Applied Information Technology And Computer Science Vol. 3 No. 2 (2022) 1130-1143
© Universiti Tun Hussein Onn Malaysia Publisher's Office

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

MyFREELANCER App

Muhammad Izzul Hanip[1], **Norhamreeza Abdul Hamid**[1]*

[1]Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

DOI: https://doi.org/10.30880/aitcs.2022.03.02.068
Received 10 August 2022; Accepted 25 October 2022; Available online 30 November 2022

Abstract: MyFREELANCER is an Android app which act as an online freelance marketplace that is based in Malaysia. Nowadays, freelancer plays a big role in Malaysia. Lack of local online freelance marketplace makes it difficult for freelancer in Malaysia to market themselves and finding jobs. Also, by using social media, it is not enough for freelancer to market themselves. Therefore, Malaysia-based online freelance marketplace Android app, MyFREELANCER, is developed for Malaysian freelancer to market themselves efficiently and for customer to find Malaysian freelancer with ease. This app is only applicable for freelancer and customer in Malaysia. Prototype model was used as the methodology to develop this project. To make the graphical user interface look slick and modern, flutter framework is used. Besides that, this project was developed by using Microsoft Visual Studio Code as integrated development environment (IDE) along with Firebase as its cloud firestore. Through this app, it will make the process of freelance assigning activity simpler and easier. It is also will increase the possibilities of freelancer getting hired as the competition is reduced from global to local.

Keywords: Freelance, online freelance marketplace, Android app, Malaysia-based.

1. Introduction

A self-worker, also known as a freelancer, is a worker who holds the type of occupations designated as "self-employment jobs," i.e., jobs where the remuneration is directly reliant on the profits gained from the goods and services provided, on their own or with one or more partners or in a cooperative [1]. It is important for any freelancers to strategically market themselves so that they can increase the possibilities of getting hired. Before this, most freelancers in Malaysia usually will promote themselves on their own personal social media like Facebook, Twitter, WhatsApp or even TikTok. Some of them even use social media as their only platform for freelance activity. For most customer in Malaysia, if they need a freelancer, they usually will just only contact people they know or they will find it through social media.

Due to the use of social media as a sole source of freelance activity in Malaysia, freelancers and customers faced difficulties in finding jobs or freelancers efficiently. The issue with the social media is it is belonging to public, which means all kinds of people are in there. Most of the followers who read the job or freelance marketing post might not be interested in it because they are not the targeted

*Corresponding author: hamreeza@uthm.edu.my audience. For customers, they only can post the job advertisement on their social media and hope that any freelancer would approach them. On top of that, the social media is obviously not enough for both freelancers and customers as they both need more platforms to succeed. Moreover, with all existing global online freelancing marketplace and lack of Malaysia-based online freelancing marketplace, freelancers in Malaysia have difficulties in finding opportunities to get hire as the scale of users and community on such systems or apps are too large. In addition, freelancers and even customers must compete with large scale communities as those systems are applicable worldwide.

Therefore, a Malaysia-based Android app is developed to provide a platform for freelancers in Malaysia to increase their opportunities of getting hired and for customers to find freelancers for their jobs. This app will help improving some of the problems and weaknesses in the current process. The objective of this project is to design a Malaysia-based freelancing mobile app, to develop a user-friendly Malaysia-based freelancing mobile app using object-oriented approach, and to test the functions of Malaysia-based freelancing mobile app. This app is only applicable for Malaysian freelancer and customer only. Towards the end of this project, several modules for users to use including Login, Registration, Profile, Main, Browse, Job Request, Job List and Job History is developed.

2. Related Work

2.1 Online Freelance Marketplace

In recent years, freelancing activity in Malaysia has thrived as work from home environment started on Movement Control Order (MCO). People who were affected by unemployment or were terminated from their previous work also started doing freelance work to gain income. Since then, online freelancing marketplace has been famous among freelancers in Malaysia. An online freelancing marketplace simply connects freelancers and customers from all around the world for a commercial partnership. It will not only assist individuals in finding employment that are suitable with their talents and skills, but also assist businesses in identifying employees or customers who are a suitable for their company and position [2]. Since the late 1990s, dedicated online markets have provided electronic freelancers and their clients with complete technological infrastructure and institutional support [3]. Furthermore, to offer job advertisement, profiles as well as other hiring information, these sites enabled real work to take place online, eliminating reliance on physical location and opening a larger range of talent. Abdul Ghafoor and Dr. N. Fatima Tabassum investigate various online freelance marketplace websites that offer these services to freelancers, including oDesk.com, Guru.com, Freelancer.com, and Elance.com [4]. However, Malaysian online freelance marketplace has not been mentioned. Until now, only a very few apps or systems available specifically for Malaysian freelance.

2.2 Comparison with the Existing Related Systems

Three existing related systems are studied and compared with the features and functions of MyFREELANCER App. This also includes some comparison of the modules. The aim of the study is to develop an application that can help improve the process of freelance activity in Malaysia. Table 1 and table 2 shows the summarization of the comparison results. Table 1 shows the criteria comparison while table 2 shows comparison of the system's modules.

Table 1: System's Criteria Comparison

Criteria	System			
	Fiverr	Upwork	Freelancer.com	MyFREELANCER
System type	Web-based, Mobile App	Web-based, two different mobile apps for	Web-based, Mobile App	Mobile App

		freelancers and customers		
Nation availability	Worldwide	Worldwide	Worldwide	Malaysia
Users	Freelancer, Customer	Freelancer, Customer	Freelancer, Customer	Freelancer, Customer
Accessibility	Must pay subscription to use more extra features	Must pay subscription to use more extra features	Must pay subscription to use more extra features	Do not have to pay anything to use features.
Suitability	For customer who want quick digital services from freelancers	For customer who want experienced person for their professional job or contract	For freelancer who want flexibility in finding jobs	For Malaysian freelancers and customers who do not want to compete with all global competitors.
Speciality	Provide subscription packages for freelancers to set on their profile	Provide various range of specializations in each category	Provide job bidding features for customer to compete for jobs and even duration of the job	Provide easy access for Malaysian users to use.

Table 2: System's Modules Comparison

Modules	System			
	Fiverr	Upwork	Freelancer.com	MyFREELANCER
Registration and login	Available	Need approval for signing in	Need approval for signing in	Available
Profile	Available. (Have level system for freelancer)	Available	Available	Available
Browse	Available	Available	Available	Available
Job Request	Available	Available	Available	Available
Job List	Available	Available	Available	Available
Job History	Available	Available	Available	Available

3. Methodology/Framework

Methodology is divided into a few general phases including Planning, Analysis, Design, Development, Testing, Installation and Maintenance [5]. Prototype model is the model that is used in this project as the methodology. Prototyping model is a systems development approach in which a prototype is constructed, tested, and then revised until an acceptable output is obtained from which the entire system or product may be developed. This project pretty much needs a lot of feedback and requirements from various users. That leads to all requirements that may not be known in advance.

Consequently, the prototype model is suitable for this project as many studies have cited the prototype model as an effective method of gathering input from end users [6].

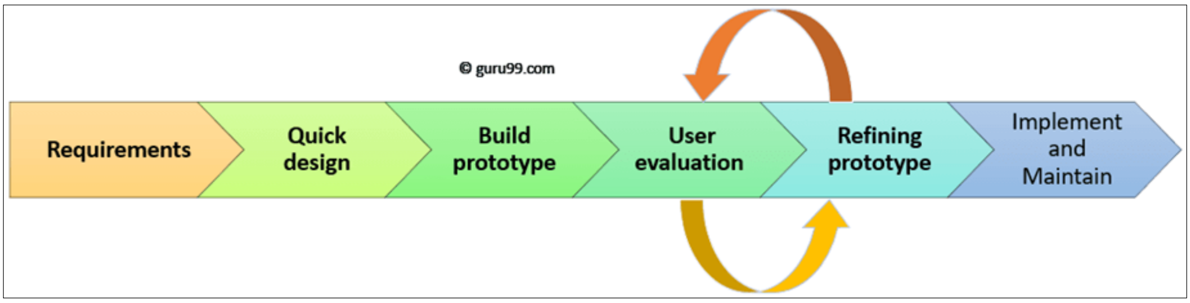


Figure 1: Prototype Model

Table 3: Project development activities and their task

Phase	Task	Output
Requirement Gathering and Analysis	● Analyze the existing systems ● Determine the users' requirement, demand, and expectation from the interview and survey. ● Determine functional and non-functional requirements.	● Table comparison on existing related systems with proposed systems. ● Decided user requirements ● Decided function and non-functional requirements.
Quick Design	● Design the UML. ● Design the Use Case DIagram. ● Design the Sequence Diagram. ● Design the Activity Diagram. ● Design the Class Diagram. ● Design the flowchart ● Design the database. ● Design the GUI.	● The designed UML. ● The designed Use Case Diagram. ● The designed Sequence Diagram. ● The designed Activity Diagram. ● The designed Class Diagram. ● The designed flowchart. ● The designed database. ● The designed GUI.
Build Prototype	● Build an initial app prototype based on quick design.	● Initial app prototype that is ready to be presented to the clients.
User Evaluation	● Present the app prototype to the clients	● Users' feedbacks and suggestions
Refining Prototype	● Adjust the existing prototype based on users' feedbacks and suggestions ● Refine and present the prototype unti; the users are satisfied.	● Refined app prototype. ● Final approved prototype. ● Final app built based on the final approved prototype

Phase	Task	Output
	● Develop the finalized app based on the approved prototype	
Implementation and Mantain	● Implement the app to the production ● Maintain the app to prevent failure.	● Ready to use app.

There are a total of six phases from the prototype model. As shown in Table 2, each phase has its own assignment and output that need to be produced during the entire project development. Besides that, the output had been completed within the specific days that have been given.

3.1 Analysis and Design

3.1.1 Requirement Analysis

Requirement analysis is a process to determine user expectations outcome from the proposed system. There are two types of requirements: business (functional) requirements, and technical (non-functional) requirements [7]. What the app system should do or task that must be accomplished is defined as the functional requirements while the criteria that can be used to judge the app system operation and specific behaviours is defined as the non-functional requirements. Table 4 show the functional requirements while table 5 shows the non-functional requirements of MyFREELANCER app.

Table 4: MyFREELANCER app functional requirements

No.	Module	Functionalities
1.	Login	● The app should allow users to login into their account by inserting their email and password. ● The app should redirect users to respective main dashboard when login successful.
2.	Registration	● The app should allow new users to register an account. ● The app should not allow users to register if the required field is not inserted or invalid. ● The app should redirect to login page if the registration is successful.
3.	Profile	● The app should allow users to update and view the information on their Profile. ● The app should allow users to view each other profile.
4.	Browse	● The app should allow users to browse/search any freelancers based on categories.
5.	Job Request	● The app should allow customers to make a request to the freelancers for their job. ● The app should allow freelancers to choose either to accept or reject the request of the job.
6.	Job List	● The app should list all the confirmed jobs. ● The app should allow customer to set the job as done.
7.	Job History	● The app should list all past job that has been done.

Table 5: MyFREELANCER app non-functional requirements

No.	Requirements	Descriptions
1.	Performance	● The interaction between the users and the app system must be smooth and clean. ● The app system should be able to be used anytime. ● The app system should not oftenly crashed.
2.	Operational	● The app should be user-friendly to users. ● The app should work with internet services.
3.	Security	● The app should only allow registered user to have access to the app through login with their correct username and password.

3.1.2 System Analysis

This app system uses object-oriented approach as the analysis and design. Thus, Unified Modelling Language (UML) Diagram is used to illustrate the results of the analysis including the use case diagram, sequence diagram, activity diagram, and class diagram as representations.

The use case diagram was developed as part of the analysis to show the overall function and components of the app system. The use case diagram is described in a form of task description or module that will implement on the app system. Figure 2 shows the use case diagram for MyFREELANCER app. The actors are freelancer and customer.

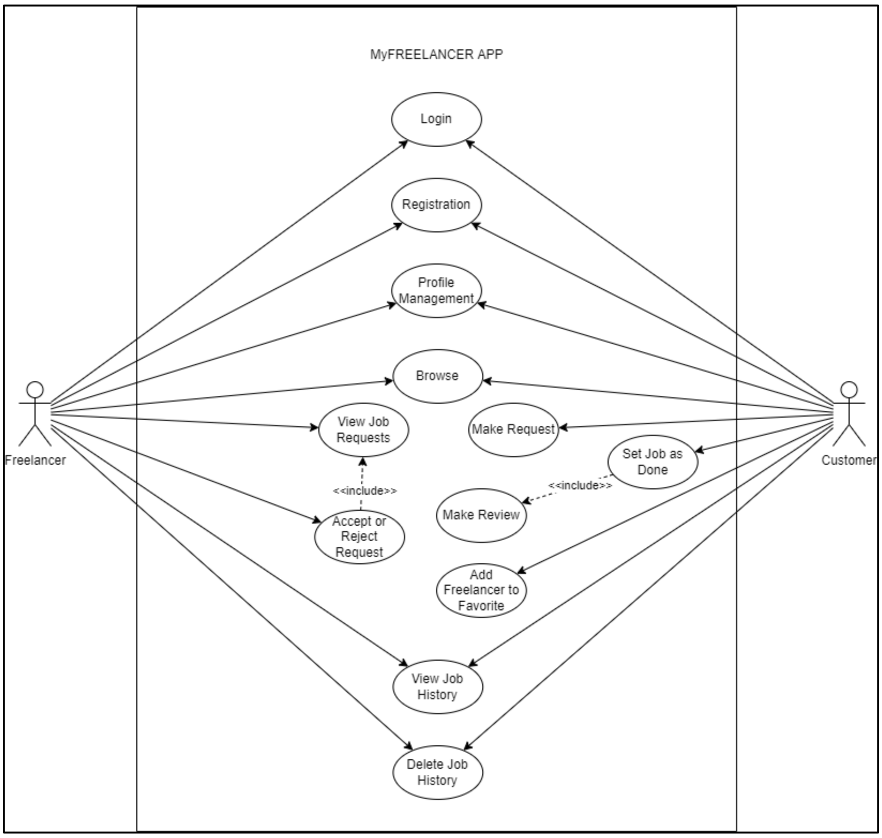


Figure 2: MyFREELANCER Use Case Diagram

To establish the attributes and methods to be used in the app system, class diagrams is produced. The relationship between classes has been established and illustrated using sequential diagrams. In Figure 3, 7 classes are identified in the app system which are User, Customer, Freelancer, Browse, Profile, JobRequest, Job List, and JobHistory.

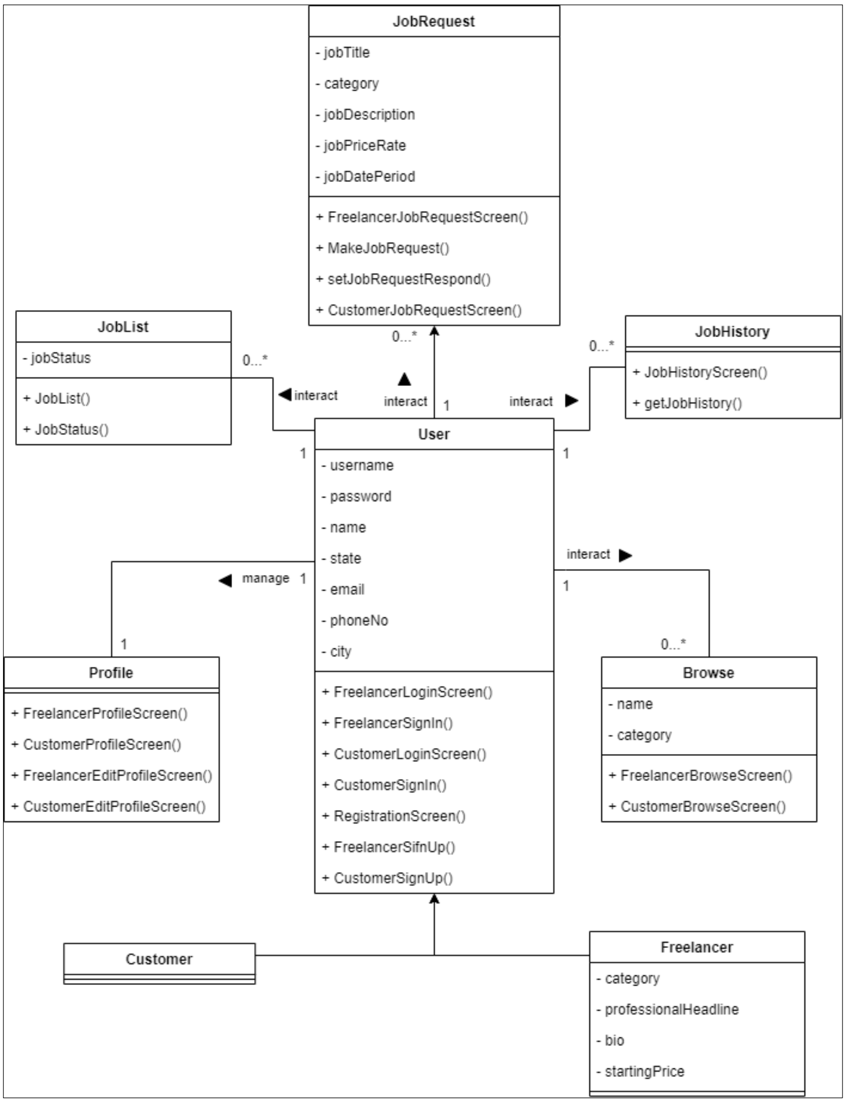


Figure 3: MyFREELANCER Class Diagram

3.2 System Design

After all the users' requirements were successfully analyze, the project will proceed on design phase. In this phase, both interface and database had been designed in order to help visualize the system before proceeding with the coding the system. One of system design are flowchart can be seen at Appendix A.

3.3 Implementation

Through this phase, interfaces, and modules MyFREELANCER app are transformed into programs and linked together in a cloud firestore to fulfil all the functionality set. The interface design is developed based on the system interface sketch performed in the previous phase. MyFREELANCER app is developed by using Microsoft Visual Studio Code along with flutter framework and dart as its programming language. Firebase is used as cloud firestore that connects with the app system.

Figure 3(a), 3(b), and 3(c) shows some of the interfaces of MyFREELANCER app which are Login page, Profile page, and Make Job Request page.

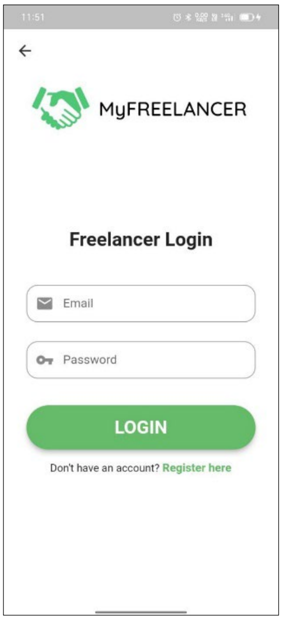
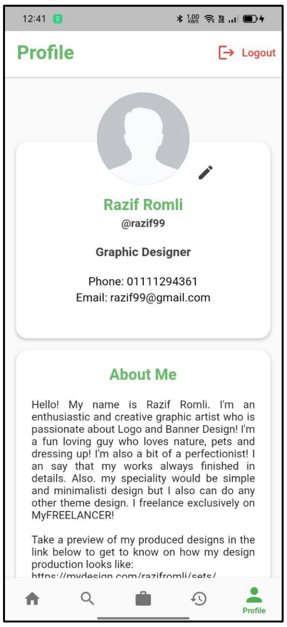


Figure 3(a): Login page



Razif Romli
@razif99

Graphic Designer

Phone: 01111294361
Email: razif99@gmail.com

About Me

Hello! My name is Razif Romli. I'm an enthusiastic and creative graphic artist who is passionate about Logo and Banner Design! I'm a fun loving guy who loves nature, pets and dressing up! I'm also a bit of a perfectionist! I an say that my works always finished in details. Also. my speciality would be simple and minimalisti design but I also can do any other theme design. I freelance exclusively on MyFREELANCER!

Take a preview of my produced designs in the link below to get to know on how my design production looks like:
https://mydesign.com/razifromli/sets/

Figure 3(b): Profile page

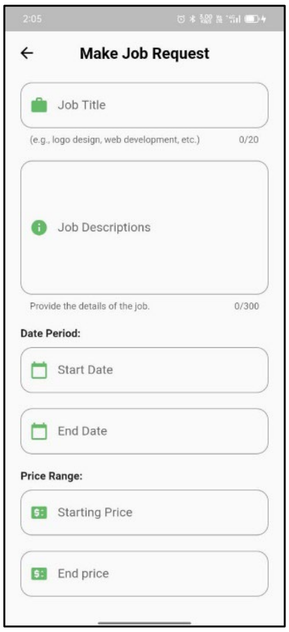
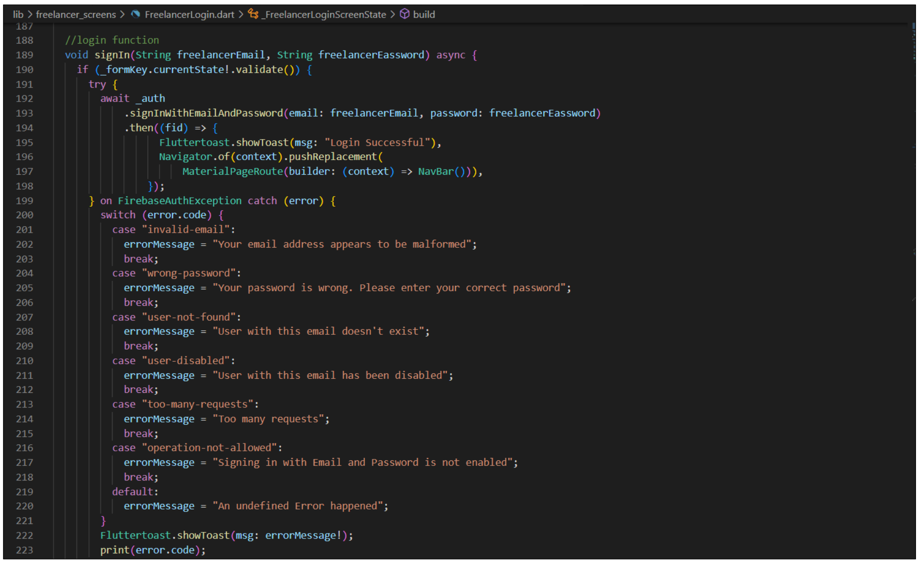


Figure 3(c): Job Request page

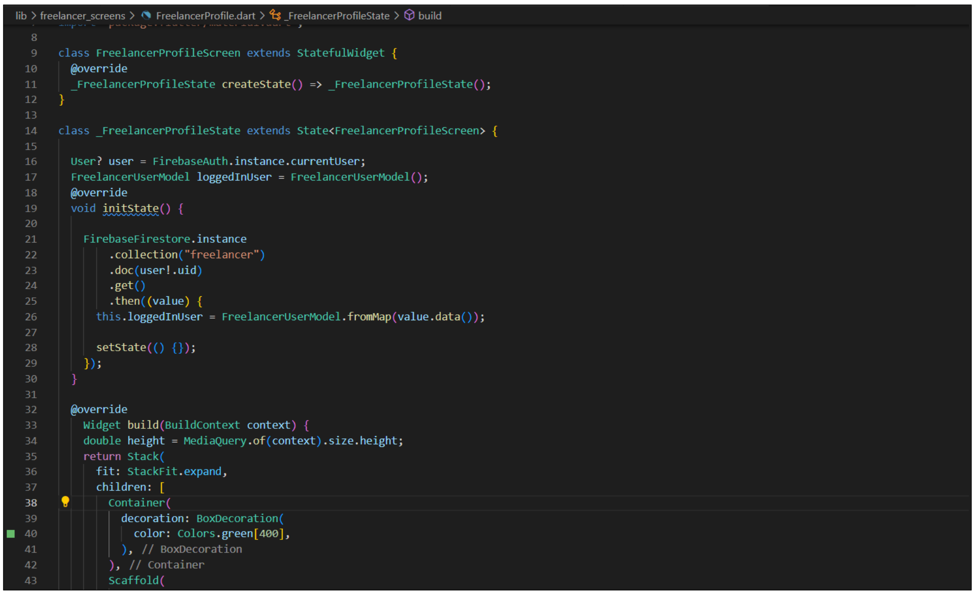
Figure 3(a) shown that users must login to the app by entering both email and password. In figure 3(b), it shows that all the profile details are displayed in Profile page. Moreover, users can even edit the profile by clicking the pencil button beside user's profile picture. Basically, this app main function is Job Request where customer able to make job requesto to selected freelancer as shown as figure 3(c). In order to do so, customer have to visit Browse page and search freelancers whether by using search bar or pick freelancer form selected category. After receiving request from customer, freelancer can choose to accept or reject the request. Then, customer can set the job as done so that the job details would transfer to Job History page.

Figure 4(a), 4(b), and 4(c) shows some of the modules code segment which are Login and Profile code segment. The development of this project used Object-Oriented Programming approach.



```
lib > freelancer_screens > FreelancerLogin.dart > _FreelancerLoginScreenState > build

  //login function
  void signIn(String freelancerEmail, String freelancerEassword) async {
    if (_formKey.currentState!.validate()) {
      try {
        await _auth
            .signInWithEmailAndPassword(email: freelancerEmail, password: freelancerEassword)
            .then((fid) => {
                  Fluttertoast.showToast(msg: "Login Successful"),
                  Navigator.of(context).pushReplacement(
                      MaterialPageRoute(builder: (context) => NavBar())),
                });
      } on FirebaseAuthException catch (error) {
        switch (error.code) {
          case "invalid-email":
            errorMessage = "Your email address appears to be malformed";
            break;
          case "wrong-password":
            errorMessage = "Your password is wrong. Please enter your correct password";
            break;
          case "user-not-found":
            errorMessage = "User with this email doesn't exist";
            break;
          case "user-disabled":
            errorMessage = "User with this email has been disabled";
            break;
          case "too-many-requests":
            errorMessage = "Too many requests";
            break;
          case "operation-not-allowed":
            errorMessage = "Signing in with Email and Password is not enabled";
            break;
          default:
            errorMessage = "An undefined Error happened";
        }
        Fluttertoast.showToast(msg: errorMessage!);
        print(error.code);
```

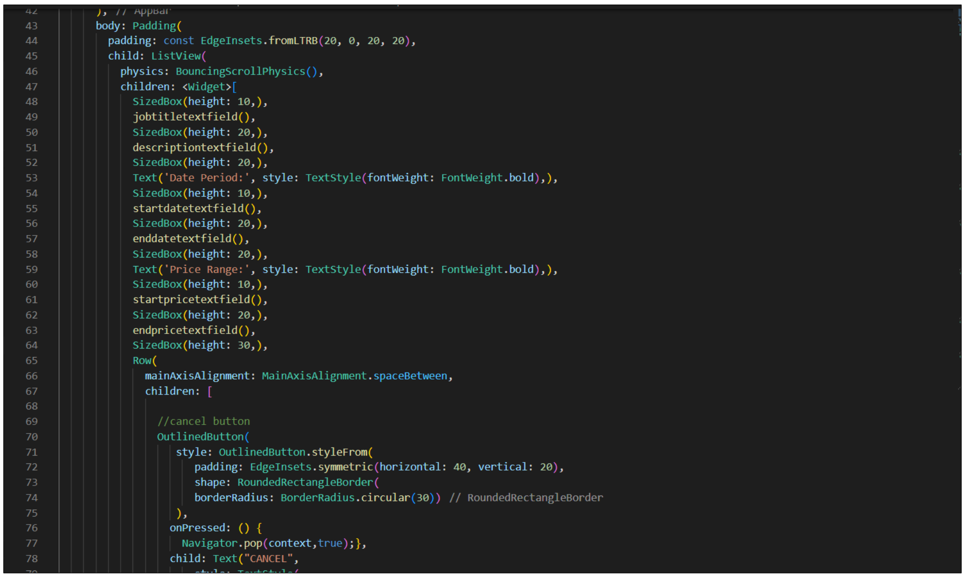
Figure 4(a): Login Code Segment



lib > freelancer_screens > FreelancerProfile.dart > _FreelancerProfileState > build

```dart

class FreelancerProfileScreen extends StatefulWidget {
  @override
  _FreelancerProfileState createState() => _FreelancerProfileState();
}

class _FreelancerProfileState extends State<FreelancerProfileScreen> {

  User? user = FirebaseAuth.instance.currentUser;
  FreelancerUserModel loggedInUser = FreelancerUserModel();
  @override
  void initState() {

    FirebaseFirestore.instance
        .collection("freelancer")
        .doc(user!.uid)
        .get()
        .then((value) {
      this.loggedInUser = FreelancerUserModel.fromMap(value.data());

      setState(() {});
    });
  }

  @override
    Widget build(BuildContext context) {
    double height = MediaQuery.of(context).size.height;
    return Stack(
      fit: StackFit.expand,
      children: [
        Container(
          decoration: BoxDecoration(
            color: Colors.green[400],
          ), // BoxDecoration
        ), // Container
        Scaffold(
```

Figure 4(b): Profile Code Segment



```dart
        ), // AppBar
        body: Padding(
          padding: const EdgeInsets.fromLTRB(20, 0, 20, 20),
          child: ListView(
            physics: BouncingScrollPhysics(),
            children: <Widget>[
              SizedBox(height: 10,),
              jobtitletextfield(),
              SizedBox(height: 20,),
              descriptiontextfield(),
              SizedBox(height: 20,),
              Text('Date Period:', style: TextStyle(fontWeight: FontWeight.bold),),
              SizedBox(height: 10,),
              startdatetextfield(),
              SizedBox(height: 20,),
              enddatetextfield(),
              SizedBox(height: 20,),
              Text('Price Range:', style: TextStyle(fontWeight: FontWeight.bold),),
              SizedBox(height: 10,),
              startpricetextfield(),
              SizedBox(height: 20,),
              endpricetextfield(),
              SizedBox(height: 30,),
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: [

                  //cancel button
                  OutlinedButton(
                     style: OutlinedButton.styleFrom(
                        padding: EdgeInsets.symmetric(horizontal: 40, vertical: 20),
                        shape: RoundedRectangleBorder(
                        borderRadius: BorderRadius.circular(30)) // RoundedRectangleBorder
                    ),
                    onPressed: () {
                      Navigator.pop(context,true);},
                    child: Text("CANCEL",
```

Figure 4(c): Job Request Code Segment

4. Results and Discussion

Testing is an essential component of the project development process. Testing is performed to determine whether there is an error in the flow of the generated project. The app must be tested to ensure that all functions work properly. Typically, testing is done with the users to demonstrate and obtain feedback on the completed project.

4.1 Functional Testing

Functional testing is done on the modules developed. This testing is carried out to ensure that the developed app performs exactly like how the user required. Test cases specify how to put a system, application, or program through its tests. A test case is a unique set of activities or instructions for a tester to follow in order to validate a specific feature of a product's or application's operation.

Table 5: Test case

No.	Test Cases	Expected Outcome	Actual Outcome	Status
LOGIN TEST CASE				
1.	Login the app using registered email and password	Login successful and redirected to home page	As expected	PASS
2.	Login the app using wrong or unregistered email and password	Login failed and it display error message that require users to enter the correct data	As expected	PASS
3.	Login the app without filling any field	Login failed and display error message that require users to fill in the field.	As expected	PASS
REGISTRATION TEST CASE				
1.	Register the app with all required field filled in	Registration successful and redirect to Login page	As expected	PASS
2.	Register the app with incomplete field	Registration failed and display error message that require users to fill in the field	As expected	PASS
3.	Register the app with existed email	Registration failed and display error message stating that the email already exists	As expected	PASS
PROFILE TEST CASE				
1.	Edit profile	Profile successfully edited and will be displayed to profile page	As expected	PASS
2.	Leave profile edit field empty	Show error message stating that all the field must be filled.	As expected	PASS
3.	Click cancel when editing profile	Fail to edit profile and it will be redirected back to the profile page	As expected	PASS
BROWSE TEST CASE				
1.	Select category	List all freelancer who are under selected category	As expected	PASS
2.	Search category, freelancer, or state in search bar	List all the searched freelancers.	As expected	PASS
3.	Click freelancer form the listed freelancers	Show the freelancer’s details	As expected	PASS
JOB REQUEST TEST CASE				
1.	Customers make request by filling all the required field	Job request successfully made, and it will be displayed in Job Request page	As expected	PASS
2.	Customers click button “Make Request” but leave some of required the field empty	Job request failed and error message is displayed stating that field cannot be empty	As expected	PASS
3.	Freelancer accept the request	Job details will be listed in Job List page	As expected	PASS
4.	Freelancer reject the request	Job detail	As expected	PASS
JOB LIST TEST CASE				

No.	Test Cases	Expected Outcome	Actual Outcome	Status
1.	Customers click "SET AS DONE" button	The job details will be transferred to Job History page	As expected	PASS
JOB HISTORY TEST CASE				
1.	Delete the job details	Job successfully deleted	As expected	PASS

4.2 User Acceptance Testing

User Acceptance Testing (UAT) for MyFREELANCER app is carried out by distributing a set of questionnaires to potential users including freelancers and any random people that can be consider as customer. The questionnaires were made by using Google Form, Also, it is attached along with the apk file of MyFREELANCER app so that the respodent can test the app first before giving feedback. A total of 30 respondents are involved during this test.

Table 6: User acceptance testing

No	Requirements	Result
1.	Allow user to login the app only with registered email and password	Success
2.	Allow user to register an account by entering all required field	Success
3.	Allow user to view and update their own profile	Success
4.	Allow user to search freelancers by category or by name	Success
5.	Allow customer user to make job request to freelancer	Success
6.	Allow freelancer user to choose either to accept or reject the job request	Success
7.	List all the confirmed jobs	Success
8.	Allow customer user to set the confirmed job as done	Success
9.	List all past jobs	Success

From table 6, it is shown that in term of functionality, most functions are working well and also met the requirement. From that, it is proven that this app is well accepted by target users and fulfils most of the objectives of the project stated in earlier phase.

5. Conclusion

To conclude, MyFREELANCER app has achieved the objectives based on the system requirements, scope, and user requirements. Although it contains some limitations, the app still has its own charm and advantages. Moreover, the app still has a room for improvement later. Most importantly, the main objective is met, and this project was done successfully.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Appendix A

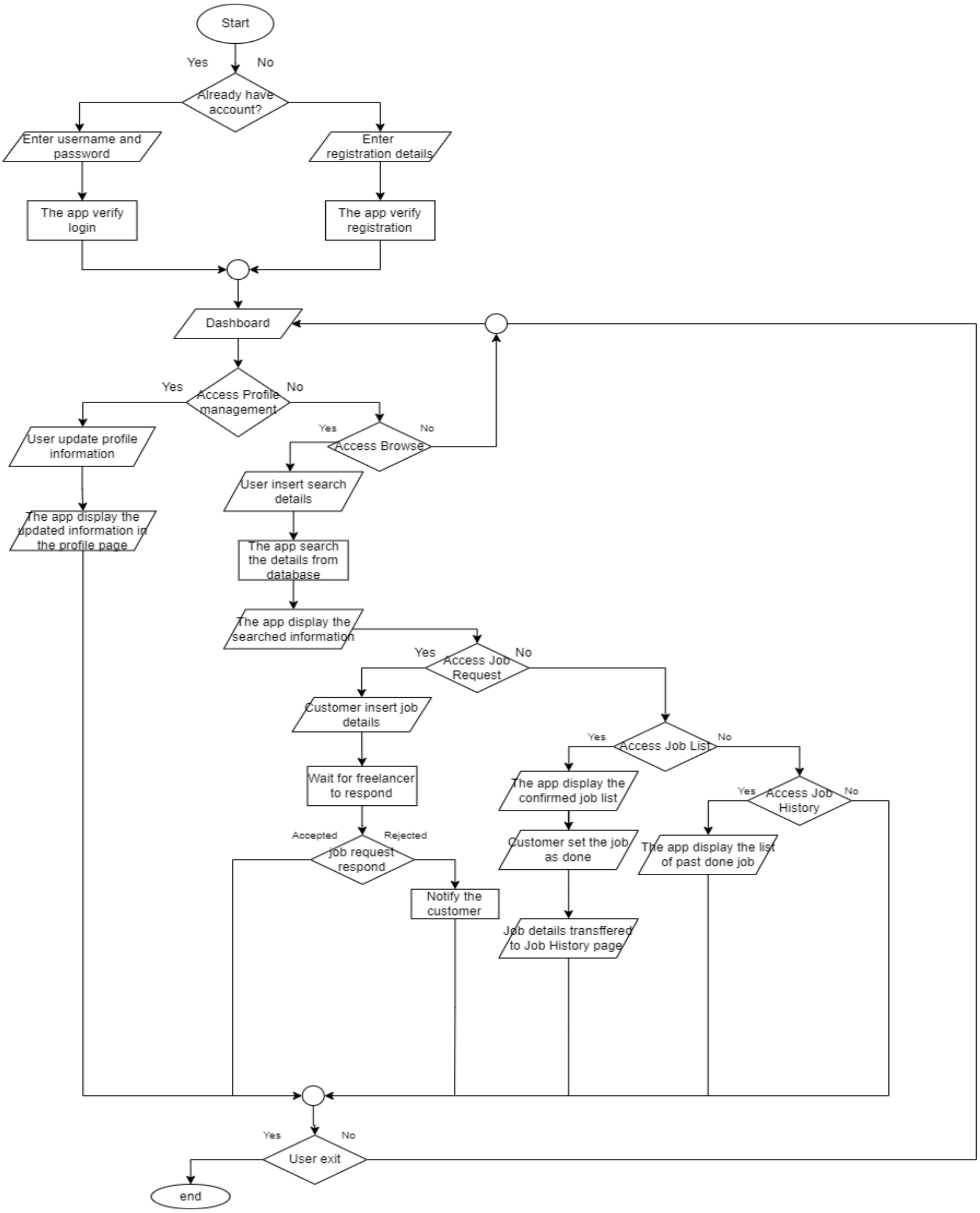

Start
Yes
No
Already have account?
Enter username and password
Enter registration details
The app verify login
The app verify registration
Dashboard
Yes
No
Access Profile management
User update profile information
The app display the updated information in the profile page
Yes
No
Access Browse
User insert search details
The app search the details from database
The app display the searched information
Yes
No
Access Job Request
Customer insert job details
Wait for freelancer to respond
Accepted
Rejected
job request respond
Notify the customer
Yes
No
Access Job List
The app display the confirmed job list
Customer set the job as done
Job details transffered to Job History page
Yes
No
Access Job History
The app display the list of past done job
Yes
No
User exit
end

Figure A: MyFREELANCER flowchart

References

[1] A. G. Kazi, R. M. Yusoff, A. Khan & S. Kazi, "The Freelancer: A Conceptual Review," Sains Humanika, 2(3), 2014.

[2] A. Ramesh, L. Krishnamoorthy, S. Baharudin & K. Karkonasas, "Intention to adaption and influence of using online freelance portal among Malaysians," International Journal of Applied Engineering Research, 11(19), 10020-10026, 2016.

[3] A. Shevchuk & D. Strebkov, "Freelancers in Russia: Remote work patterns and e-markets," Economic Sociology: The European Electronic Newsletter, 13(2), 37-45, 2012

[4] N. T. Fathima, "Burgeoning of Remote Jobs in Developing Countries –Special Reference to Asia Pacific Region," International Journal of Research Publication's, Vol 4, issue no.6, 2014

[5] D. Kumar & D. Pushkar, "Software Development Life Cycle (SDLC) Analytical Comparison and Survey on Traditional and Agile Methodology," Abhinav National Monthly Referred Journal of Research in Science & Technology. 2. 22-30, 2013

[6] L. Mathiassen, T. Seewaldt, and J. Stage, "Prototyping and Specifying: Principles and Practices of a Mixed Approach,". scandinavian journal of information systems (7:1) 1995, pp 55-72. 1995.

[7] Jiao, J. X., and Chen, C. H. (2006). Customer Requirement Management in Product Development. A Review of Research Issues, Vol 14, Issue 3.

[8] A. Susanto, Meiryani, "System Development Method with the Prototype Method," International Journal of Scientific & Technology Research. 8. 07. 141-144, 2019