



Design and Build a Mobile-Based Pet Care Information System with Personal Extreme Programming Method

Ida Ayu Dela Aurellia, Royana Afwani^(✉), Budi Irmawati⁽ⁱ⁾, Moh. Ali Albar, Noor Alamsyah, and I Gde Putu Wirarama WW

Mataram University, Mataram, Indonesia
royana@unram.ac.id

Abstract. Pets are animals that are tamed and cared for by their owners and have an emotional bond between the two that will form a relationship. The high interest in raising animals has led many communities and animal lovers because animal owners have the same interest in caring for and caring for animals. The author has interviewed several animal lovers in the city of Mataram. Five out of seven animal lovers who have left their pets at a pet shop experience the same problem: it is difficult to find a reliable place of care when they want to travel far. It is not possible to bring their pets to that time. Three out of five people are more confident in entrusting their pets to the animal lover community because the care and maintenance carried out by pet shops are often not optimal; The high cost of the pet shop is also an obstacle. Based on these problems, the authors created a “Mobile-Based Animal Care Application”, which is a forum that connects fellow animal lovers who need each other’s help to deposit and care for their pets according to the desired facilities and budget. Work on this system is carried out by the author individually. This author must handle changes in the system’s functionality that will be developed according to what the client wants quickly and precisely. Therefore, this application is made using the PXP method, which makes it easy to handle changes in the system’s functionality to be developed because PXP has a high level of flexibility in planning and implementing an application/software. This application system will use the Java programming language and firebase for data storage in its design. From the test after developing and making the system, the results obtained from black-box testing with the Linkert Scale that all scenarios made are appropriate and can help the needs of the community and clients, the score for strongly agree is 50% from the client-side and 57.1% from the community side.

Keywords: Inormation System · PXP Method · Android Apps · West Nusa Tenggara · Animal Care.nf

1 Introduction

Pets are animals that are tamed and cared for by their owners and have an emotional bond between two that will form a relationship between humans and animals [1]. Having a pet is one of the ways that humans express their love for animals. Many positive things can be obtained from raising animals, including reducing boredom and stress, training to be more responsible, making someone more diligent, feeling at home, strengthening family relationships, and positively influencing health [2]. Based on data obtained through the Rakuten Insight Survey, Indonesia is a country that has quite a lot of pets, with a percentage of only 28% of people who do not have pets and the country with the highest percentage of cat owners in Asia [2]. The high interest in raising animals has created many communities and animal lovers.

Based on interviews that have been conducted with several animal lovers in Mataram City, five out of seven animal lovers who have brought their pets to a pet shop experience the same problem when they want to leave their pets when they want to travel far, and it is not possible to bring their pets at that time. Three of the five people are more confident in entrusting their pets to the animal lover community because the care and maintenance carried out by pet shops for animals are often not optimal. The high cost of the pet shop is also an obstacle. Based on these problems, the authors make “Animal Care Applications with Personal Extreme Programming Methods”.

The information system created will be used by pet owners to find a place of care that meets the criteria for animal owners. This information system works by looking for a place for pet owners to leave their pets. First, the system will display the profiles and facilities of various animal care centers. Pet owners can choose the care they want after browsing the profiles and facilities that have been displayed. After that, the pet owner can send a message to the animal care provider to ask and confirm the pet care.

The pet care center must previously fill in complete personal data or profile and data regarding the facility. The admin will check the data entered through the web admin, which has been integrated with the mobile application. After the admin checks and verifies, the data for the animal care center can be displayed on the application. They can receive animal deposits from pet owners.

2 Literature Review

2.1 Related Research

The creation of this animal care information system refers to research conducted previously. These studies will be use as a reference source in developing this system.

In a previous study released in 2020, there was an application for animal care using the iOS-based payment feature. When the animal is deposit, this application serves to monitor. The advantages of this application are online payments at cooperating pet shops, which are still not widely found. SUS's average score is 87.13%. These results indicate that the application built can be used quickly and well [3].

In 2018, research was released on animal care services that make it easier to manage admin data to minimize errors in making reports on animal care data faster, more precise,

accurate, and relevant. In this study, it was found that the design of animal care applications can be used as an excellent solution to improve the quality-of-service quality [4].

The 2018 study had a reservation application for animal care services. The application allows customers to make reservations and find animal care and care services available. The advantage of making applications is that it can be done in a native way that can use all the features of a smartphone, such as notifications, calendars, and others, and use lighter memory. From this research, it can be concluded that this mobile application can find healthy, comfortable, and safe animal care centers. In addition, pets who are reliable daycare providers can use this application to offer their services and simplify the process of booking a place [5].

Design and Implementation of the Android-Mobile Marketing Surveyor System (A-MS2) in the previous study using the personal extreme programming (PXP) method. The development of the A-MS2 system will apply the personal extreme programming (PXP) model. The PXP process model is considered more able to handle every stage carried out during the A-MS2 system work process [6].

The information system is entitled “Program Keluarga Harapan (PKH),” and the research was developed using extreme programming. This method is applied because it will make it easier to explore the client’s needs and convert them into features that will be applied to the system later. It also makes it easier to follow the changes that the client wants so that the results provided are to the client’s needs [7].

The design and manufacture of “Android-Based Interactive Mobile Learning” in previous studies used the personal extreme programming method. PXP is used because it makes it easy to estimate and estimate the functional priority of the developed system [8].

3 Methodology

See Fig. 1.

3.1 Requirement

After collecting data, a needs analysis was carried out. The needs analysis describes making information systems and analyzing needs using the collected data. Data were obtained from many sources, such as through interviews with several pet shops and animal lovers.

3.2 System Design

After collecting and getting all the requirements needed in system development, the next step is making a system design. At this stage, the system design is using an object-oriented approach, namely Unidentified Modeling Language (UML) and Design Prototype (Figs. 2 and 3).

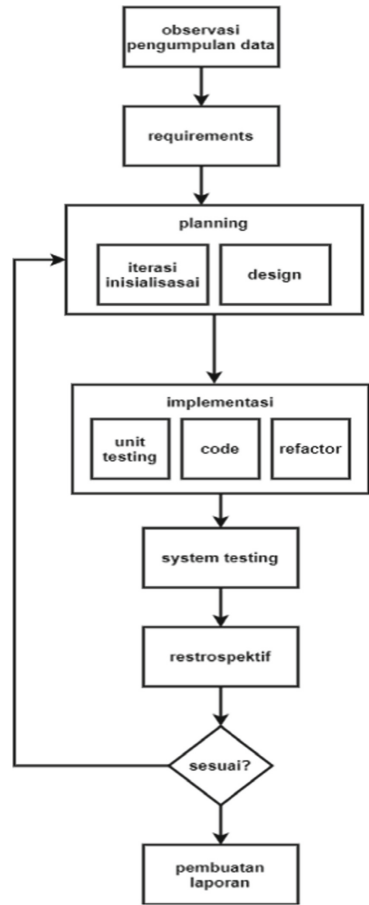


Fig. 1. Research flow chart.

3.3 Product Testing

In this system, a test is carrying using a questionnaire. The questionnaire is a research instrument that consists of a series of questions that aim to collect information from respondents. The questionnaire can be considered a written interview. The assessment of this questionnaire is intended for the custodian and recipient. In this research, we used questioner that utilized the Linkert scale to measure acceptance of the application.

4 Result and Discussion

The chapter in this study will explain the stages carried out in the research process including implementation, planning, and implementation of user stories obtained from the requirements process.

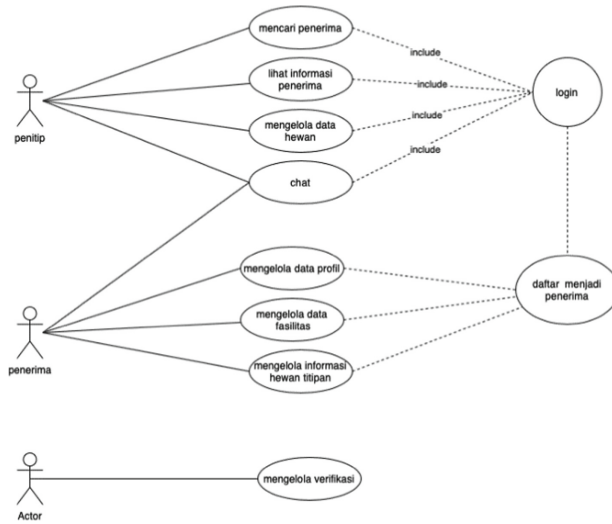


Fig. 2. Use Case Diagram.

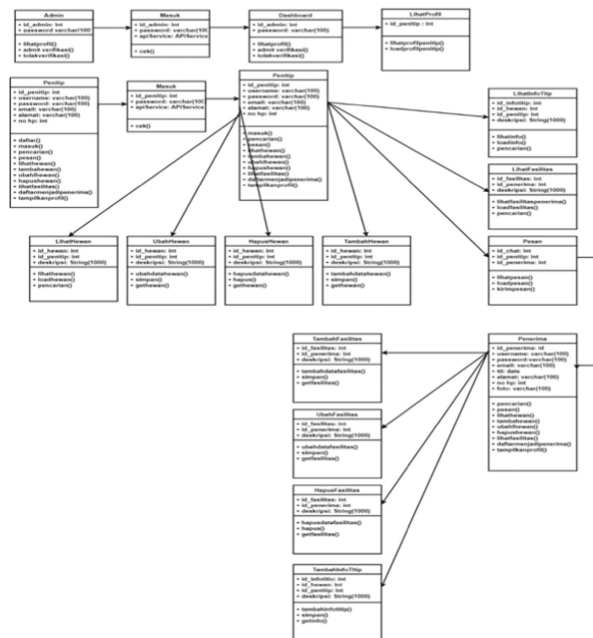


Fig. 3. Class Diagram.

4.1 Estimated User Stories

User stories have three risk factors, and each factor has an index. The developer himself determines the index for each risk factor. The index determination is filled by the knowledge and abilities possessed by the developer. Then the results of determining the index for each risk factor will be discussed with the client so that the client can provide input on the index that the developer has determined.

4.2 Iteration Planning

In this iteration, user stories work on during the process. Determination of user stories carried out in iterations is determined by discussing with the client and based on the priority of the user stories until the velocity value is met (Table 1).

Table 1. Iteration Planning

Iteration 1			
User Code Stories	Priority		Story Points
	Value	Risk	
Stories– 01	Critical	(2) Medium	2
Stories– 02	Critical	(3) Medium	1
Stories– 03	Critical	(3) Medium	1
Velocity			4
Iteration 2			
User Code Stories	Priority		Story Points
	Value	Risk	
Stories– 09	Critical	(2) Medium	2
Stories– 10	Critical	(5) Medium	2
Stories– 07	Critical	(3) Medium	2
Stories– 08	Critical	(3) Medium	2
Velocity			8
Iteration 3			
User Code Stories	Priority		Story Points
	Value	Risk	
Stories- 04	Critical	(3) Medium	2
Stories- 05	Critical	(2) Medium	2
Stories- 06	Critical	(3) Medium	3
Stories– 11	Critical	(3) Medium	1
Velocity			8

Iteration 1. After the planning stage has been carried out, the results of this stage have been found, namely in system development. There is an iteration plan which will be carried out in three iterations. In the first iteration, developers will work on Stories – 01, Stories – 02, and stories – 03. Stories - 01 works that the custodian, recipient, and admin get access rights to the system. Stories - 02 functions as a place for depositors to become recipients. Stories - 03 functions to verify the depositor's data that will be the recipient by the admin.

Design. The design made by the developer is a CRC Card and spike solution prototype. The CRC Card will show each class's tasks and what classes interact with that class. Spike solution prototype is an interface design scheme based on the CRC Card from user stories in the first iteration (Figs. 4, 5 and 6).

Program Code Generation. After completing the CRC prototype, it will continue with the next stage, namely the creation of program code. This phase is divided into three stages: Unit Testing, Code Generation, and Code Refactoring.

- Unit Test

Unit testing is done to test the function of the code to be made. In creating program code for unit testing, the developer first makes unit testing based on the CRC Card that has been made.

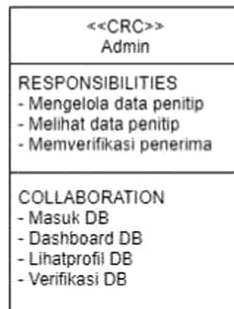


Fig. 4. Admin CRC Card



Fig. 5. Verification CRC Card.

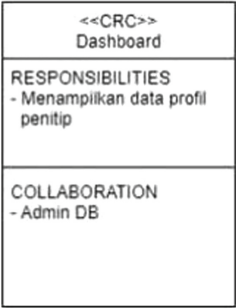


Fig. 6. Dashboard CRC Card.

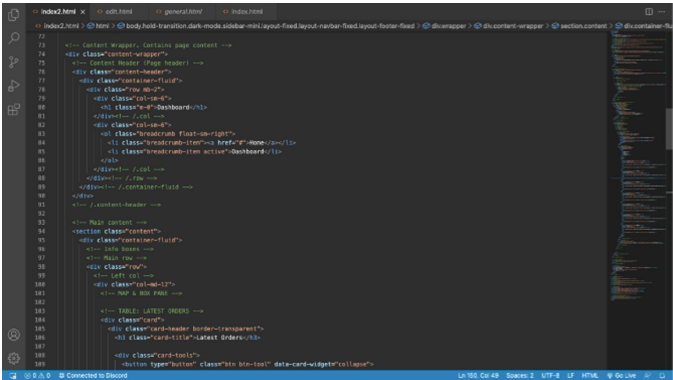


Fig. 7. Iteration – 1 Code Generation.

- Code Generation
After all unit testing has passed the test, the next step is to create program code consisting of the features that exist in each user story from the first iteration. This feature is in the form of a login feature for admin verification, custodians and recipients.
- Code Refactoring
This stage is carried out after the completion of the program code, the developer then maximizes the program code that has been created. In this iteration, the developer does not refactor because the code created does not cause any redundancies (Fig. 7).

Iteration 2. After the planning stage has been carried out, the results of this stage have been found. In system development, there is an iteration plan carried out in three iterations. Developers are working on stories - 09, stories - 10, stories - 07, and stories – 08 in the second iteration. Stories - 9 serve to fill in complete information on pet shop and pet lovers, stories – 10 serve to find out the facilities owned by the recipient, stories - 07 serve to enter pet data for each owner, stories - 08 serve to find out how many animals have been deposited at a particular pet shop or pet lovers.

Design. The design made by the developer is a CRC Card and spike solution prototype. The CRC Card will show each class’s tasks and what classes interact with that



Fig. 8. Pet Lovers and Pet Shop CRC Card for user stories in the second iteration.

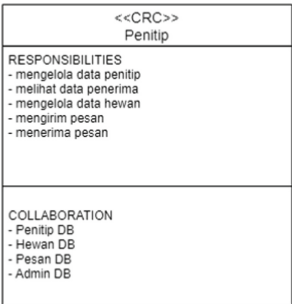


Fig. 9. Pet Lovers CRC Card for user stories in the second iteration.

class. Spike solution prototype is an interface design scheme based on the CRC Card from user stories in the second iteration (Figs. 8 and 9).

Program Code Generation. After completing the CRC prototype, it will continue with the next stage, namely the creation of program code. This phase is divided into three stages: Unit Testing, Code Generation, and Code Refactoring.

- **Unit Test**
Unit testing is done to test the function of the code to be made. In creating program code for unit testing, the developer first makes unit testing based on the CRC Card that has been made.
- **Code Generation**
After all unit testing has passed the test, the next step is to create program code consisting of the features that exist in each user story from the first iteration. This feature is in the form of a login feature for admin verification, custodians, and recipients (Fig. 10).
- **Code Refactoring**
This stage is carried out after the completion of the program code, the developer then maximizes the program code that has been created. In this iteration, the developer does not refactor because the code created does not cause any redundancies.

```
16 4' public void onCreateView(@NonNull View view, @Nullable Bundle savedInstanceState) {
17     mAuth = FirebaseAuth.getInstance();
18     mDb = FirebaseDatabase.getInstance().getReference().child("Users");
19
20     userName = (TextView) getActivity().findViewById(R.id.userName);
21     userUnama = (TextView) getActivity().findViewById(R.id.userUnama);
22     userPhone = (TextView) getActivity().findViewById(R.id.userPhone);
23     userEmail = (TextView) getActivity().findViewById(R.id.userEmail);
24     userHewan = (TextView) getActivity().findViewById(R.id.userHewan);
25     userCirihewan = (TextView) getActivity().findViewById(R.id.userCirihewan);
26     userFasilitas = (TextView) getActivity().findViewById(R.id.Fasilitas);
27     userAddress = (TextView) getActivity().findViewById(R.id.userAddress);
28
29     String user_id = mAuth.getCurrentUser().getId();
30     DatabaseReference userIni = mDb.child(user_id);
31
32     userIni.addValueEventListener(new ValueEventListener() {
33         @Override
34         4' public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
35             suserNama = (String) dataSnapshot.child("name").getValue();
36             userNama.setText(suserNama);
37             susername = (String) dataSnapshot.child("username").getValue();
38             userUnama.setText(susername);
39             sphone_number = (String) dataSnapshot.child("phone_number").getValue();
40             userPhone.setText(sphone_number);
41             saddress = (String) dataSnapshot.child("address").getValue();
42         }
43     });
44 }
```

Fig. 10. Iteration – 2 Code Generation.

<<CRC>> Penerima
RESPONSIBILITIES - mengelola data penerima - melihat data penitip - melihat data hewan - mengirim pesan - menerima pesan
COLLABORATION - Penitip DB - Hewan DB

Fig. 11. Pet Lovers and Pet Shop CRC Card for user stories in the third iteration.

Iteration 3. After the planning stage has been carried out, the results of this stage have been found, namely in system development, there is an iteration plan which will be carried out in three iterations. In the third iteration, the developer will work on stories - 05 which has a function to manage recruitment participant data. Meanwhile, stories - 06 have functions for managing value data and reporting participant results.

Design. The design made by the developer is a CRC Card and spike solution prototype. The CRC Card will show each class’s tasks and what classes interact with that class. Spike solution prototype is an interface design scheme based on the CRC Card from user stories in the third iteration (Figs. 11 and 12).

Program Code Generation. After completing the CRC prototype, it will continue with the next stage, namely the creation of program code. This phase is divided into three stages: Unit Testing, Code Generation, and Code Refactoring.

- **Unit Test**
Unit testing is done to test the function of the code to be made. In creating program code for unit testing, the developer first makes unit testing based on the CRC Card that has been made.

<<CRC>> Penitip
RESPONSIBILITIES - mengelola data penitip - melihat data penerima - mengelola data hewan - mengirim pesan - menerima pesan
COLLABORATION - Penitip DB - Hewan DB - Pesan DB - Admin DB

Fig. 12. Pet Lovers CRC Card for user stories in the third iteration.

```
64 public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
65     mAuth = FirebaseAuth.getInstance();
66     mDb = FirebaseDatabase.getInstance().getReference().child("users");
67
68     userName = (TextView) getActivity().findViewById(R.id.userName);
69     username = (TextView) getActivity().findViewById(R.id.username);
70     userPhone = (TextView) getActivity().findViewById(R.id.userPhone);
71     userEmail = (TextView) getActivity().findViewById(R.id.userEmail);
72     userHewan = (TextView) getActivity().findViewById(R.id.userHewan);
73     userCiriHewan = (TextView) getActivity().findViewById(R.id.userCiriHewan);
74     userFasilitas = (TextView) getActivity().findViewById(R.id.Fasilitas);
75     userAddress = (TextView) getActivity().findViewById(R.id.userAddress);
76
77     String user_id = mAuth.getCurrentUser().getId();
78     DatabaseReference userIni = mDb.child(user_id);
79
80     userIni.addValueEventListener(new ValueEventListener() {
81         @Override
82         public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
83             userName = (String) dataSnapshot.child("name").getValue();
84             username.setText(userName);
85             username = (String) dataSnapshot.child("username").getValue();
86             username.setText(username);
87             sphone_number = (String) dataSnapshot.child("phone_number").getValue();
88             userPhone.setText(sphone_number);
89             address = (String) dataSnapshot.child("address").getValue();
90         }
91     });
92 }
```

Fig. 13. Iteration – 3 Code Generation.

- **Code Generation**
After all unit testing has passed the test, the next step is to create program code consisting of the features that exist in each user story from the first iteration. This feature is in the form of a login feature for admin verification, custodians, and recipients (Fig. 13).
- **Code Refactoring**
This stage is carried out after the completion of the program code, the developer then maximizes the program code that has been created. In this iteration, the developer does not refactor because the code created does not cause any redundancies.

4.3 Application Result

See Figs. 14, 15, 16, 17, 18 and 19.

4.4 System Testing

After doing code refactoring, the next thing to do is test the overall implementation results in the third iteration of user stories. Testing uses the black box method, which the

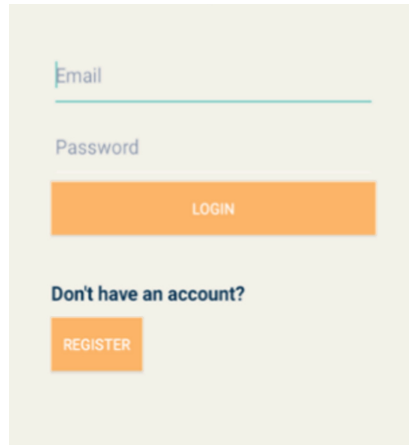
The image shows a login interface on a light beige background. At the top, there is a label 'Email' in blue text above a horizontal input field. Below this is a label 'Password' in blue text above another horizontal input field. Under the password field is an orange rectangular button with the word 'LOGIN' in white capital letters. Below the button is the text 'Don't have an account?' in blue. At the bottom is an orange rectangular button with the word 'REGISTER' in white capital letters.

Fig. 14. Login Feature.

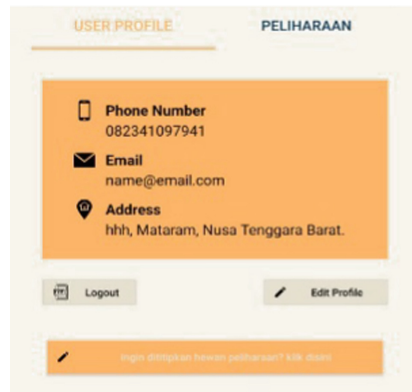
The image shows a user profile interface. At the top, there are two tabs: 'USER PROFILE' and 'PELIHARAAN'. Below the tabs is a large orange rectangular box containing user information. Inside this box, there is a phone icon next to 'Phone Number' with the value '082341097941', an envelope icon next to 'Email' with the value 'name@email.com', and a location pin icon next to 'Address' with the value 'hhh, Mataram, Nusa Tenggara Barat.' Below the orange box are two buttons: 'Logout' with a small icon and 'Edit Profile' with a pencil icon. At the bottom is an orange button with a pencil icon and the text 'ingin ditampilan kembali profilku? klik disini'.

Fig. 15. User Profil Feature.

developer and the client will carry out. Black box testing focuses on the system's functionality, so developers will create test cases to test the system's functionality previously determined. Furthermore, the functions of the system will be operated and reviewed. The client will provide input if changes or additions to the system are based on the visible test case results. In determining the assessment of the system, developers use the Linkert scale because many previous researchers used the Linkert scale to determine the assessment of the system.

5 Conclusion and Future Work

Based on the descriptions and explanations in the previous chapters, an analysis of the system running on the Mobile-Based Animal Care Information System has been carried out and tried to provide solutions by developing the system. Design of Mobile-Based



Fig. 16. Jelajah Feature.



Fig. 17. Chat Feature.



Fig. 18. Disscussion Feature.

ACCOUNT

Name

Email

Phone Number

Username

Password

ADDRESS DETAILS

Alamat Lengkap

REGISTER

Already have an account?

LOGIN

Fig. 19. Register Feature.

Animal Care Information System as an alternative solution in dealing with problems that occur.

5.1 Conclusion

Based on the descriptions and explanations in the previous chapters, an analysis of the system running on the Mobile-Based Animal Care Information System has been carried out and tried to provide solutions by developing the system. Design of Mobile-Based Animal Care Information System as an alternative solution in dealing with problems that occur.

1. Mobile-Based Pet Care Information System can make it easier for customers to find Pet Shops or for Pet Lovers to leave their pets
2. Mobile-Based Animal Care Information System can add job opportunities for some people
3. Mobile-Based Animal Care Information System can make it easier for customers to get information online about animal care.

5.2 Future Work

This Mobile-Based Animal Care Information System still has many shortcomings. As for suggestions to improve application performance in the future, namely:

1. Mobile-Based Pet Care Information System can make it easier for customers to find Pet Shops or for Pet Lovers to leave their pets
2. Mobile-Based Animal Care Information System can add job opportunities for some people
3. Mobile-Based Animal Care Information System can make it easier for customers to get information online about animal care.

References

1. R. Mayasari.: Differences in the level of loneliness and subjective well-being in individuals who live far from their families in terms of pet ownership. *Psychoborneo J. Ilm. psychology.*, vol. 6, no. 1, pp. 136–145 (2018)
2. R. Juliadilla and SC Hastuti H.: The Role of Pets (Pets) in Stress Levels of Retired Employees. *J. Psycho. Integra.*, vol. 6, no. 2, p. 153 (2019)
3. S. Sains, RR Nugroho, and SR Henim.: Pet Care Applications at Pet Shops Using iOS-Based Virtual Payment Features (2020)
4. IFB Andoro, RR Fayola, and SJ Prasentiono.: Development of Android-Based E-Health as an Effective, Efficient, Humane, Fair and Non-Discriminatory Health Service System. *Snatif*, vol. 5, pp. 409–414 (2018)
5. YY & A. Wijayanto: Easily Create and Do Business Android Applications with Android Studio - Yudha Yudhanto & Ardhi Wijayanto. Elex Media Komputindo, Jakarta (2018)
6. W. Sadewo.: Android-Mobile Design and Implementation System Marketing Surveyor (a-Ms2) Using the Personal Extreme Programming Method (Pxp). vol.7, pp. 39–45 (2016)
7. AF Septiyanto, W. Suharso, and I. Nuryasin.: The Family Hope Program Information System (PKH) Uses the Personal Extreme Programming Method with the Ranking Priority Method. *J. Repos.*, vol. 2, no. 12, p. 1671 (2020)
8. H. Rizal, S. Adhy, and PW Wirawan.: Design and Development of Android-Based Interactive Mobile Learning Using Personal Extreme Programming Method. *J. Masy. information.*, vol. 4, no. 8, pp. 1–10 (2013)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

