Penerbit Akademia Baru

Journal of Advanced Research Design

Journal homepage:
https://akademiabaru.com/submit/index.php/ard
ISSN: 2289-7984


# Implementing Multiple Dynamic Flip Algorithm: *Bread on Mobile* Case Study

Mustafa Man[1], Wan Aezwani Wan Abu Bakar[2,*], Rosaida Rosly[1], Terry Yoew[3], Evizal Abdul Kadir[4]

[1] Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu (UMT), 21030 Kuala Nerus, Terengganu, Malaysia
[2] Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UniSZA), Besut Campus, 22200 Besut, Terengganu, Malaysia
[3] Upaya Vista Sdn. Bhd., No. 67, Jalan Perindustrian 4/KUB, Kawasan Perindustrian Meru Selatan, 41050 Klang, Selangor, Malaysia
[4] Universitas Islam Riau, Jl. Kaharuddin Nasution 113, Pekanbaru, 28284 Riau, Indonesia

**ARTICLE INFO**

***Article history:***
Received 29 August 2024
Received in revised form 22 December 2024
Accepted 4 April 2025
Available online 30 April 2025

***Keywords:***
Multiple dynamic flip algorithm; mobile application; database; internet

**ABSTRACT**

Mobile apps developers encounter a range of challenges and problems during the development. These issues can vary depending on the platform (iOS, Android, cross-platform) and the specific project, but here are some common problems faced by mobile app developers. The need for a structured mechanism in mobile apps development is urgently crucial to reduce the cost of manpower charges in conjunction with the simple and easy framework to assist mobile apps developer. In response to these scenarios, this paper introduces the Multiple Dynamic Flip Algorithm (MDFA), a new algorithm in developing mobile-based applications that serves as the mobile app's framework. The algorithm developed allows the data to be stored in the database and the administration of the system can only change the data according to the categories of data that have been systematically arranged in the database. This technique allows changes to information and data to be done automatically without requiring users to run back to mobile-based programming. Process of updating information needs to be generated back in mobile-based programming and needs to be uploaded back in the play store and users must download the application software back. By using this MDFA algorithm, the problem is no longer needed and can be solved more quickly. Implementation results of MDFA via university-industry real case study through Bread on Mobile (BOM) apps used for mobile apps bread ordering.

## 1. Introduction

A mobile application, commonly known as a mobile app, is a software program designed to run on various mobile devices, such as smartphones, tablets, and smart wearables. According to Alotaibi *et al.,* [1], statistics from the United States indicate that, on average, each user has around 100 apps installed on their device. Users tend to check their smartphones approximately 63 times a day, and a significant 87% of mobile users glance at their phones for at least an hour before going to sleep, while 69% do the same for a minimum of five minutes [2]. Surprisingly, about 79% of users may stop using a digital product just one day after installation. These statistics underscore the significant role

* *Corresponding author*
*E-mail address: wanaezwani@unisza.edu.my*

https://doi.org/10.37934/ard.129.1.2032

mobile apps play in the digital media landscape, with 57% of digital media users engaging with mobile apps.

As of 2021, there are approximately 7 billion mobile users worldwide. During this year, the total number of downloaded mobile apps is expected to reach 258 billion, a significant increase from the 168 billion recorded in 2017. Additionally, the consumer spending in app stores is on the rise, showing a remarkable 92% increase, amounting to a total of $157 billion on a global scale. These trends highlight the increasing importance of internet usage among mobile users in our rapidly evolving and dynamic world. Figure 1 showcases the statistics on Malaysia mobile internet users in Jan 2020.

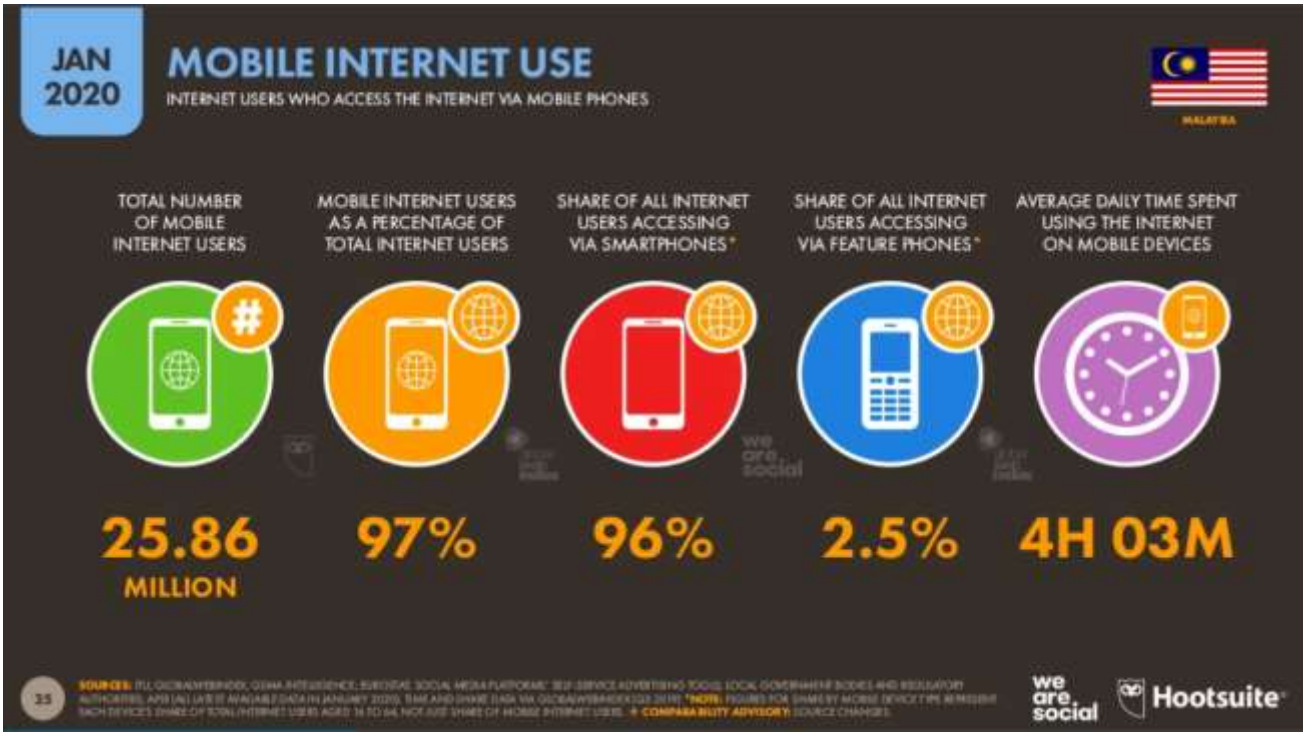



**Fig. 1.** Malaysia population on mobile internet users

The report by Razak *et al.,* [3] highlights on 31.54 million out of 32.7 million Malaysia population are the mobile internet users which indicates 97% of mobile internet users. It also portrays that 96% sharing resources via mobiles. They spend 4 hours and 3 minutes in daily average. The statistics has signalled for a rapid emerging lifestyle on using smart devices. The situation urges for a fast and easy strategy in apps development to fulfil the current market needs.

### *1.1 Mobile Website vs. Mobile Apps*

From both the developer's and the user's perspectives, the challenges revolve around the decision of whether to prioritize the creation of a downloadable mobile application, commonly referred to as "apps," or to develop a mobile website, or possibly even both [4]. Initially, mobile websites and apps may appear quite similar, making it crucial to determine which option aligns better with your specific requirements [5]. Several factors can influence this choice, including the target audience, available budget, intended purposes, and the necessary features. The trend in mobile browsing has emerged towards the changing in mobile internet surfing where internet users may have choices between mobile websites versus mobile apps [6]. Mobile websites primarily offer informative content, whereas web apps deliver interactive content to users. When it comes to web apps, there is an increasing utilization of various elements such as text, images, videos, social

networking features, contact forms, geolocation services, games, and couponing, surpassing what is typically found on mobile websites (refer to Figure 2).


**Fig. 2.** Different aspects between mobile websites and mobile apps

A mobile website comprises browser-based HTML pages interconnected and accessible over the Internet using Wi-Fi or through 3G to 5G networks [7]. What sets a mobile website apart from a standard website is its specific design catering to small, compact handheld displays and touch-screen interfaces. Currently, interactive, and responsive web design has become the new standard for websites. Key attributes of this standard encompass mobile-friendly websites with adaptable dimensions suitable for a wide range of devices, from desktop computers to tablets and handheld smartphones. These mobile websites should possess the capability to provide a responsive layout, accommodating text content, data, images, and videos [8]. Additionally, they may include mobile-specific features like click-to-call functionality (allowing users to dial phone numbers directly) and geographic location maps.

Whereas mobile apps or simply Apps, short for applications, are software programs that users download and install directly onto their mobile devices, as opposed to accessing them through a web browser. Users typically visit device-specific marketplaces, such as Apple's App Store or Google Play Store, to search for and acquire the apps through their respective mobile operating systems. These apps can retrieve data content from the Internet, much like websites do, or they can pre-download content for offline access, allowing users to use them without an active internet connection.

### *1.2 Current Scenario*

Citing Papadopoulos *et al.,* [9], mobile app developers encounter the following top three challenges: Challenge #1: Lack of a Unified Mobile Development Strategy, Challenge #2: Issues with Platform Compatibility, and Challenge #3: Concerns about User Experience/User Interface (UX/UI) and Meeting Customer Preferences. Regarding the first challenge, there is a lack of standardized procedures (SOP) for initiating mobile app development. Developers have the flexibility to choose whether to start by setting up a database or by designing the user interface, aiming to enhance user navigation and overall experience.

The second challenge revolves around the difficulty of selecting the most suitable platform compatibility, which poses a significant dilemma for both developers and users. Users typically request that apps be compatible with their existing operating system platform. However, developers may encounter certain constraints or complexities that necessitate adjustments.

The third challenge pertains to the user's current experience with using apps. Users often prefer interfaces that closely resemble what they are accustomed to, as it provides a greater sense of comfort and familiarity while navigating the apps. However, this preference becomes problematic because mobile apps must adapt to the smaller screen size of mobile devices.

Mobile development landscape is continually evolving, and new challenges may have emerged since then [10,11]. Some of the key issues mobile developers have to deal with are fragmentation, security, privacy regulations, app performance, cross-platform development, the apps stores policies, how user engages and retention, testing and quality assurance, monetization strategies, maintenance and updates as well as staying up to date with emerging technology infrastructures like 5G, Virtual reality (VR) and Augmented Reality (AR) [12-15]. Fragmentation happens when Android has a wide range of devices with different screen sizes, resolutions, and hardware capabilities. This makes it challenging for developers to ensure their apps work seamlessly across all devices [16-18]. Mobile apps are vulnerable to various security threats, such as data breaches, malware, and hacking. Developers need to prioritize security and regularly update their apps to address vulnerabilities. Privacy Regulations refers to compliance with data privacy regulations like GDPR and CCPA is essential. Developers must ensure that their apps collect and handle user data in a compliant manner, which can involve significant changes in app design and functionality. Users expect fast and responsive apps. Performance optimization is an ongoing concern, and developers must consider factors like app startup time, responsiveness, and battery consumption. The Cross-Platform Development regards to the situation where developers struggle with dilemma of whether to build native apps for iOS and Android separately or opt for cross-platform development frameworks like React Native, Flutter, or Xamarin. Each approach has its pros and cons. Other issues pertain to App Store Policies where developers must navigate the guidelines and policies of app stores (Apple App Store and Google Play Store). Violating these policies can result in app removal or rejection. The issues of User Engagement and Retention is quite challenging where it’s not just about building the app but also keeping users engaged. Developers need to consider strategies for user retention, push notifications, and in-app marketing. Comprehensive testing, including compatibility testing on various devices and operating systems, is crucial. Automated testing tools can help streamline this process [19].

Other key issues rely on the Monetization Strategies. Deciding how to monetize the app (e.g., through ads, in-app purchases, or subscriptions) and implementing those strategies effectively can be challenging. Keeping the app up to date with the latest OS versions, devices, and user expectations is an ongoing task for Updates and Maintenance [20]. Finally, the Emerging Technologies like 5G, augmented reality (AR), and virtual reality (VR) is important for developers looking to adopt new features and capabilities in their apps.

Considering the challenges that give rise to issues and dilemmas frequently encountered by mobile app developers, this paper proposes the implementation of a novel approach known as the Multiple Dynamic Flip Algorithm (MDFA) for the development of mobile-based apps regardless of the mobile operating system platforms (i.e., Android or iOS).

The remaining sections are structured as follows: Section 2 delves into the research methodology used in the Mobile Development Framework for Anti-Drugs Awareness (MDFA) project, while Section 3 elaborates on the practical application of MDFA in constructing Bread on Mobile (BOM). In Section

4, the outcomes and ensuing discussions are presented, while Section 5 offers the conclusion and outlines potential future recommendations.

## 2. Methodology

The MDFA framework is introduced with the aim of alleviating the challenges associated with the frequent changes and updates requested by users to mobile app developers. The MDFA methodology encompasses eight distinct stages, outlined as follows: Stage 1 involves Development and Programming where the front-end (User Interface) and the back-end (database structure) is conducted for the BOM mobile apps. Then, Stage 2 encompasses the Compilation of both front-end and the back-end development. Thirdly in Stage 3, which focuses on Vuforia conversion such that the Application Programmer's Interface (API) generated is then converted into Vuforia compatible format. Then Stage 4 entails the generation of the Android Package Kit (APK) file for the BOM apps. Next the Stage 5 pertains to the uploading of the APK into Google Play Store or Apple Store. Moving forward, the Stage 6 addresses if any change/update process request prior to moving on to the Stage 7 handles change/update requests directly within the database management system in the web portal. Finally, the Stage 8 marks the concluding step, involving the refreshing of the apps. Figure 3 provides an overview of the MDFA methodology in contrast to the conventional approach used in mobile app development. MDFA introduces an innovative framework in which any required changes or updates are specifically addressed in Stage 7, where modifications are implemented within the database management system. Subsequently, these new updates seamlessly take effect in the apps by simply refreshing them in Stage 8.

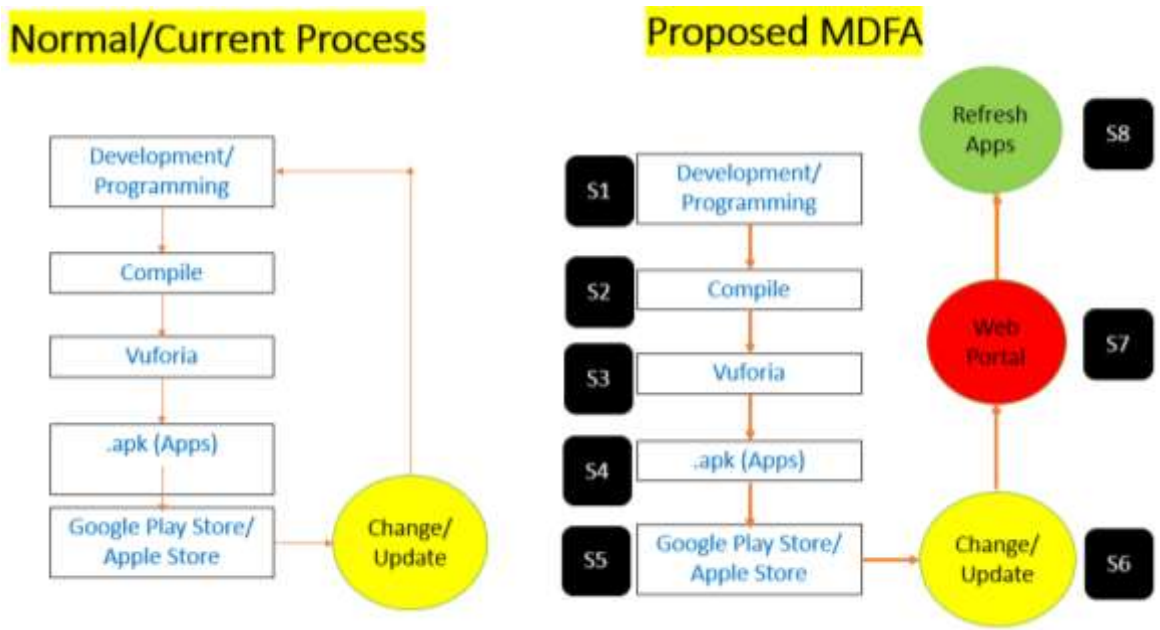



**Fig. 3.** Innovation done in MDFA methodology

## 3. Results

### *3.1 Computing Environment*

The experimentation is conducted with the machine specification as follows; Operating under Windows 11 Home, VivoBook ASUS model, 11th Gen Intel(R) Core (TM) i5-1135G7 @ 2.40GHz

Processor with 12288MB RAM. The MDFA is implemented in the development of Bread on Mobile (BOM).

### *3.2 Development Process*

Figure 4 illustrates the straightforward two-step approach employed in MDFA for mobile app development. In this approach, the back end is represented by the database engine, while the front end utilizes the Flutter framework with the DART programming language. In the context of the AADK case study, multiple video flips are seamlessly integrated into the application.

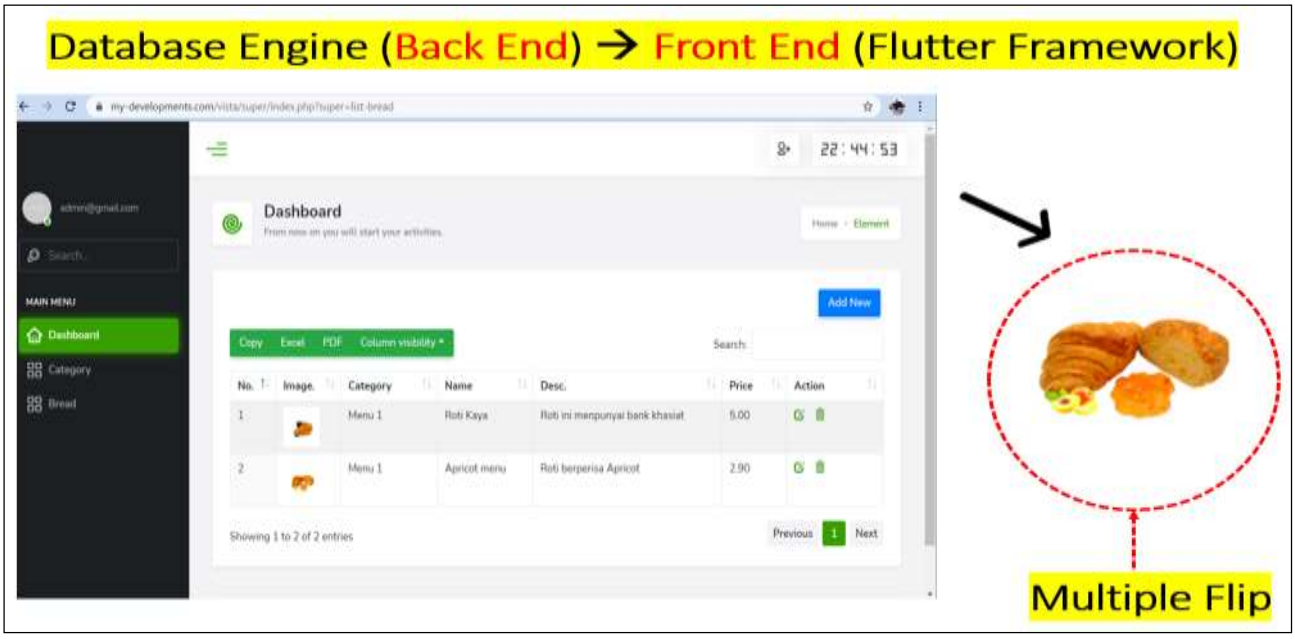



**Fig. 4.** Database engine or back-end versus MDFA framework or back-end using flutter

Figure 5 through Figure 10 provide a glimpse into the development of Bread on Mobile (BOM) from the perspective of the database engine, which represents the back end. Figure 5 offers an overview of back-end or the primary page also referring to dashboard, displaying statistics related to total users (1), category (5), gallery (28) and video (28) in the database while Figure 6 presents the BOM welcoming pages at the front-end side.

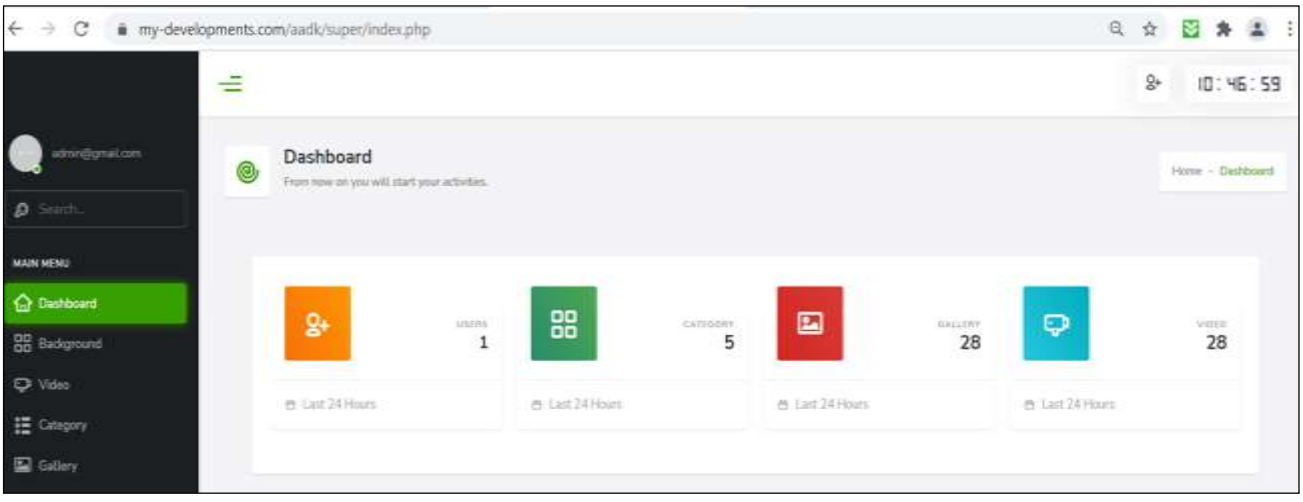
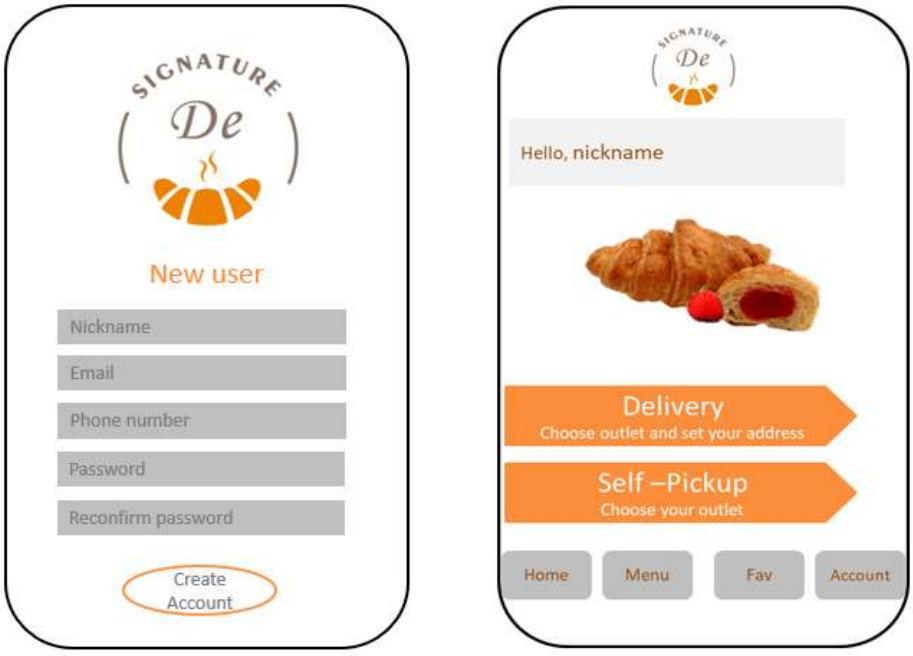



**Fig. 5.** The BOM user's dashboard


**Fig. 6.** The BOM Welcoming and Login pages

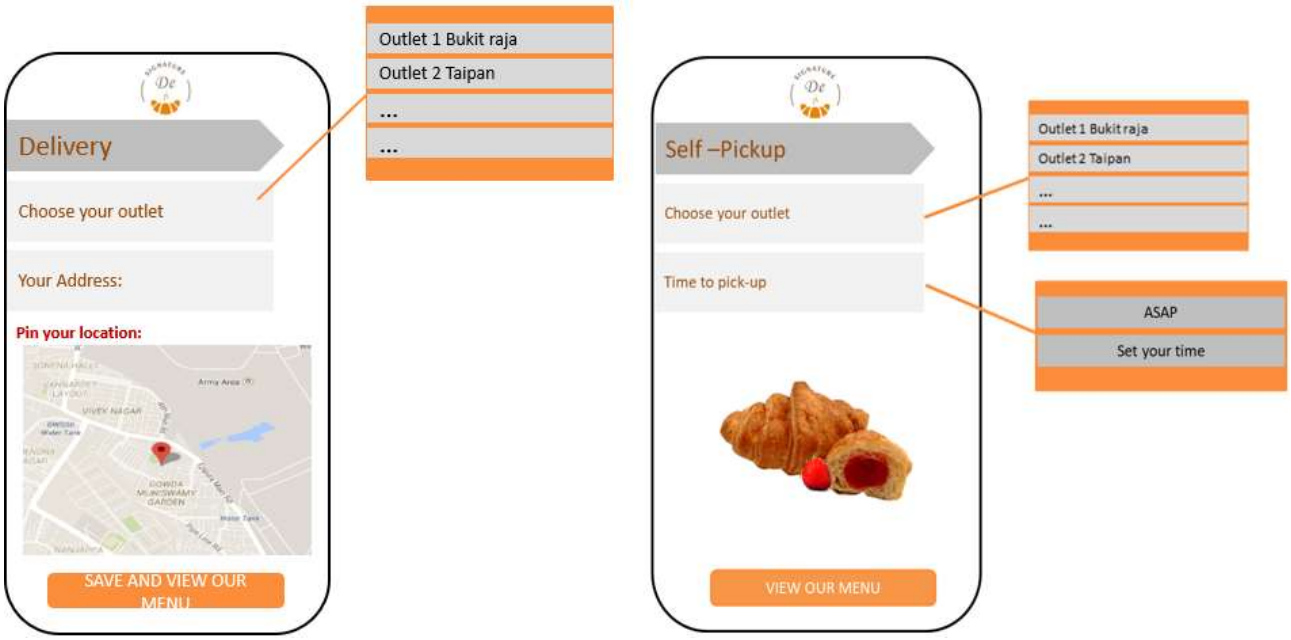
In Figure 7, the new user needs to sign up to create the BOM user's account prior to using the BOM apps. Then user might choose the mode of getting the items through delivery or self-pickup.




(a) (b)

**Fig. 7.** (a) The BOM New User Sign-up and (b) The BOM Menu displayed to new users

Figure 8 comprehends the details information (the outlet and time) needs to be entered either for delivery or self-pickup.




(a) (b)

**Fig. 8.** The BOM mode of getting the items (a) The BOM delivery selection (b) The BOM self-pickup selection

Figure 9 reveals the complete set of items categories (Croissants, drinks or sets) integrated into the Bread on Mobile apps.

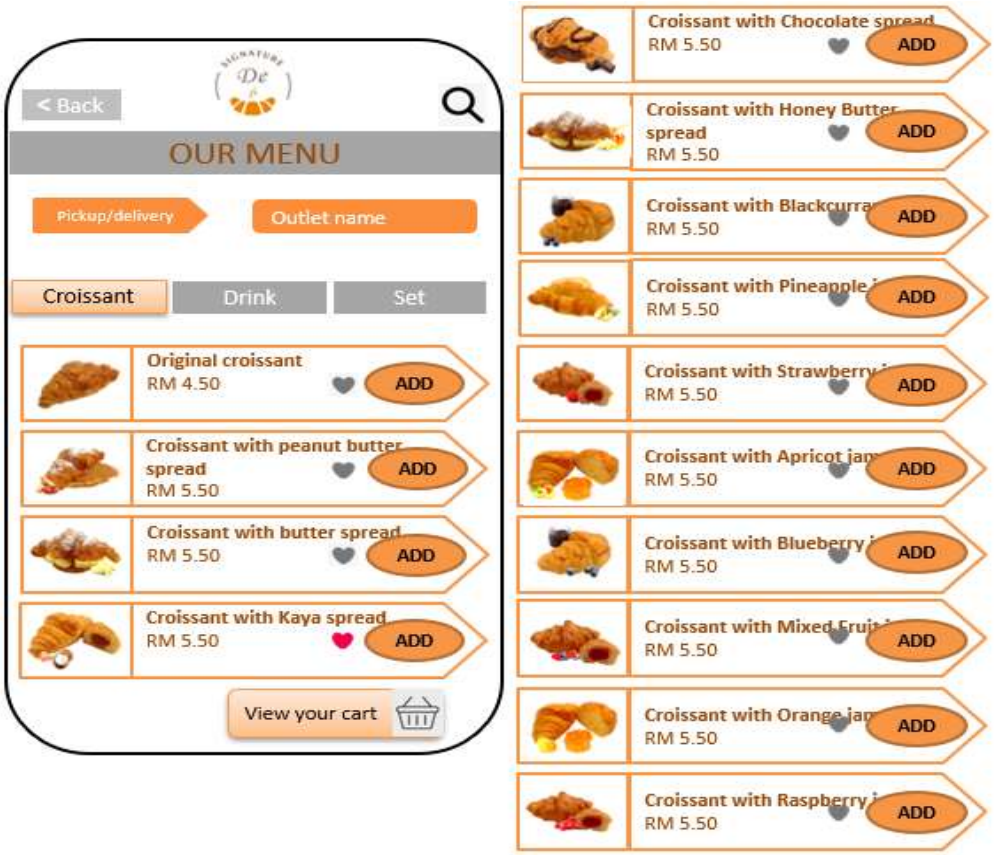



**Fig. 9.** The BOM Menu on several Croissant selections

The selected items shall be added in the cart as shown in Figure 10 whereas the receipt will be issued as illustrated in Figure 11.

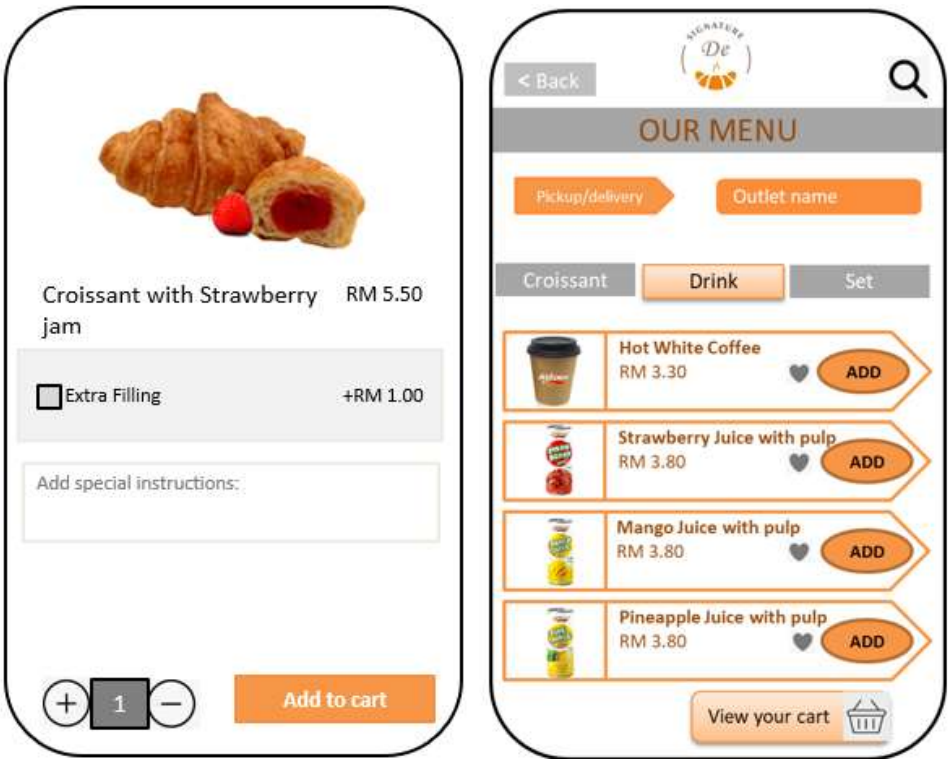



**Fig. 10.** The BOM add to cart for croissant

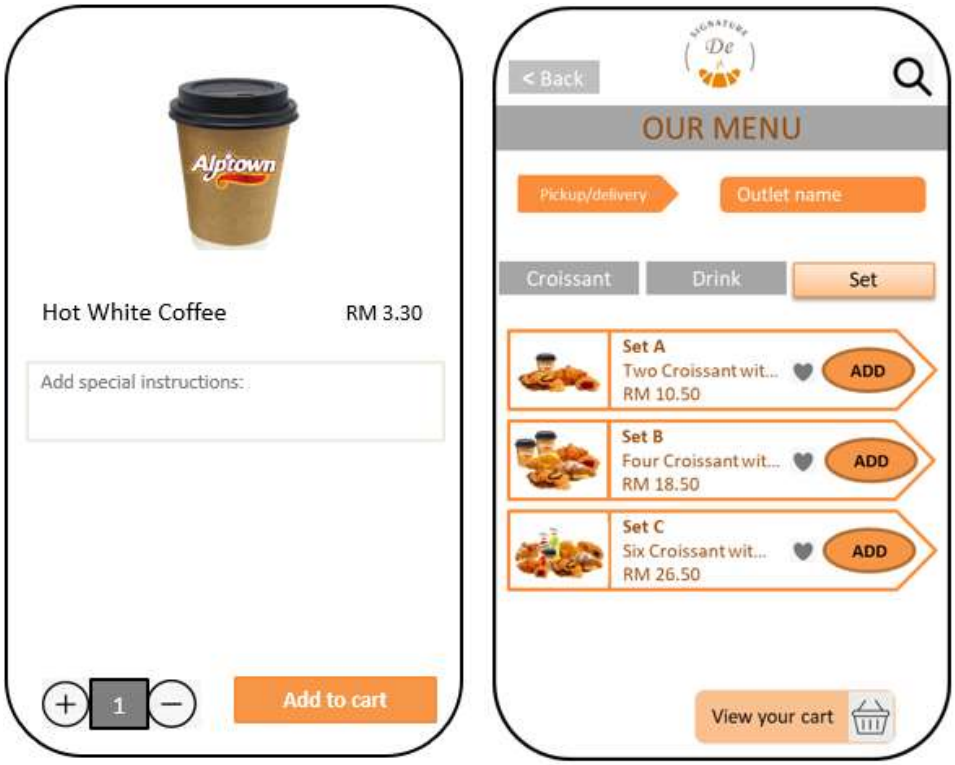



**Fig. 11.** The BOM add to cart for drinks


**Fig. 12.** The BOM receipt issued

The basic algorithm for multiple dynamic flip is given in Figure 13 by which the Dart package is imported before the scripting is implemented using Flutter open-source framework.

**Basic algorithm for Dynamic**
**For backend (Logic and State Management)**
**State Initialization**:
Create a StatefulWidget to manage the state of the animation.
Inside the State class, initialize the AnimationController and Animation<double>.
**Animation Setup**:
Configure the AnimationController with a duration and vsync.
Use a Tween<double> to animate the flip angle from 0 to 1.
Bind the Animation<double> to the AnimationController.
**Trigger Flip Logic**:
Implement a method to toggle the flip state (e.g., _toggleCard()).
Use this method to start the animation by calling forward() or reverse() on the AnimationController depending on the current state.
**For Frontend (UI Design)**
**UI Structure**:
Design the front and back sides of the card as separate widgets.
Use Container widgets with BoxDecoration for styling.
**Apply Animation**:
Wrap the card UI in an AnimatedBuilder to rebuild the UI as the animation progresses.
Use the Transform widget to rotate the card along the Y-axis based on the animation value.
**User Interaction**:
Wrap the card in a GestureDetector to detect tap gestures.
Link the tap gesture to the _toggleCard() method to trigger the flip.

**Fig. 13.** The sample algorithm used in BOM

## 4. Discussion

The BOM mobile app has undergone successful testing and deployment on the BOM platform. The BOM case study serves as a clear indicator that the innovative Multiple Dynamic Flip Algorithm (MDFA) significantly enhances the landscape of mobile-based app development by establishing itself as a standard framework for the development of any mobile application. This achievement has the potential to address prevalent challenges faced by mobile app developers, including the pursuit of user satisfaction and preferences [8], as well as the mitigation of privacy information leaks often associated with mobile apps [9,10]. The MDFA framework simplifies the dynamic development of mobile apps, fostering a smoother interaction between developers and users. When mobile developers adhere to this standardized framework, the process of extracting valuable information for specific needs becomes more convenient [11,12].

## 5. Conclusion

The proposed algorithm, Multiple Dynamic Flip Algorithm (MDFA) seems to project a significant contribution to both entity such as the developer and the end-user in a mobile application development. The implementation of a Bread on Mobile (BOM) case study has achieved the main objectives related to the fast mobile apps development provided with the *ready-to-use* template for any domain application to attract users (industry) in product marketing or end-users (community) in product usage. The persistent issues in mobile apps development have been mitigated by the proposed algorithm in Fragmentation addressed by Yan *et al.,* [16] and Lecomte *et al.,* [17], platform compatibility and code simplification raised by Alshayban *et al.,* [18], performance optimization [19], user-centric design, and easy of testing and debugging [20]. Fragmentation relates to the responsive design principles to ensure that the app adapts gracefully to different screen sizes and resolutions from the flip format of MDFA. The platform compatibility of MDFA referring to the cross-platform development using cross-platform development frameworks like React Native, Flutter, or Xamarin to write code once and deploy it across multiple platforms. The code optimization by MDFA is achieved through the minimizing resource usage, reducing processing overhead, and avoiding memory leaks. MDFA serves for user-centric design where the user can easily understand of the mobile apps' contents through the easy flipping of the UI/UX design accordingly. The easily accessibility features such as voice commands, screen readers, and text-to-speech functionality to make the app accessible to users with disabilities. MDFA offers for easy and automated testing where it adopts automated testing frameworks and tools for functional testing, regression testing, and UI testing to streamline the testing process and identify bugs early in the development cycle. Through the structured database implementation in MDFA, it integrates the automated testing and deployment pipelines into the development workflow to ensure that code changes are thoroughly tested and deployed to production environments in a timely manner [21-23]. In our future endeavours, we might approach the mobile apps development agencies for the commercial potential to benefit both apps developer as well as apps end users in the digital world. Furthermore, we aim to explore the development of mobile apps associated with rapidly changing content.

**Acknowledgement**
This research received funding through the Public Private Research Grant Scheme (UMT- Upaya Vista Sdn. Bhd. - Matching Grant-Vot no: 55375). We extend our heartfelt appreciation to the following individuals for their invaluable contributions: Assoc. Prof. Ts. Dr. Mustafa Man, who played a pivotal role in idea generation, system design, and establishing network connections. Dr. Wan Aezwani Wan

Abu Bakar, who documents the details and editor of the article, and Dr. Mohd. Kamir Yusof, the system developer who significantly contributed to the project. We would also like to express our gratitude to our collaborators from UniSZA and UMT, as well as our industry collaborator, Mr. Terry Yeow from Upaya Vista Sdn. Bhd. Their funding, moral support, proof-checking efforts, and insightful comments and suggestions have been instrumental in the success of this project.

## References

[1] Alotaibi, Ali S., Paul T. Chiou, and William GJ Halfond. "Automated repair of size-based inaccessibility issues in mobile applications." In *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 730-742. IEEE, 2021. https://doi.org/10.1109/ASE51524.2021.9678625

[2] Biduski, Daiana, Ericles Andrei Bellei, João Pedro Mazuco Rodriguez, Luciana Aparecida Martinez Zaina, and Ana Carolina Bertoletti De Marchi. "Assessing long-term user experience on a mobile health application through an in-app embedded conversation-based questionnaire." *Computers in Human Behavior* 104 (2020): 106169. https://doi.org/10.1016/j.chb.2019.106169

[3] Razak, Nurshuhaida Abdul, Hidayahtul Khusna Abdul Malik, Fadilla Rentah, and Mohd Nor Hisham Mazlan. "A comprehensive study of privacy and security risk awareness among mobile internet users for social networks sites in Malaysia." *International Journal of Business and Technology Management* 3, no. 1 (2021): 1-20.

[4] Okuboyejo, Senanu, and Ooreofeoluwa Koyejo. "Examining Users' Concerns while Using Mobile Learning Apps." *International Journal of Interactive Mobile Technologies* 15, no. 15 (2021). https://doi.org/10.3991/ijim.v15i15.22345

[5] Aljedaani, Bakheet, Aakash Ahmad, Mansooreh Zahedi, and M. Ali Babar. "An empirical study on developing secure mobile health apps: The developers' perspective." In *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*, pp. 208-217. IEEE, 2020. https://doi.org/10.1109/APSEC51365.2020.00029

[6] Pandey, Mamta, Ratnesh Litoriya, and Prateek Pandey. "Perception-based classification of mobile apps: A critical review." *Smart computational strategies: Theoretical and practical aspects* (2019): 121-133. https://doi.org/10.1007/978-981-13-6295-8_11

[7] Al-Amri, Redhwan, Raja Kumar Murugesan, Mustafa Man, Alaa Fareed Abdulateef, Mohammed A. Al-Sharafi, and Ammar Ahmed Alkahtani. "A review of machine learning and deep learning techniques for anomaly detection in IoT data." *Applied Sciences* 11, no. 12 (2021): 5320. https://doi.org/10.3390/app11125320

[8] Sabri, Ily Amalina Ahmad, and Mustafa Man. "Improving performance of DOM in semi-structured data extraction using WEIDJ model." *Indonesian Journal of Electrical Engineering and Computer Science* 9, no. 3 (2018): 752-763. https://doi.org/10.11591/ijeecs.v9.i3.pp752-763

[9] Papadopoulos, Elias P., Michalis Diamantaris, Panagiotis Papadopoulos, Thanasis Petsas, Sotiris Ioannidis, and Evangelos P. Markatos. "The long-standing privacy debate: Mobile websites vs mobile apps." In *Proceedings of the 26th International Conference on World Wide Web*, pp. 153-162. 2017. https://doi.org/10.1145/3038912.3052691

[10] Montiel, Ivan, Javier Delgado-Ceballos, Natalia Ortiz-de-Mandojana, and Raquel Antolin-Lopez. "New ways of teaching: using technology and mobile apps to educate on societal grand challenges." *Journal of business ethics* 161 (2020): 243-251. https://doi.org/10.1007/s10551-019-04184-x

[11] Zohud, Tasnim, and Samer Zein. "Cross-platform mobile app development in industry: A multiple case-study." *International Journal of Computing* 20, no. 1 (2021): 46-54. https://doi.org/10.47839/ijc.20.1.2091

[12] Shah, Kewal, Harsh Sinha, and Payal Mishra. "Analysis of cross-platform mobile app development tools." In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, pp. 1-7. IEEE, 2019. https://doi.org/10.1109/I2CT45611.2019.9033872

[13] Khalajzadeh, Hourieh, Mojtaba Shahin, Humphrey O. Obie, Pragya Agrawal, and John Grundy. "Supporting developers in addressing human-centric issues in mobile apps." *IEEE Transactions on Software Engineering* 49, no. 4 (2022): 2149-2168. https://doi.org/10.1109/TSE.2022.3212329

[14] Sarin, Pooja, Arpan Kumar Kar, and Vigneswara P. Ilavarasan. "Exploring engagement among mobile app developers–Insights from mining big data in user generated content." *Journal of Advances in Management Research* 18, no. 4 (2021): 585-608. https://doi.org/10.1108/JAMR-06-2020-0128

[15] Aljedaani, Wajdi, Meiyappan Nagappan, Bram Adams, and Michael Godfrey. "A comparison of bugs across the ios and android platforms of two open source cross platform browser apps." In *2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, pp. 76-86. IEEE, 2019. https://doi.org/10.1109/MOBILESoft.2019.00021

[16] Yan, Shunguo, and P. G. Ramachandran. "The current status of accessibility in mobile apps." *ACM Transactions on Accessible Computing (TACCESS)* 12, no. 1 (2019): 1-31. https://doi.org/10.1145/3300176

[17] Lecomte, Tania, Stéphane Potvin, Marc Corbière, Stéphane Guay, Crystal Samson, Briana Cloutier, Audrey Francoeur, Antoine Pennou, and Yasser Khazaal. "Mobile apps for mental health issues: meta-review of meta-analyses." *JMIR mHealth and uHealth* 8, no. 5 (2020): e17458. https://doi.org/10.2196/17458

[18] Alshayban, Abdulaziz, Iftekhar Ahmed, and Sam Malek. "Accessibility issues in android apps: state of affairs, sentiments, and ways forward." In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pp. 1323-1334. 2020. https://doi.org/10.1145/3377811.3380392

[19] Khalid, Hammad, Emad Shihab, Meiyappan Nagappan, and Ahmed E. Hassan. "What do mobile app users complain about?." *IEEE software* 32, no. 3 (2014): 70-77. https://doi.org/10.1109/MS.2014.50

[20] Hu, Han, Yujin Huang, Qiuyuan Chen, Terry Yue Zhuo, and Chunyang Chen. "A first look at on-device models in ios apps." *ACM Transactions on Software Engineering and Methodology* 33, no. 1 (2023): 1-30. https://doi.org/10.1145/3617177

[21] Shatnawi, Hashem, and N. Alqahtani Mohammad. "Delving into the Revolutionary Impact of Artificial Intelligence on Mechanical Systems: A Review." *Semarak International Journal of Machine Learning* 1, no. 1 (2024). https://doi.org/10.37934/sijml.1.1.3140

[22] Yahya, Abdulsamad E., Atef Gharbi, Wael MS Yafooz, and Arafat Al-Dhaqm. "A novel hybrid deep learning model for detecting and classifying non-functional requirements of mobile apps issues." *Electronics* 12, no. 5 (2023): 1258. https://doi.org/10.3390/electronics12051258

[23] Wang, Hui Hui, Yin Chai Wang, Bui Lin Wee, Xian Rong Chin, and Mengyun Hu. "Smart Shopping Cart Calculator using Barcode Scanner for Supermarkets in Kuching, Sarawak." *Journal of Advanced Research in Applied Sciences and Engineering Technology 49*, no. 1 (2025): 117-128. https://doi.org/10.37934/araset.49.1.117128