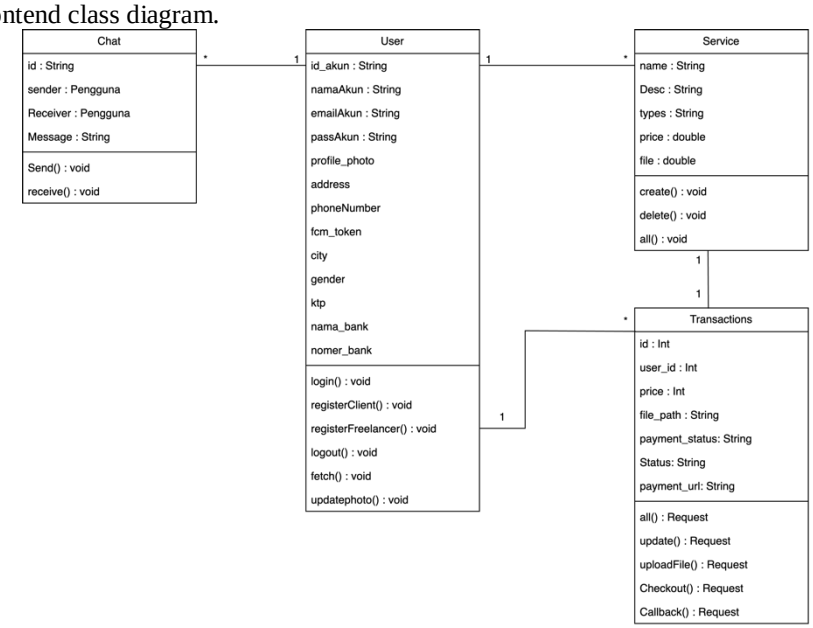


Figure 4. Backend Class Diagram

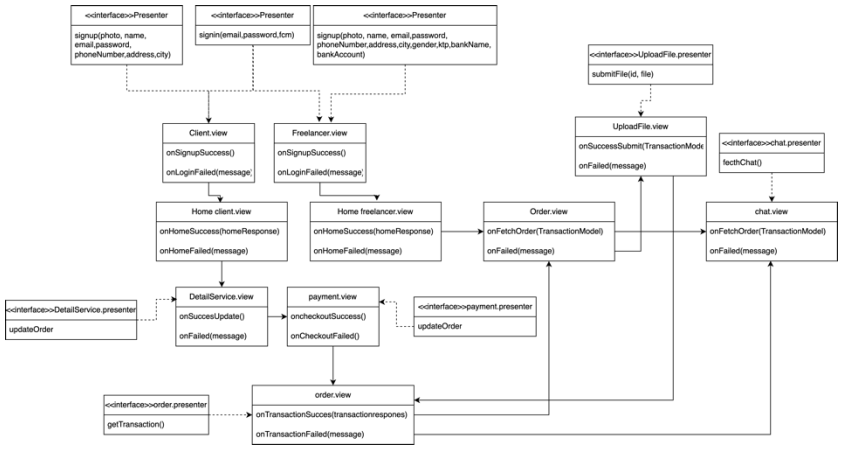


Figure 5. Frontend Class Diagram

At this stage, each iteration of the feature also determines the Hi-fi User Interface (UI) of each feature. Figure 6 shows Hi-fi design on freelance service marketplace application.

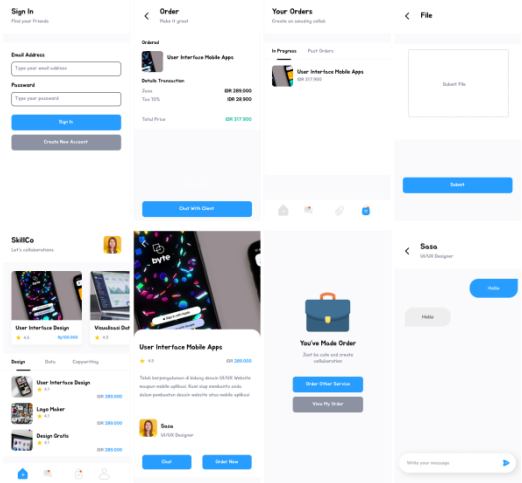


Figure 6. Hi-fi User Interface

2.2.5 Build by Features

At this stage, after finalizing the design models for the backend and frontend stacks, coding implementation began using Visual Studio Code and Android Studio. The development process started with the admin actor by creating a dedicated admin website to monitor users and transactions. This was done using Laravel to build the backend REST API. Coding development for each feature was carefully scheduled, following an iterative approach where each iteration focused on completing one feature. This structured schedule ensured that each feature was developed thoroughly before moving on to the next. Once a feature is completed, unit testing is performed using Blackbox testing to validate functionality and ensure that the feature meets the specified requirements with a total of 81 Blackbox testing test cases for all features. This rigorous approach not only ensures high-quality code but also

enables early detection and resolution of issues, ultimately leading to a more stable and reliable application.

3. RESULT

This research produces a mobile-based freelance services marketplace application with two actors that have been completed. Figure 7 shows the results of the development of a mobile-based freelance services marketplace application with various features that can use by clients and freelancers.

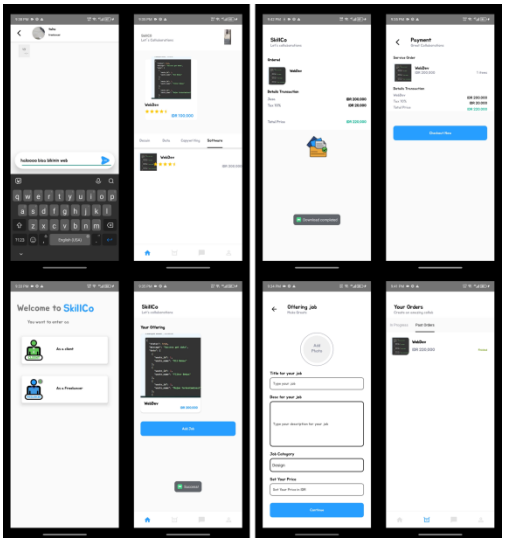


Figure 7. Freelancer Apps

3.1. User Satisfaction

The assessment of user satisfaction when using the freelance services marketplace application is conducted through a survey with a scale of 1-5. This survey includes a total of 20 respondents, consisting of 10 freelancers and 10 individuals interested in or unfamiliar with freelancing who agreed to be interviewed beforehand. In the context of ease of use, questions were designed to measure how easy respondents found the application to use after trying it. Then, the respondents were asked to evaluate whether each feature functions well and meets their expectations, ensuring that the features in the application operate smoothly.

Table 4. Distribution Score

Aspect	Category	Total Respondent	Percentage
Easy of Use	Strongly Agreed	13/20	65%
Easy of Use	Agreed	7/20	35%
Feature are Working Well	Strongly Agreed	10/20	50%
Feature are Working Well	Agreed	10/20	50%

Table 4 shows the score distribution results with most respondents (65%) strongly agreed with the ease of use, indicating high satisfaction in this regard. 35% agreed, indicating that there are some areas that could possibly be further improved. As for the Features are Working Well aspect, the distribution (50% strongly agreed and 50% agreed) indicates that respondents feel the features are working well.

Table 5. Assesment Aspect

Assessment Aspect	Score Average
Ease of Use	4.65
Features are working well	4.5
Total	4.575

Based on the survey results shown in Table 5, it is evident that the aspect of ease of use has an average score of 4.65. This data was obtained from the interviews, where 65% of the 20 respondents chose a scale of 5 (strongly agree) and 35% chose a scale of 4 (agree). The next aspect, well-functioning features, has an average score of 4.5. This data was gathered from the interviews, indicating that 50% of the 20 respondents chose a scale of 5 (strongly agree) and 50% chose a scale of 4 (agree). Therefore, the overall user satisfaction level has an average score of 4.575 out of 5, showing that most users are satisfied with the designed application.

In designing this application from the plan contained in the list of features in the FDD method with a total of 13 features. A total of 13 features can be realized properly. Table 6 shows the date of realization of each feature that has been built.

Table 6. Status Completion Feature

Feature	Completion Data	Status
Login/Register Admin	15-Mar-2024	Successful
Manage User	21-Mar-2024	Successful
Manage Transaksi	25-Mar-2024	Successful
Login/Register Client	28-Mar-2024	Successful
Dashboard Client	29-Mar-2024	Successful
Detail Service	04-Apr-2024	Successful
Order	09-Apr-2024	Successful
Chat	16-Apr-2024	Successful
Payment	19-Apr-2024	Successful
Login/Register Freelancer	21-Apr-2024	Successful
Post Job	25-Apr-2024	Successful
Order	27-Apr-2024	Successful
Send Task/Upload File	29-Apr-2024	Successful

3.2. Development Efficiency

In this study, the initial estimation for the development of the freelance services marketplace

application was approximately 3 months. The project was completed in less than two months, from March 15, 2024, to April 29, 2024. Figure 8 demonstrates the effectiveness of FDD in project completion time compared to the initial estimation. It shows that the project, utilizing FDD, was completed 48,91% faster than the estimated time, highlighting the method's effectiveness in shortening the project duration compared to the initial estimate.

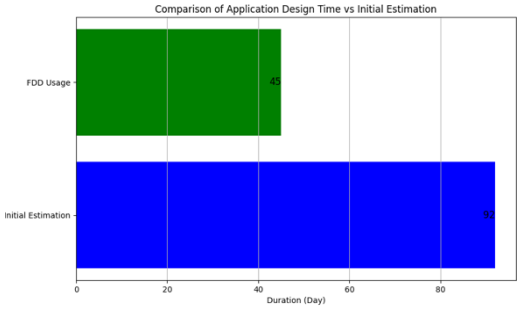


Figure 8. comparison of Application Design time vs initial Estimation

The total development time for the application project was completed within 45 days, starting from March 15, 2024, to April 29, 2024. Figure 9 shows the number of features completed each month. This data serves as an indicator of the productivity level during the development process.

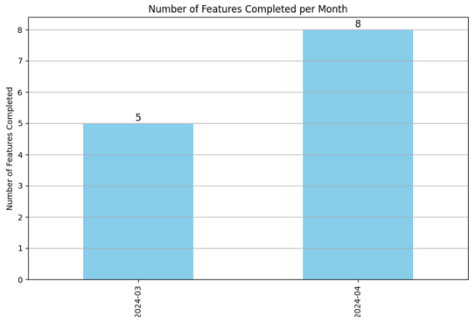


Figure 9. Number of features completed per Month

3.3. Code Quality

During the application development, before the application was released and used by users, two bugs were discovered. The first bug was related to the camera feature. When the camera was selected to input job images, the system reverted to the freelancer dashboard. However, when the gallery was selected, the input was accepted. The second bug was related to the chat feature. The messages sent by the client appeared on the freelancer's side or the recipient's layout instead of the sender's layout. The Third dashboard doesn't reload data after click back to dashboard.

$$Bug\ per\ feature\ ratio = \frac{number\ of\ bugs}{number\ of\ feature} \quad (1)$$

$$Bug\ per\ feature\ ratio = \frac{3}{13} \approx 0.2307 \quad (2)$$

These results show that the ratio of bugs found per feature is quite low, indicating that the development process with the FDD method is successful in producing features that have good quality.

$$Defect\ Density = \frac{Total\ Defect}{Line\ of\ Code} \quad (3)$$

$$Defect\ Density = \frac{3}{6448} \approx 0.000465 \quad (4)$$

Defect density provides an important indication of overall code fitness, with lower values indicating greater stability and fewer bugs per line of code. In this study, the defect density was approximately 0.000465, or about 0.0465%, reflecting the relatively high level of conformance in the application design code. This low defect density indicates the strong quality assurance process implemented during development, which emphasizes meticulous coding standards and thorough testing procedures. Such a measure is critical to supporting claims regarding system quality and reliability. By offering quantifiable metrics, they provide valuable insight into the software development process, underscoring the effectiveness of FDD and the overall integrity of the application. This analysis not only reinforces the credibility of the developed system but also highlights the importance of maintaining strict quality control in software engineering practices.

3.4. Security

The security of personal data is a top priority in mobile application development. To protect user information, this application development uses Laravel's standard hashing for password encryption. The hashing is performed using the Bcrypt algorithm, which provides a high level of security and prevents the retrieval of passwords to their original form, even if the admin can access the data. Thus, the implementation of this hashing technique effectively meets the security objective, ensuring that personal data is not vulnerable to common data attacks.

In this application, all transactions use Midtrans as a third-party payment gateway. Midtrans is licensed and registered with Bank Indonesia under number 20/320/DKSP/Srt/B. Regarding security, Midtrans holds an ISO 27001 data certificate and uses an in-house fraud detection system called Aegis. This system helps maintain a low payment failure rate

with fraud levels below 0.1%. As a result, the use of the FDD method supports the implementation of the payment gateway in the freelance application, ensuring transaction security standards and compliance with payment systems applicable in Indonesia.

4. DISCUSION

4.1. User Satisfaction

High satisfaction on the Ease-of-Use aspect indicates that the interface or use of the system is good, with most respondents feeling very satisfied addressed by the data distribution score and Assessment aspect. However, there may be small details that can still be improved to further enhance the user experience such as responsiveness and interactive features. Meanwhile, on the aspect of well-functioning features, there is an even split between respondents who strongly agree and those who agree. This indicates that while the features are adequate, there is still room for further improvement. An in-depth evaluation of feature feedback can help identify features that need to be improved, with the aim of achieving higher levels of user satisfaction. From these results, the use of Feature-Driven Development (FDD) is not the only method that can help in improving aspects of user satisfaction, so other approaches are also worth considering according to project needs.

4.2. Structured Process and Quality Assurance

The results show that the FDD approach exhibits a structured process and strict quality assurance, resulting in a robust and reliable application. When compared to other methodologies, such as Scrum, Scrum prioritizes flexibility at each step, which contrasts with FDD which emphasizes systematic planning and execution. every feature designed will be quality-assured through five predefined processes. These five steps provide a clear roadmap and minimize the risk of deviation, which is especially important for projects with high feature complexity.

4.3. Feature Approach

FDD's focus on feature-based development is well aligned with the needs of applications that have complex features especially with the needs of applications that have two actors. While Scrum methodology provides flexibility in developing complex features, FDD offers a more concentrated and disciplined approach. By maintaining a structured feature list and a clear development plan for each feature, FDD ensures the completeness and quality of each feature. This structured approach is particularly advantageous for freelance services marketplace apps, which require functionality. Moreover, the clarity provided by a structured plan helps in tracking progress and maintaining high quality standards throughout the development process. This not only improves the usability of the app but also ensures that it meets the complex and evolving demands of its users, ultimately resulting in a more reliable and user-centric product.

4.4. Adaptability and Responsiveness

Based on the results and a series of studies conducted, it is evident that while Feature-Driven Development (FDD) excels at delivering high-quality, feature-rich solutions through robust initial design and planning, it has limitations in adapting to change. FDD's structured approach provides a solid foundation that minimizes the need for future modifications. However, when compared to Scrum, which is renowned for its adaptability to changing needs and its iterative development process, FDD does less well in dynamic environments. Scrum's sprint cycle allows for continuous feedback and adjustments. This makes Scrum particularly advantageous for projects with frequently changing needs, as it can quickly incorporate new insights and feedback. Thus, while FDD is highly effective for projects where the scope and requirements are well defined from the start, Scrum offers superior flexibility and responsiveness, making it more suitable for dynamic projects that require adaptability.

4.5. Development Speed and Efficiency

The results also show that using FDD can deliver a project 48,91% faster compared to initial estimates. While this is not the primary focus of FDD, various factors such as developer skill levels can influence this outcome. In this aspect, Scrum can potentially achieve faster project completion times if the requirements are well-defined. Scrum's high flexibility and adaptability to changes contribute to its efficiency. Although FDD provides a more systematic and clearer schedule, Scrum can be more efficient for projects that require rapid development.

5. LIMITATIONS

Although this research provides good insights into the development of a Mobile-Based Freelance Services Marketplace Application System with Feature-Driven Development (FDD) Methodology, several limitations need to be noted. Firstly, the sample size of the respondents involved in the interviews may not cover a wide diversity of users as the scope is only around the city of Bandung, and the age range lacks diversity and other demographic reasons, especially in terms of user background and user experience in freelancing or using the platform. This may limit the generalizability of the findings to a wider population of users and affect the description of features that are ideal for them. Secondly, in the collection of qualitative feedback

data, there is a possibility of subjective bias in the interpretation of data provided by respondents or respondents' understanding of the questions asked. Thirdly, technical limitations and available resources may affect the depth of analysis and the scope of implementation of the design proposed in this study. Finally, the use of FDD is not fully robust in all aspects; there are some areas where other methods can be more reliable. Nonetheless, recognizing these limitations provides opportunities for the development of more comprehensive and in-depth research in the future.

6. CONCLUSION AND FUTURE WORK

The results show that FDD is an effective methodology for developing a freelance service marketplace application with a focus on high-quality features with excellent results in the aspects of User satisfaction, Efficiency, Code Quality, and Security. While Scrum offers advantages in terms of adaptability and efficiency, the FDD process by offering structure, systematic, and focus on feature-based development makes it well suited for projects that require high-quality feature solutions. From these findings, it is recommended that application development should consider using FDD to ensure that each feature produced is of high quality, thereby enhancing user satisfaction, efficiency, security and code quality overall.

Future research could explore a hybrid approach with FDD within the framework of other methods to further improve all aspects of development. This would allow for the enhancement of the strengths of both methodologies, combining the rigorous planning and quality assurance of FDD with the capabilities of other methods, which would ultimately create a robust development process. In addition, future application development can be focused on improving the user experience with more responsive and interactive features. application development can also be focused on using a programming language that can design IOS-based applications or can use a hybrid language that can design IOS and android without reducing the quality of the resulting features.

REFERENCE

[1] N. U. Haq, A. A. Raja, S. Nosheen, and M. F. Sajjad, "Determinants of client satisfaction in web development projects from freelance marketplaces," *International Journal of Managing Projects in Business*, vol. 11, no. 3, pp. 583–607, 2018, doi: 10.1108/IJMPB-02-2017-0017.

[2] F. Stephany, O. Kässi, U. Rani, and V. Lehdonvirta, "Online Labour Index 2020: New ways to measure the world's remote freelancing market," *Big Data Soc*, vol. 8, no. 2, 2021, doi: 10.1177/20539517211043240.

[3] 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, doi: 10.1016/j.procs.2019.08.150.

[4] J. Brunzel, "An empirical analysis of linguistic styles in new work services: The case of Fiverr.com," *European Management Review*, 2023, doi: 10.1111/emre.12562.

[5] A. Blaising and L. Dabbish, "Managing the Transition to Online Freelance Platforms: Self-Directed Socialization," *Proc ACM Hum Comput Interact*, vol. 6, Nov. 2022, doi: 10.1145/3555201.

[6] A. Hannák, A. Mislove, C. Wagner, M. Strohmaier, D. Garcia, and C. Wilson, "Bias in Online freelance marketplaces: Evidence from TaskRabbit and Fiverr," in *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, Association for Computing Machinery, Feb. 2017, pp. 1914–1933. doi: 10.1145/2998181.2998327.

[7] L. Jia and L. Ruan, "Going global: Comparing chinese mobile applications' data and user privacy governance at home and abroad," *Internet Policy Review*, vol. 9, no. 3, pp. 1–22, 2020, doi: 10.14763/2020.3.1502.

[8] A. Hidayati, E. K. Budiardjo, and B. Purwandari, "Software Engineer Competencies in Global Software Development: An Indonesian Perspective," *Tehnicki Vjesnik*, vol. 29, no. 2, pp. 683–691, Apr. 2022, doi: 10.17559/TV-20210103153044.

[9] M. I. Hanip and N. A. Hamid, "MyFREELANCER App," *Applied Information Technology And Computer Science*, vol. 3, no. 2, pp. 1130–1143, 2022, doi: 10.30880/aitcs.2022.03.02.068.

[10] F. Anwer *et al.*, "Comparative Analysis of FDD and SFDD Zahid Nawaz Xint Solutions Comparative Analysis of FDD and SFDD," 2018. [Online]. Available: https://www.researchgate.net/publication/323560594

[11] S. Al-Saqqa, S. Sawalha, and H. Abdelnabi, "Agile software development: Methodologies and trends," *International Journal of Interactive Mobile Technologies*, vol. 14, no. 11, pp. 246–270, 2020, doi: 10.3991/ijim.v14i11.13269.

[12] S. S., S. Ali, and A. Babu, "A Hybrid Agile model using SCRUM and Feature Driven Development," *Int J Comput Appl*, vol. 156,

no. 5, pp. 1–5, Dec. 2016, doi: 10.5120/ijca2016912443.

[13] M. Javanmard and M. Alian, "Comparison between Agile and Traditional software development methodologies," *Cumhuriyet University Faculty of Science Science Journal (CSJ)*, vol. 36, no. 3, p. 36, 2015, [Online]. Available: http://dergi.cumhuriyet.edu.tr/cumuscij©2015

[14] S. Aftab *et al.*, "Using FDD for small project: An empirical case study," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 3, pp. 151–158, 2019, doi: 10.14569/IJACSA.2019.0100319.

[15] Shama PS and Shivamanth A, "A Review of Agile Software Development Methodologies," *International Journal of Advanced Studies in Computers, Science and Engineering*, vol. 4, 2015. [Online]. Available: www.ijascse.org

[16] S. Goyal, "Major Seminar On Feature Driven Development Agile Techniques for Project Management and Software Engineering," Jennifer Schiller Chair of Applied Software Engineering. Aug 2008, pp. 90-91.

[17] S. G. Tetteh, "Empirical Study of Agile Software Development Methodologies: A Comparative Analysis," *Asian Journal of Research in Computer Science*, vol. 17, no. 5, pp. 30–42, Feb. 2024, doi: 10.9734/ajrcos/2024/v17i5436.

[18] B. Suri, S. Taneja, I. Bhanot, H. Sharma, and A. Raj, "Cross-Platform Empirical Analysis of Mobile Application Development frameworks: Kotlin, React Native and Flutter," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Dec. 2022. doi: 10.1145/3590837.3590897.

[19] R. Coppola, L. Ardito, and M. Torchiano, "Characterizing the transition to kotlin of android apps: A study on F-Droid, Play Store, and GitHub," *WAMA 2019 - Proceedings of the 3rd ACM SIGSOFT International Workshop on App Market Analytics, co-located with ESEC/FSE 2019*, pp. 8–14, 2019, doi: 10.1145/3340496.3342759.

[20] A. Ribeiro, J. F. Ferreira, and A. Mendes, "EcoAndroid: An Android Studio Plugin for Developing Energy-Efficient Java Mobile Applications." in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*, pp. 62-69, IEEE, 2021.