Dawwerha Project

CS Department

Project ID: UQU-CS-2021-G03

December 2021- May 2022

Dawwerha Application

An Application That Enhances People Awareness In Recycling And Disposing Wastes By Using Image Classification

Name	ID
Amani Mohammed Almuqati	438000292
Alaa Hesham Nabhan	438015431
Maram Abdullah Metro	436012457
Omniah Attiah Alharthi	438006477
Abrar Hassn Aljfry	438006568

Authors

Supervised by: Dr. Reem Saleh Al Ashaikh

Dept. of Computer Science

Faculty of Computer and Information Systems

Umm Al Qura University, KSA.

Contact Information

This project report is submitted to the Department of Computer Science at Umm Al-Qura University in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

Author(s):

Amani Mohammed Almuqati

E-mail: amine1999m@hotmail.com

Alaa Hesham Nabhan

E-mail: alaanabhan@hotmail.com

Maram Abdullah Metro

Email: marammetro@gmail.com

Omniah Attiah Alharthi

E-mail: omniahalharthi@gmail.com

Abrar Hassn Aljfry

E-mail: abraraljfry@gmail.com

University supervisor(s):

Dr. Reem Saleh Al Ashaikh

Dept. of Computer Science
Faculty of Computer and Information Systems
Umm Al Qura University

Internet: http://uqu.edu.sa
Phone: +966
Fax: +966
Kingdom of Saudi Arabia

Intellectual Property Right Declaration

This is to declare that the work under the supervision of Dr. Reem Saleh Al Ashaikh having title “Dawwerha Application: An application that enhances people awareness in recycling and disposing wastes by using image classification” carried out in partial fulfillment of the requirements of Bachelor of Science in Computer Science, is the sole property of the Umm Al Qura University and the respective supervisor and is protected under the intellectual property right laws and conventions. It can only be considered/ used for purposes like extension for further enhancement, product development, adoption for commercial/organizational usage, etc., with the permission of the University and respective supervisor.

This above statement applies to all students and faculty members.

Date: _____

Author(s):

Name: Amani Mohammed Almuqati Signature: _____

Name: Alaa Hesham Nabhan Signature: _____

Name: Maram Abdullah Metro Signature: _____

Name: Omniah Attiah Alharthi Signature: _____

Name: Abrar Hassn Aljfry Signature: _____

Supervisor:

Name: Dr. Reem Saleh Al Ashaikh Signature: _____

Anti-Plagiarism Declaration

This is to declare that the above publication produced under the supervision of Dr.Reem Saleh AL Ashaikh having title " Dawwerha Application :An application that enhances people awareness in recycling and disposing wastes by using image classification" is the sole contribution of the author(s) and no part hereof has been reproduced illegally (cut and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is proven.

Date: _____

Author(s):

Name: Amani Mohammed Almuqati Signature: _____

Name: Alaa Hesham Nabhan Signature: _____

Name: Maram Abdullah Metro Signature: _____

Name: Omniah Attiah Alharthi Signature: _____

Name: Abrar Hassn Aljfry Signature: _____

ACKNOWLEDGMENTS

قال تعالى { يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ ۚ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ }

First and foremost, we would like to thank God for all the blessings and for this success, we also thank for our parents for their praying and support from the first step in our education journey until this moment. This project would not be complete without Dr. Reem Al-Sheikh that guided us in doing this project, she provided us with invaluable advice and helped us in difficult periods, her motivation and help contributed tremendously to the successful completion of the project. We would like also to thank the people who made suggestions that contributed to the development of this project. We hope from God that this project will make progress in increasing the people awareness in recycling and disposing wastes.

ABSTRACT

Undoubtedly, recycling is important for reducing environmental pollution, saving resources, reducing incineration and saving energy. Recycling has become a necessity because many factors have led to an increase in environmental pollution, including an increase in human waste due to an increase in resource consumption, a lack of appropriate recycling methods and a lack of awareness. Our application of recycling enhances awareness among people by providing ways to dispose of waste that can be recycled and ideas to benefit from it. providing image classification technology using machine learning that display the classification of the material in the image according to the materials that are recycled. calculating the amount of waste consumption of the person and its effects on the environment and motivating people to recycle by display their impact on the environment. We hope in the future that recycling companies will adopt the provision of financial incentives or discounts through our application for users and provide representatives to receive materials collected by users that can be recycled from all regions in Saudi Arabia, or provide containers in the neighborhoods of regions in Saudi Arabia to collect materials that can be recycled.

Keywords: Environment, Recycling, Supervised machine learning, Image classification, Persuasive of technology, Teachable Machine, Dataset.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION 1

1.1. Background 1

1.1.1. Recycling 1

1.1.2. Machine Learning 1

1.1.2.1. Image Classification 1

1.1.2.2. TensorFlow 2

1.1.2.3. Teachable Machine 2

1.1.3. Persuasive Technology 4

1.1.4. End User Characteristics 4

1.2. Existing System 5

1.2.1. Existing System Description 5

1.2.2. Problems in The Existing System 5

1.3. Proposed Solutions 6

1.4. Alternative Solutions 6

1.5. Future Enhancement 6

1.6. Purpose of The Project 6

1.7. Purpose of This Document 7

1.8. Development Method 7

1.9. Overview of this Document 8

1.9.1. Chapter 1: Introduction 8

1.9.2. Chapter 2: System Analysis 8

1.9.3. Chapter 3: System Design Constraints 8

1.9.4. Chapter 4: Prototype 8

1.9.5. Chapter 5: Implementation 8

1.9.6. Chapter 6: Testing 8

1.10. Project Management Plan 9

CHAPTER 2: SYSTEM ANALYSIS 10

2.1. Data Collection 10

2.1.1. Questionnaire 10

2.2. Data Analysis 10

2.2.1. System Requirement 10

2.2.1.1. Functional Requirements 10

2.2.1.2. Non-Functional Requirements 13

2.2.2. Data Flow Diagrams 14

2.2.2.1. Context Diagram 14
2.2.2.2. DFD level 0 15
CHAPTER 3: SYSTEM DESIGN CONSTRAINTS 16
3.1. System Modeling 16
3.1.1. Interaction Models 16
3.1.1.1. Use Case Diagram 16
3.1.1.2. Use Case Scenario 17
3.1.1.3. Sequence Diagram 25
3.1.2. Structural Models 32
3.1.2.1. Class Diagram 32
3.1.2.2. ER Diagram 33
3.1.2.3. Schema Relationships 34
3.1.2.4. Document Diagram 35
3.1.3. Behavioral Models 36
3.1.3.1. Activity Diagram 36
3.1.4. Architecture Design 41
3.1.4.1. MVC Pattern 41
3.2. Design constraints 41
3.2.1. Hardware and Software Environment 41
3.2.1.1. Hardware Environment 41
3.2.1.2. Software Environment 42
CHAPTER 4: Prototype 43
4.1. Prototype 43
CHAPTER 5: Implementation 47
5.1 Dawwerha App Codes 47
5.1.1 Sign up 47
5.1.2 Login 50
5.1.3 Forget password 51
5.1.4 Tutorial 52
5.1.5 Profile 53
5.1.6 Change password 55
5.1.7 Take photo 56
5.1.8 Add material quantity 59
5.1.9 Companies contact information 61
5.1.10 Recycling codes 62

5.1.11 Suggested channels 63
5.1.12 User impact 64
5.1.13 Logout 66
5.1.14 Bottom bar 67
5.1.15 Top-drop-down menu 67
5.2 Database Implementation 68
5.3 Machine learning Implementation 73
CHAPTER 6: Testing 75
6.1 Development Testing 75
6.1.1 Unit testing 75
6.1.1.1 Opening and welcome unit testing case 75
6.1.1.2 Sign up unit testing case 76
6.1.1.3 Log in unit testing case 77
6.1.1.4 Forgot password unit testing case 78
6.1.1.5 Homepage unit testing case 79
6.1.1.6 Add material quantity unit testing case 79
6.1.1.7 Take photo unit testing case 80
6.1.1.8 User impact unit testing case 81
6.1.1.9 Recycling companies contact information unit testing case 83
6.1.1.10 User profile unit testing case 83
6.1.1.11 Change password unit testing case 84
6.1.1.12 Suggested channels for recycling ideas unit testing case 84
6.1.1.13 Recycling codes unit testing case 85
6.1.1.14 Tutorial unit testing case 85
6.1.1.15 About us unit testing case 86
6.1.1.16 Skip unit testing case 86
6.1.1.17 Notifications unit testing case 87
6.1.2 Machine learning testing 88
6.1.2.1 Paper classification testing case 88
6.1.2.2 Glass classification testing case 90
6.1.2.3 Metal classification testing case 92
6.1.2.4 Plastic classification testing case 94
6.1.3 Usability testing 96
Conclusion 97
References 98

Appendix.....101

LIST OF FIGURES

Figure 1.1: Teachable Machine Model Classes Accuracy 2
Figure 1.2: Teachable Machine Setting 3
Figure 1.3: QKONS app 5
Figure 1.4: Trashly app 5
Figure 1.5: Tadweer app 5
Figure 1.6: live love recycle app 5
Figure 1.7: Waterfall model 7
Figure 1.8: Project management plan 9
Figure 2.1: Context Diagram for Dawwerha application 14
Figure 2.2: DFD Diagram (level 0) for Dawwerha application cont. 15
Figure 2.3: DFD Diagram (level 0) for Dawwerha application 15
Figure 3.1: Use Case Diagram for Dawwerha application 16
Figure 3.2: Sequence Diagram for Sign up 25
Figure 3.3: Sequence Diagram for Log in 25
Figure 3.4: Sequence Diagram for Tutorial 26
Figure 3.5: Sequence Diagram for View user's profile 26
Figure 3.6: Sequence Diagram for Edit user's profile 27
Figure 3.7: Sequence Diagram for Take photo 27
Figure 3.8: Sequence Diagram for Recycling materials categories 28
Figure 3.9: Sequence Diagram for Add material quantity 28
Figure 3.10: Sequence Diagram for View companies contact 29
Figure 3.11: Sequence Diagram for Learn more 29
Figure 3.12: Sequence Diagram for Recycling ideas 30
Figure 3.13: Sequence Diagram for Find out user environmental impact 30
Figure 3.14: Sequence Diagram for View about us 31
Figure 3.15: Sequence Diagram for Log out 31
Figure 3.16: Class Diagram for Dawwerha application 32
Figure 3.17: ER Diagram for Dawwerha application 33
Figure 3.18: Schema relationship for Dawwerha application 34
Figure 3.19: Document Diagram for Dawwerha application 35
Figure 3.20: Activity Diagram for Sign up 36
Figure 3.21: Activity Diagram for Log in 36
Figure 3.22: Activity Diagram for Tutorial 37
Figure 3.23: Activity Diagram for View user's profile 37
Figure 3.24: Activity Diagram for Edit user's profile 37
Figure 3.25: Activity Diagram for Take photo 38
Figure 3.26: Activity Diagram for Recycling materials categories 38
Figure 3.27: Activity Diagram for Add material quantity 38
Figure 3.28: Activity Diagram for View companies contact 39
Figure 3.29: Activity Diagram for Learn more 39
Figure 3.30: Activity Diagram for Recycling ideas 39
Figure 3.31: Activity Diagram for Find out user environmental impact 40
Figure 3.32: Activity Diagram for View about us 40
Figure 3.33: Activity Diagram for Log out 40
Figure 3.34: MVC Pattern for Dawwerha application 41

Figure 4.1: Opening Interface 44
Figure 4.2: Welcome Interface 44
Figure 4.3: Sign up Interface 44
Figure 4.4: Log in Interface 44
Figure 4.5: Forget password Interface 44
Figure 4.6: Homepage Interface 44
Figure 4.7: Select material Interface 45
Figure 4.8: Take a photo Interface 45
Figure 4.9: Add material Interface 45
Figure 4.10: Impact on the environment Interface 45
Figure 4.11: Sample Result Interface 45
Figure 4.12: Recycling Companies Contacts Interface 45
Figure 4.13: Homepage list Interface 46
Figure 4.14: View profile Interface 46
Figure 4.15: Change password Interface 46
Figure 4.16: Suggested channels Interface 46
Figure 4.17: Recycling codes Interface 46
Figure 4.18: Tutorial Interface 46
Figure 4.19: About us Interface 46
Figure 5.1: Dawwerha App Code for Sign up Conditions 47
Figure 5.2: Dawwerha App Code for Sign up Conditions 47
Figure 5.3: Dawwerha App Code for email already existed 48
Figure 5.4: Dawwerha App Code for Sign up process 48
Figure 5.5: Dawwerha App Code for Sign up process 48
Figure 5.6: Dawwerha App Code for sends notifications 49
Figure 3.7: Sequence Diagram for Take photo 60
Figure 5.8: Dawwerha App Code for login process 50
Figure 5.9: Dawwerha App Code for skip 51
Figure 5.10: Dawwerha App Code for Forget password 51
Figure 5.11: Dawwerha App Code for Forget password 52
Figure 5.12: Dawwerha App Code for Tutorial 52
Figure 5.13: Dawwerha App Code for Profile 53
Figure 5.14: Dawwerha App Code for Profile 53
Figure 5.15: Dawwerha App Code for Profile 54
Figure 5.16: Dawwerha App Code for Profile 54
Figure 5.17: Dawwerha App Code for change Password 55
Figure 5.18: Dawwerha App Code for change Password 55
Figure 5.19: Dawwerha App Code for Take photo 56
Figure 5.20: Dawwerha App Code for classify 56
Figure 5.21: Dawwerha App Code for classify 57
Figure 5.22: Dawwerha App Code for add quantity after take photo 57
Figure 5.23: Dawwerha App Code for add quantity after take photo 58
Figure 5.24: Dawwerha App Code for retrieve quantity 58
Figure 5.25: Dawwerha App Code for help 59
Figure 5.26: Dawwerha App Code for add material quantity 59
Figure 5.27: Dawwerha App Code for add material quantity 60
Figure 5.28: Dawwerha App Code for retrieve quantity 60

Figure 5.29: Dawwerha App Code for companies contact information....61
Figure 5.30: Dawwerha App Code for companies contact information....61
Figure 5.31: Dawwerha App Code for Recycling codes....62
Figure 5.32: Dawwerha App Code for Recycling codes....62
Figure 5.33: Dawwerha App Code for Suggested channels....63
Figure 5.34: Dawwerha App Code for Suggested channels....63
Figure 5.35: Dawwerha App Code for user impact calculation....64
Figure 5.36: Dawwerha App Code for user impact calculation....64
Figure 5.37: Dawwerha App Code for user impact....65
Figure 5.38: Dawwerha App Code for user impact....65
Figure 5.39: Dawwerha App Code for retrieves questions....66
Figure 5.40: Dawwerha App Code for logout....66
Figure 5.41: Dawwerha App Code for bottom bar....67
Figure 5.42: Dawwerha App Code for top-drop-down menu....67
Figure 5.43: Database Firebase Authentication contains the email and password for every user....68
Figure 5.44: Firebase Real-time Database contains the parent five nodes....69
Figure 5.45: Firebase Real-time Database for the content of Companies node....69
Figure 5.46: Firebase Real-time Database for the content of Recycling Codes node....70
Figure 5.47: Firebase Real-time Database for the content of User Impact node....70
Figure 5.48: Firebase Real-time Database for the content of Users node....71
Figure 5.49: Firebase Real-time Database for the content of channels node....71
Figure 5.50: Firebase Storage for storage images....72
Figure 5.51: Firebase Cloud massages for Notifications....72
Figure 5.52: Accuracy of machine training in jupyter....73
Figure 5.53: Machine training result (correct)....74
Figure 5.54: Machine training result (incorrect)....74
Figure 6.1: Unit testing for Start page....75
Figure 6.2: Unit testing for Welcome page....75
Figure 6.3: Unit testing for Sign up....76
Figure 6.4: Unit testing for Sign up without filling user name....76
Figure 6.5: Unit testing for Sign up without filling user email....76
Figure 6.6: Unit testing for Sign up with writing wrong form email....76
Figure 6.7: Unit testing for Sign up without filling user password....76
Figure 6.8: Unit testing for Sign up without filling user confirmation password....76
Figure 6.9: Unit testing for Sign up with writing confirmation password doesn't match the password....76
Figure 6.10: Unit testing for Sign up successfully....76
Figure 6.11: Unit testing for Log in....77
Figure 6.12: Unit testing for Log in without filling user email....77
Figure 6.13: Unit testing for Log in without filling user password....77
Figure 6.14: Unit testing for Log in with writing wrong email or password....77
Figure 6.15: Unit testing for Log in successfully....77
Figure 6.16: Unit testing for Forgot password....78
Figure 6.17: Unit testing for Forgot password without filling user email....78
Figure 6.18: Unit testing for Forgot password with writing email not found in database....78
Figure 6.19: Unit testing for Forgot password link sent successfully....78
Figure 6.20: Unit testing for Forgot password reset link....78
Figure 6.21: Unit testing for Forgot password reset password....78

Figure 6.22: Unit testing for Forgot password password changed successfully....78
Figure 6.23: Unit testing for Homepage....79
Figure 6.24: Unit testing for Homepage drop-down menu and a bottom bar....79
Figure 6.25: Unit testing for Add material quantity....79
Figure 6.26: Unit testing for Add material quantity successfully....79
Figure 6.27: Unit testing for Take photo....80
Figure 6.28: Unit testing for Add material before taking photo....80
Figure 6.29: Unit testing for Take photo help button....80
Figure 6.30: Unit testing for Take photo classify....80
Figure 6.31: Unit testing for add material quantity successfully....80
Figure 6.32: Unit testing for User impact page with questions 1,2....81
Figure 6.33: Unit testing for User impact page with questions 3,4....81
Figure 6.34: Unit testing for User impact page with question 5....81
Figure 6.35: Unit testing for User impact page with questions 6,7....81
Figure 6.36: Unit testing for User impact page with questions 8,9....81
Figure 6.37: Unit testing for User impact page with question 10....81
Figure 6.38: Unit testing for excellent result user impact....82
Figure 6.39: Unit testing for good result user impact....82
Figure 6.40: Unit testing for bad result user impact....82
Figure 6.41: Unit testing for User impact with not answering all the questions....82
Figure 6.42: Unit testing for Recycling companies contact information page....83
Figure 6.43: Unit testing for Profile page....83
Figure 6.44: Unit testing for Log out successfully....83
Figure 6.45: Unit testing for Change password page....84
Figure 6.46: Unit testing for Change password successfully....84
Figure 6.47: Unit testing for Suggested channels for recycling ideas page....84
Figure 6.48: Unit testing for Recycling codes page....85
Figure 6.49: Unit testing for Tutorial page....85
Figure 6.50: Unit testing for About us page....86
Figure 6.51: Unit testing for Skip....86
Figure 6.52: Unit testing for Notifications....87
Figure 6.53: Machine learning testing for correct paper classification....88
Figure 6.54: Machine learning testing for correct paper classification....88
Figure 6.55: Machine learning testing for correct paper classification....88
Figure 6.56: Machine learning testing for correct paper classification....88
Figure 6.57: Machine learning testing for correct paper classification....88
Figure 6.58: Machine learning testing for correct paper classification....88
Figure 6.59: Machine learning testing for correct paper classification....89
Figure 6.60: Machine learning testing for correct paper classification....89
Figure 6.61: Machine learning testing for wrong paper classification....89
Figure 6.62: Machine learning testing for correct paper classification....89
Figure 6.63: Machine learning testing for wrong paper classification....89
Figure 6.64: Machine learning testing for wrong paper classification....89
Figure 6.65: Machine learning testing for correct glass classification....90
Figure 6.66: Machine learning testing for correct glass classification....90
Figure 6.67: Machine learning testing for correct glass classification....90
Figure 6.68: Machine learning testing for correct glass classification....90

Figure 6.69: Machine learning testing for correct glass classification 90
Figure 6.70: Machine learning testing for correct glass classification 90
Figure 6.71: Machine learning testing for correct glass classification 91
Figure 6.72: Machine learning testing for correct glass classification 91
Figure 6.73: Machine learning testing for correct glass classification 91
Figure 6.74: Machine learning testing for wrong glass classification 91
Figure 6.75: Machine learning testing for wrong glass classification 91
Figure 6.76: Machine learning testing for correct metal classification 92
Figure 6.77: Machine learning testing for correct metal classification 92
Figure 6.78: Machine learning testing for correct metal classification 92
Figure 6.79: Machine learning testing for correct metal classification 92
Figure 6.80: Machine learning testing for correct metal classification 92
Figure 6.81: Machine learning testing for correct metal classification 92
Figure 6.82: Machine learning testing for correct metal classification 93
Figure 6.83: Machine learning testing for wrong metal classification 93
Figure 6.84: Machine learning testing for correct metal classification 93
Figure 6.85: Machine learning testing for wrong metal classification 93
Figure 6.86: Machine learning testing for wrong metal classification 93
Figure 6.87: Machine learning testing for correct plastic classification 94
Figure 6.88: Machine learning testing for correct plastic classification 94
Figure 6.89: Machine learning testing for correct plastic classification 94
Figure 6.90: Machine learning testing for correct plastic classification 94
Figure 6.91: Machine learning testing for correct plastic classification 94
Figure 6.92: Machine learning testing for correct plastic classification 94
Figure 6.93: Machine learning testing for correct plastic classification 95
Figure 6.94: Machine learning testing for correct plastic classification 95
Figure 6.95: Machine learning testing for correct plastic classification 95
Figure 6.96: Machine learning testing for wrong plastic classification 95
Figure 6.97: Machine learning testing for correct plastic classification 95
Figure 7: Questionnaire for Dawwerha application 102

LIST OF TABLES

Table 3.1: Use Case Scenario for Sign up 17
Table 3.2: Use Case Scenario for Log in 18
Table 3.3: Use Case Scenario for Tutorial 18
Table 3.4: Use Case Scenario for View user's profile 19
Table 3.5: Use Case Scenario for Edit user's profile 19
Table 3.6: Use Case Scenario for Take photo 20
Table 3.7: Use Case Scenario for Recycling materials categories 20
Table 3.8: Use Case Scenario for Add material quantity 21
Table 3.9: Use Case Scenario for View companies contact 21
Table 3.10: Use Case Scenario for Learn more 22
Table 3.11: Use Case Scenario for Recycling ideas 22
Table 3.12: Use Case Scenario for Find out user environmental impact 23
Table 3.13: Use Case Scenario for View about us 23
Table 3.14: Use Case Scenario for Log out 24
Table 6.1: Usability testing 96

CHAPTER 1: INTRODUCTION

This chapter will introduce you to the document and will be cover the background, existing systems, proposed solutions, alternative solutions, future enhancement, purpose of this project, purpose of this document, development method, overview of this document and project management plan.

1.1. Background

Many factors led to an increase in environmental pollution including the increase of human waste due to the increase in the consumption of resources the lack of appropriate recycling methods and the lack of awareness, so from here our role as an influential part in this world begins and how to help solving problem using an artificial intelligence technology. It keeps pace with technological progress and our study in the field of computer science and to align with the vision of the Kingdom of Saudi Arabia 2030, which aims to protect the environment by recycling waste. Our idea is about enhancing awareness among people by helping them to dispose and benefits from the waste that can be recycled by using helping feature like image classification technology and calculate the amount of the person's consumption of this waste and its effects on the environment where every person has a role in protecting it. [1]

1.1.1. Recycling

Recycling is the process of collecting and processing materials that would otherwise be thrown away as trash turning them into new products and it goes through three stages (collection and processing, manufacturing, purchasing new products made from recycled materials). The original product is destroyed in this process, usually through a melting process, but it used to form new products. Examples are plastic water bottles, glass, cans, and many more. Recycling can benefit your community and the reduces environment pollution. [2]

1.1.2. Machine Learning

Can be broadly defined as computational methods that use experience to improve performance or make accurate predictions. Machine learning is a sophisticated branch of computational algorithms designed to simulate human intelligence by learning from the surrounding environment. Machine learning is classified into several types, in this project we will use the supervised learning specifically the classification type that we need in this project. [3] [4]

1.1.2.1. Image Classification

refers to the task that attempts to comprehend an entire image as a whole and extracting information classes from image by using a suitable classification technique that compares the image patterns with the target patterns. The image classification accepts the given input images and produces output classification for identifying it. [5] [6]

1.1.2.2. TensorFlow

Google's large-scale machine learning library, this examination learns about image classification by utilizing the deep neural network (DNN) or otherwise called deep learning by utilizing system tensor flow. TensorFlow used with Python. [6] [7]

1.1.2.3. Teachable Machine

Is a web-based tool created by Google, stoj and Use All Five, with the help of TensorFlowJS, to make machine learning and artificial intelligence available to everyone. it allows you to train the computer to recognize either your images, sounds, and poses fast and easy without writing any machine learning code and creating machine learning models for you based on specific requirements through some steps.

We have followed a set of steps using Teachable Machine:

First searching for preprocessing datasets to train the model and make predictions for the four types of recycling materials that we have, we combined more than one dataset to fit our need because the pictures inside them were taken from different angles and with different strength of light which helps the model to classify more accurately and they were from (GitHub, Kaggle).

The total dataset contains 7215 pictures classify into four types of recycling materials, glass contains 2011 pictures, metal contains 1952 pictures, paper contains 1941 pictures and plastic contains 1347 pictures.

Second In the training and learning model process, these datasets are divided into training sets which contains 85% samples from each type and testing sets which contains 15% samples from each type for training and measuring the accuracy of the situation, and there is no validation sets because it consider as extra set to train multiple models and get the best model out of different models and there is no need of it in this project .The accuracy is only getting a number between 0 and 1 witch's equal to 1 or 100 percent , the accuracy of glass is 0.95, metal is 0.93, paper is 0.96 and plastic is 0.95 and the total accuracy is 0.9475 as shown in Figure 1.1. [22][23][24][25]

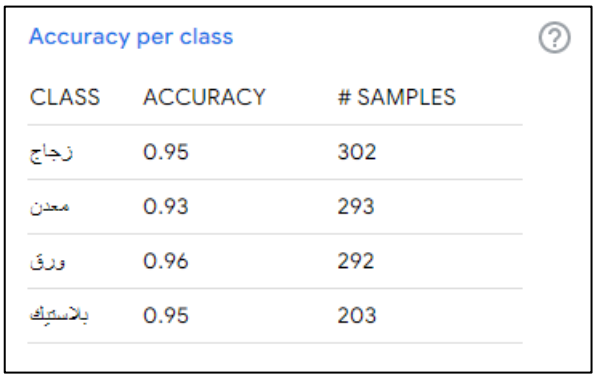


Figure 1.1: Teachable Machine Model Classes Accuracy

The teachable Machin models uses technique called transfer learning witch's a popular method in computer vision because it allows us to build accurate models in a timesaving way. instead of starting the learning process from scratch, you start from patterns that have been learned (pre-trained models) when solving a different problem. Accordingly, due to the computational cost of training such models, it is common practice to import and use models from published literature like MobileNet that has been used in this project.

The MobileNet model is designed to be used in mobile applications, and it is TensorFlow's first mobile computer vision model, and it is a class of CNN which is a good for training the classifiers.

There's a pretrained neural network and when creating the classes, the pictures of the classes becoming the last layer or step of the neural net.

Before creating the model there's need to specify several things the start with batch size is the number of samples that are passed to the network at once ,we have tried more than one batch size start from 32 to get a good result and finally reach the best batch size for the project which is 128.Then epochs, the one epoch refers to one cycle through the full training dataset which calculate it by the equation (Dataset ÷ Batch size) , in this project the epochs will equal 56 epochs, Then learning rate is a hyper-parameter that controls how much we are adjusting the weights of our network with respect the loss gradient, we have tried more than one learning rate to reduce the loss function and finally reach the best learning rate for the project which is 0.001 as shown in Figure 1.2.

last after completing the training process we can exporting the final model which have a good result so we can use it in our application.[26][27][28][29][30][31][32][33][34]

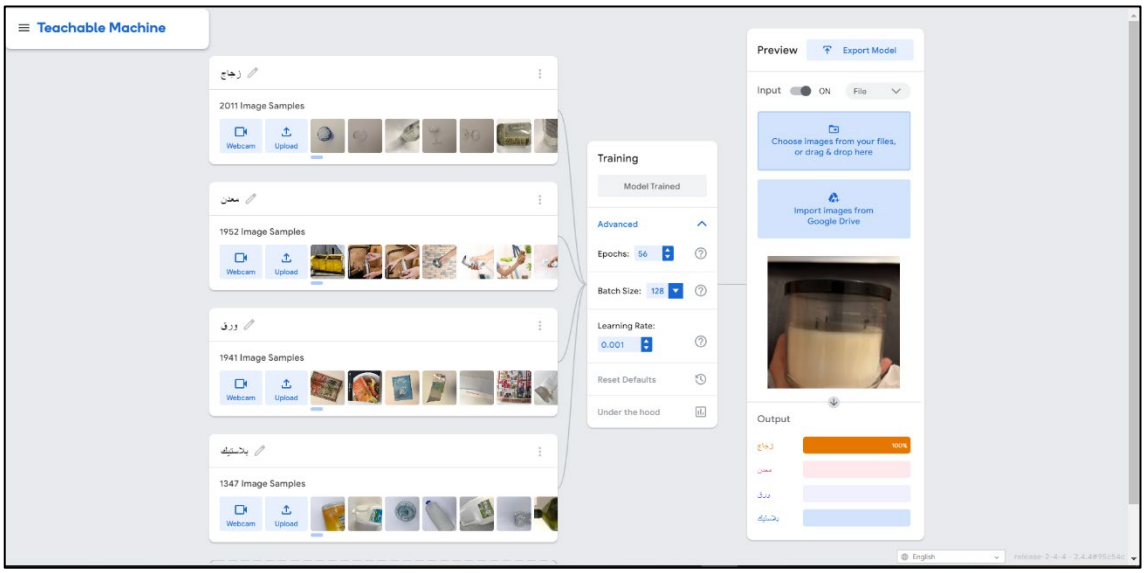


Figure 1.2: Teachable Machine Setting

1.1.3. Persuasive Technology

Today technology is taking an important role in user choices and behaviors, the emergence of the Internet has led to a proliferation of applications designed to persuade or motivate people to change their attitudes and behavior by using the persuasive technology. A persuasive technology is interactive tool designed to change attitudes or behaviors or both by making a desired outcome easier to achieve. persuasive technologies can be performed as tools, media, or social actors. [8] [9]

- **Primary Task Support (Reduction)**
  Strategy making the complex task simpler through by one click, the user can complete the process with ease and fast and this is increasing the effectiveness of users like image classification, add materials quantity, company button in our application. [9]

- **Dialogue Support (Reminders)**
  A strategy that makes the system remind users of their target behavior, the user can receive notification every 10 days that remind him and motivate the user to recycle in our application. [8]

- **Dialogue Support (Liking)**
  A strategy that aims to make the system more attractive to its users, so that our application contains interfaces related to recycling. [8]

- **System Credibility Support (Real-word feel)**
  This principle seeks to increase the credibility of the system as the system provides information about organizations and people who contribute to satisfying the user's need. [8]

1.1.4. End User Characteristics

This project aims to increase people's awareness about the environment and disposing of waste safely by collecting recyclable materials, as well as using materials that can be recycled at home. This application can be used by some organizations, schools and individuals who are concerned with the environment.

1.2. Existing System

We have searched for several applications similar to our idea for the project so that we have a general picture of what has been done in this field.

1.2.1. Existing System Description

1.2.1.1.	 *Figure 1.3: QKONS app*	QKONS	QKONS app is a digital waste exchange platform that allowing users to benefit economically from their transactions, while minimizing environmental impacts. The app allows waste generators to dispose of their waste in the most convenient and sustainable way by providing them with information about the nearest and most convenient waste bins and rewarding them for every transaction. [10]
1.2.1.2.	 *Figure 1.4: Trashly app*	Trashly	Trashly app is an application for recycling by product photography. The application works on iOS and Android systems and among the features of the application its alternatives to the landfill and provides recommendations to get rid of the most unused objects. [11]
1.2.1.3.	 *Figure 1.5: Tadweer app*	Tadweer	Tadweer app is responsible for monitoring and coordinating waste management activities throughout Dubai, including the implementation of the waste management strategy and all aspects of providing services necessary to establish an integrated waste management system with modern methods. [12]
1.2.1.4.	 *Figure 1.6: live love recycle app*	live love recycle	live love recycle is an app developed for the charitable association in Lebanon currently and is supported by several organizations to save the environment and the country. Recycling service is for several different types of things that can be recycled with determining the maximum capacity that the association can take and working times for them. [13]

Table 1.1: Existing System Description

1.2.2. Problems in The Existing System

As we reviewed, there are a few recycling applications, and if it is found, there are missing features on it such as it does not support the Arabic language, or it is not working, or does not support regions of Saudi Arabia.

Therefore, our application will enhance people awareness in recycling and disposing wastes and combines some features from these applications and provides them in the Arabic language.

1.3. Proposed Solutions

- The user can take photo of the products and then the type of the product is recognized using image classification technology.
- The user can also choose the material directly type without the need for photography.
- Through the application, the user can specify the quantities of materials that the user wants to recycle.
- The application allows them to view the quantities that have been added for recycling or reuse.
- The application also allows the user to know his/her long-term impact on the environment by answering several questions.
- The application provides information about recycling companies that the user can communicate with.
- The application suggests to the user to view ideas for recycling at home.
- The application explains the meanings of the recycling codes to the user.

1.4. Alternative Solutions

The service of communication and receiving materials that are intended to be recycled by the companies from the users through the delegate, but we already have a lot of services in our current application and adding this feature may make us go out of the scope of the application.

1.5. Future Enhancement

In the future, we will increase functional requirements such as:

- Add Communication feature between companies and users through the application.
- Add points and rewards within the application in cooperation with the companies to encourage users to recycle.
- Users obtain financial support in a certain percentage for each recycling quantity.
- Add feature display nearby companies' location.

1.6. Purpose of The Project

The application aims to enhances people awareness in recycling, by providing an image classification Technique that helps the user recognize the type of material to be recycled and its quantity and communicate with recycling companies easily, it is suitable for the age group of 18 and above and everyone who contributes and strives to save the environment.

The objectives of this project are:

- Take advantage of technology to save the environment by recycling materials.
- Facilitating reach contact information to organizations that can recycle materials.
- Using persuasive some technology technique in the motivating process.

1.7. Purpose of This Document

This document is aimed to describe the project's idea, including requirement gathering, designing process, and system analysis. The document analyzes and discusses related work and existing systems and states their advantages and disadvantages. It makes this valuable information clear and available to everyone.

1.8. Development Method

The waterfall model was chosen to be the model to develop this project, because the waterfall is flexible to use and it uses a clear structure and simple, also the requirements are known, and it goes through a clear and sequence of steps and never moving forward until the previous phase has been completed, and aims to document all the information throughout the project life cycle. [14]

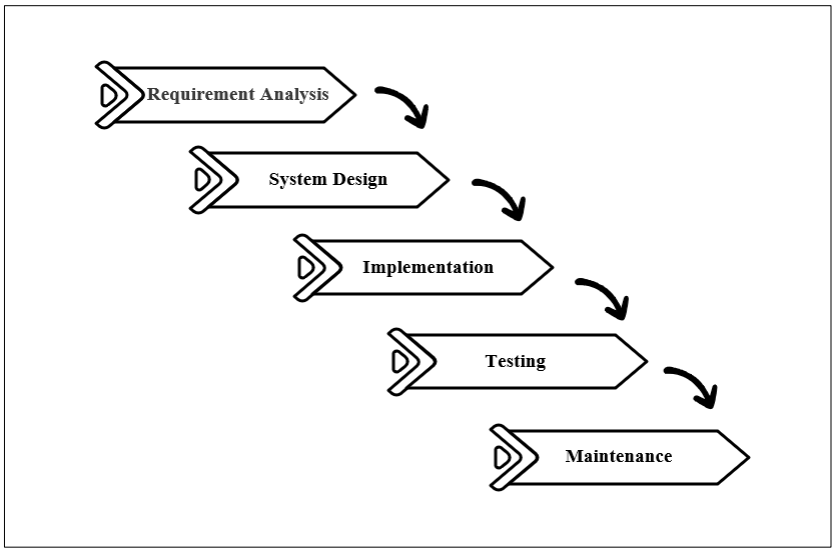


Figure 1.7: Waterfall model

As shown in Figure 1.7 the waterfall model contains five stages:

We worked on the first four waterfall methodology stages and we will clarify it based on our work on the project.

- Analyzing
  We fully understood the recycling idea and then we analyzed the requirements we need to build the application based on a questionnaire that was published to society.

- Designing
  The application requirements were completely and clearly defined, then we drew all the diagrams we need.

- Implementation
  We developed the entire application based on the requirements and the prototype together.

- Testing
  All functions have been tested for all interfaces, and we have also tried all possible possibilities for user input.

1.9. Overview of this Document

In this document, we laid out a full conception of our project, this document includes 4 chapters which are:

1.9.1. Chapter 1: Introduction

In this chapter we introduced the document, the purpose of this project, the purpose of writing this document, take an overview of the document, the existing system, the development method that used in the system, and the project management plan.

1.9.2. Chapter 2: System Analysis

In this chapter we described the system analysis phase which contains the data collection stage, the system requirements stage, the proposed and alternative solutions.

1.9.3. Chapter 3: System Design Constraints

In this chapter we represented the unified modeling language (UML) diagrams that needed in our system, the purpose of using it.

1.9.4. Chapter 4: Prototype

In this chapter we showed the prototype interfaces.

1.9.5. Chapter 5: Implementation

In this chapter we represented the implementation phase which including the application codes for the main functionality, database implementation and machine learning implementation.

1.9.6. Chapter 6: Testing

In this chapter we represented the testing phase including unit testing, machine learning testing and usability testing.

1.10. Project Management Plan

Figure 1.8 shows the progress of our works during the two semesters.

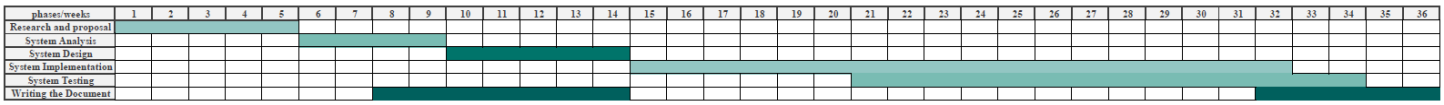


Figure 1.8: Project management plan

CHAPTER 2: SYSTEM ANALYSIS

This chapter will cover system analysis phase which contains the data collection, data analysis and data flow diagrams.

2.1. Data Collection

To collect data, we published a questionnaire in order to know people's opinions about the idea of the application and how much the need of it, via WhatsApp.

2.1.1. Questionnaire

The purpose of publishing this survey is to find details and information to clarify our idea more, and the survey was published on social network (WhatsApp). We asked the community members 10 closed questions and one open question, the first section of a questionnaire is about personal information such gender, age and city .The questions of the second section is: "Have you heard Before about the term recycling" and "how much they care about saving the environment", above 65% of the responses said they care about saving the environment, and "their consumption rate of products that can be recycled", "the difficulties in identifying consumables", "the difficulty of communicating with recycling companies", "their experience using smart phone applications" and "their need to an application that helps them in how to dispose of waste", more than 75% of the responses wants an application that helps them in recycling. In the open question there was lots of suggestions that helps us to build and develop our application, and we noticed that the most requested suggestions are: "ease of use of the application", "ease of communication with companies that is responsible for recycling", "providing recycling ideas that helps the user to use and recycle materials in a simple and easy way" and "motivational notifications messages to reminds the user".

Figure 5 in the appendix represents the questionnaire for Dawwerha application.

2.2. Data Analysis

Collect data from the environment, inspecting, cleansing, transforming, and modelling data with the goal of discovering useful information. [15]

2.2.1. System Requirement

The descriptions of the system services and constraints in the requirements engineering process. [15]

2.2.1.1. Functional Requirements

1. The user shall be able to Sign up in the System.

1.1 The system shall allow the user to sign up using his/her username, email and password.

1.1.1 If email is not existed in the database (unique), the system shall allow the user to sign up.

1.1.2 If email is existed in the database, the system shall view an error message to the user and asks him/her to re-write a not used email.

1.1.3 If username, email and password are written correctly (the username must not exceed 20 characters, the email must contain @ symbol, .com term and the password must contain upper-

and lower-case letters, numbers and must be eight characters), the system shall save all the information to the database and allow the user to sign up.

1.1.4 If the username email and/or password are written incorrectly the system shall view an error message to the user and asks him/her to re-write the wrong entry again.

1.1.5 If user entered the required data correctly, the system shall move the user to the home page.

1.1.6 If user entered the required data incorrectly, the system shall view an error message to the user and asks him/her to re-write the wrong entry again.

1.2 The system shall allow the user to skip the sign up.

1.3 The system shall send a motivational and reminding notification to the user every ten days.

2. The user shall be able to Login to the System.

2.1 The system shall allow the user to login using his/her email and password.

2.1.1 The system shall authentication from the database the email and password.

2.1.2 If the email and password are correct, the system shall move the user to the home page.

2.1.3 If the email or password is incorrect the system shall view an error message to the user and asks him/her to re-write them again or sign up.

2.1.4 If the user forgot the password the system should allow the user to reset his/her password by sending a link message through user's email.

2.2 The system shall allow the user to skip the Login.

3. The user shall be able to view a Tutorial (How to use Dawwerha app).

3.1 The system shall allow the user to view a tutorial about Dawwerha app which shows to the user how each part of the application works.

4. The user shall be able to view his/her Profile.

4.1 The system shall allow the user to view his/her information that retrieved from the database (username, email), and user collected quantity of material from the four categories.

5. The user shall be able to change his/her password

5.1 The system shall allow the user to change his/her password by entering (his/her email, the previous password and the new one) from his/her profile page and update it in the database.

6. The user shall be able to Take photo by using image classification feature.

6.1 The system shall allow the user to click the camera icon and take photo from it (by use image classification feature).

6.1.1 The system shall allow the user to view the instruction of taking the photo after click the help icon.

6.1.2 The system shall analyze the photo that taken by the user to classify it to one of the types from materials categories (plastic, glass, metal, paper).

6.2 The system shall allow the user to add the quantity number of the photographed material and added and update it in the database.

7. The user shall be able to add material quantity that he/she wants to recycle from the Recycling materials categories.

7.1 The system shall allow the user to choose one or more of the recycling materials categories that the user wants to be recycled.

7.1.1 The system shall allow the user to add the quantity that the user has from recycling material to the recycling category that has been chosen and added and update it in the database.

8. The user shall be able to add material quantity that he/she wants to recycle.

8.1 The system shall allow the user to add quantity number of materials that was classified by the system after taking photo by use image classification feature and added and update it in the database.

8.2 The system shall allow the user to add quantity number of materials that was specified by the user after choosing which category the material belongs to and added and update it in the database.

8.3 The system shall allow the user to view each type quantity of collected materials in the collected materials quantity in the profile page, after retrieved from the database.

9. The user shall be able to view Recycling companies contact information.

9.1 The system shall allow the user to view recycling companies contact information (name, phone number and address) after retrieve it from the database.

10. The user shall be able to view Recycling codes.

10.1 The system shall allow the user to view (Symbol, Code, Examples) of recycling codes for each category.

11. The user shall be able to view Suggested channels.

11.1 The system shall allow the user to view suggested channels of ideas for reusing materials that can be recycled at their home.

11.2 The system shall retrieve the suggested channels from the database and display it to the user.

12. The user shall be able to find out his/her Impact on the environment.

12.1 The system shall allow the user to find out his/her impact of waste on the environment through ten questions.

12.1.1 system shall allow the user to answer the impact on the environment questions after retrieve it from the database.

12.1.2 The system shall allow the user to view his/her result which will appear as percent, and several lines showing the user impact on the environment for the long term according to his/her answers.

13. The user shall be able to view About us.

13.1 The system shall allow the user to view about us, which explain the idea of dawwerha , and information (name, email) about the developers of this application.

14. The user shall be able to Logout from the system.

14.1 The system shall allow the user to logout from his/her profile page, then move the user to the welcome page.

2.2.1.2. Non-Functional Requirements

1. Security Requirements

1.1 The users shall set a strong password that includes UPPER- AND LOWER-CASE letters and numbers.

2. Usability Requirements

2.1 The system will be easy to use by users through using images and clear icons and buttons.

2.2 The system should give error messages if the user makes any mistakes.

2.3 The system should have a helping guide.

3. Availability Requirements

3.1 The application will be free.

3.2 The application will be available all the time.

3.3 The probability of the system's failure should not exceed five seconds per a day.

4. Performance Requirements

4.1 The system cannot take more than 1 millisecond for the response time.

5. Operational Requirements

5.1 The application will be a mobile application that can be run on Android devices smart phones.

6. Throughput Requirements

6.1 The system should be able to serve 100,000 users at the same time.

7. Cultural Requirements

7.1 The application will support the Arabic language.

2.2.2. Data Flow Diagrams

Data flow diagrams are used to graphically represent the flow of data. [16]

2.2.2.1. Context Diagram

The context diagram represents all the external components that may interact with the system. [16]

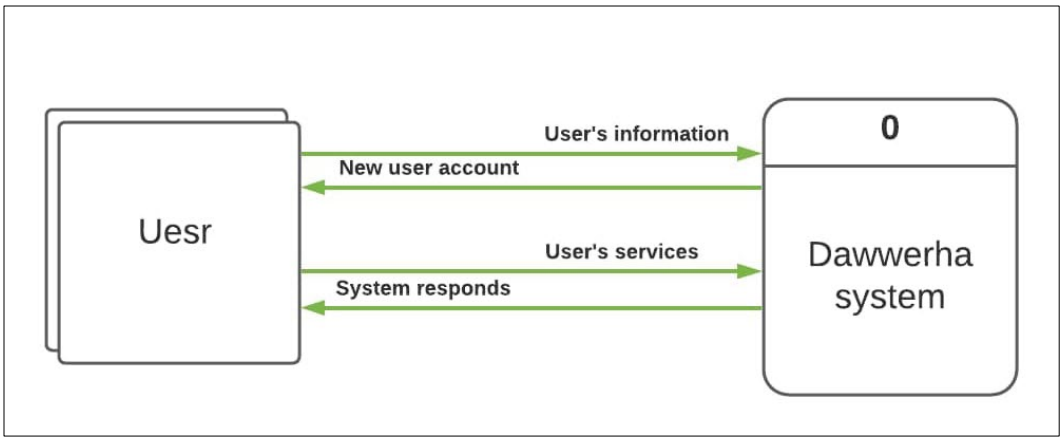


Figure 2.1: Context Diagram for Dawwerha application

2.2.2.2. DFD level 0

Data flow diagrams show the processes in the system, the data stores, and the data that passes between them and depicts the logical flow of information in a system, DFD helps system analysts and customers understand the processes during the requirements analysis phase. [16]

Figures 2.2 – 2.3 represents the DFD level 0 for Dawwerha application.

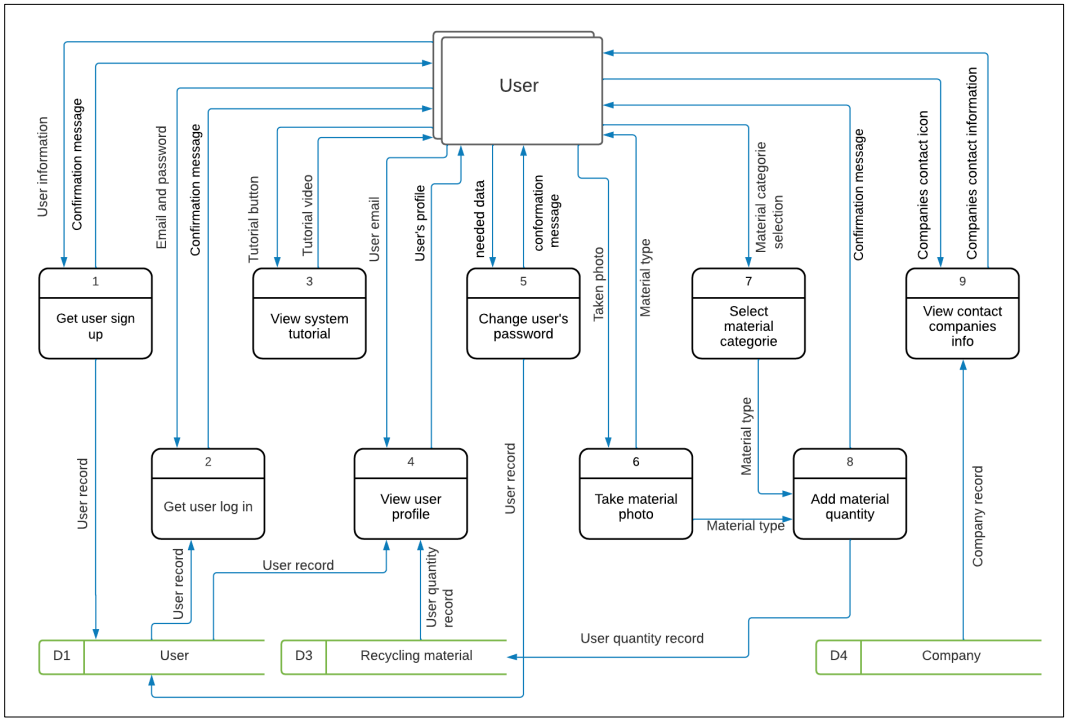


Figure 2.2: DFD Diagram (level 0) for Dawwerha application cont..

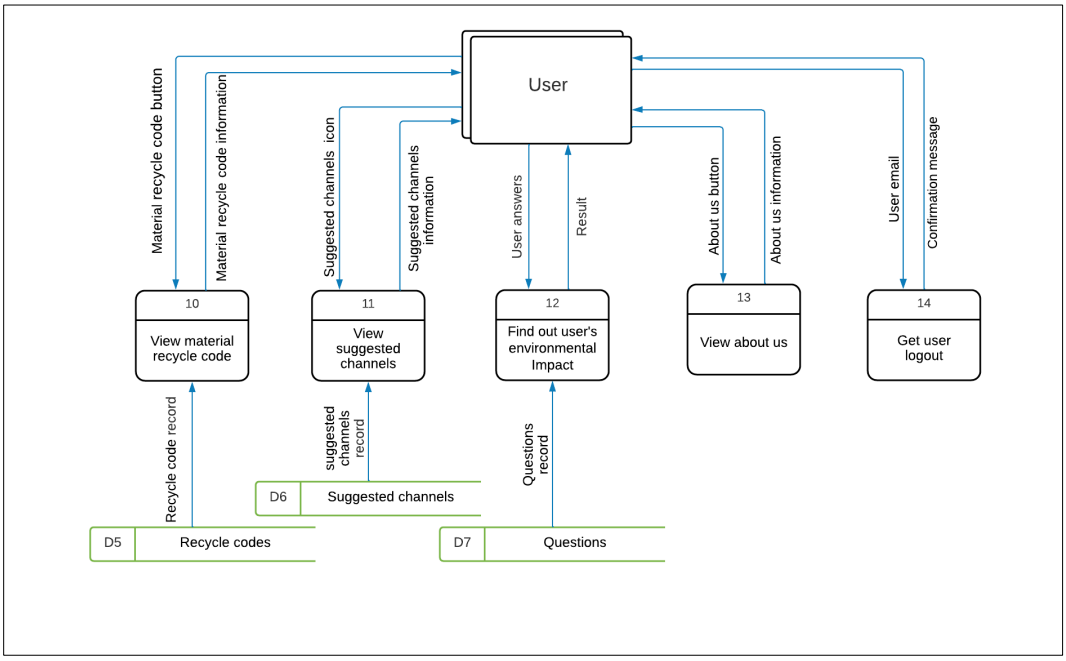


Figure 2.3: DFD Diagram (level 0) for Dawwerha application

CHAPTER 3: SYSTEM DESIGN CONSTRAINTS

This chapter will cover the system modeling diagrams including the interaction models, structural models, and behavioral models, architecture design and it will cover the design constraints to explain the hardware and software environment.

3.1. System Modeling

The process of developing abstract models of a system, in which each model presents a different viewpoint or perspective on that system. [15]

3.1.1. Interaction Models

Represents user interaction in the system, includes user inputs and outputs, and also shows the interaction between the components of the system among themselves. User interaction modeling is important because it helps define user requirements. [15]

3.1.1.1. Use Case Diagram

A use case diagram is a type of interaction diagram that aims to communicate at a high level what the system needs to do and the purpose of use shows the main functions of the system and the types of users who will interact with it, it also expresses system events and their flows. [16]

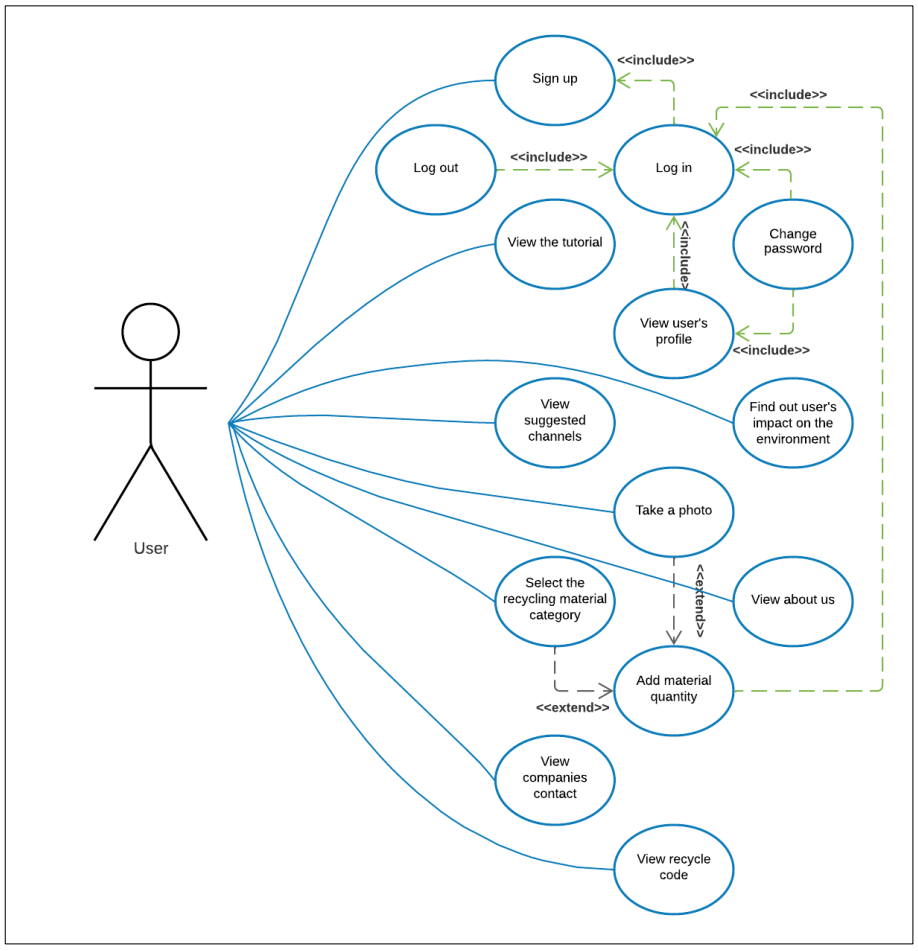


Figure 3.1: Use Case Diagram for Dawwerha application

3.1.1.2. Use Case Scenario

A use case scenario is a type of interaction diagram used to describe use cases and system requirements in more detail. [17]

Dawwerha System: Sign up	
Actors	User.
Description	▪ The system allows the user to Sign up using his/her username, email, password. ▪ If email is not existed in the database (unique), the system allows the user to sign up. ▪ If email is existed in the database, the system views an error message to the user and asks him/her to re-write a valid email. ▪ If both of email and password are written correctly (the email must contain @ symbol, .com term and the password must contain upper- and lower-case letters, numbers and must be eight characters), the system allow the user to sign up. ▪ If the email and/or password are written incorrectly the system shall view an error message to the user and asks him/her to re-write the wrong entry again. ▪ If user entered the required data correctly, the system moves the user to the home page. ▪ The system shall send a motivational and reminding notification to the user every ten days.
Data	▪ Username, email, password.
Stimulus	▪ Click on the sign-up button.
Response	▪ Checks the entries if they are written correctly and then they are saved in the database. ▪ Account is created. ▪ Successfully registered to the system and show the home page. ▪ If registered was wrong, an error message will appear.
comments	▪ If the user does not want to create the account just wants to view the application, click the skip button.

Table 3.1: Use Case Scenario for Sign up

Dawwerha System: Login	
Actors	User.
Description	▪ The system allows the user to login using his/her email and password. ▪ The system checks in the database the email and password. ▪ If the email is correct and the password is correct, the system moves the user to the home page. ▪ If the email or password is incorrect the system views an error message to the user and asks him/her to re-write them again or sign up.
Data	▪ email, password.
Stimulus	▪ Click on the login button.
Response	▪ Check the input in the database for logging into the system. ▪ Successfully login to the system and show the home page, if logging-in was wrong, an error message will appear.
comments	▪ If the user does not want to log in just wants to view the application, click the skip button.

Table 3.2: Use Case Scenario for Log in

Dawwerha System: View the tutorial (How to use Dawwerha app)	
Actors	User.
Description	▪ The system allows the user to view a tutorial about Dawwerha app which shows to the user how each part of the application works.
Data	▪ None.
Stimulus	▪ Click on the tutorial icon.
Response	▪ View the tutorial.
comments	▪ None.

Table 3.3: Use Case Scenario for Tutorial

Dawwerha System: View user's profile	
Actors	User.
Description	▪ The system allows the user to view his/her information (username, email), and user collected quantity of material from the four categories.
Data	▪ email.
Stimulus	▪ Click on profile icon.
Response	▪ View the profile.
comments	▪ The user not be able to view the profile unless having an account.

Table 3.4: Use Case Scenario for View user's profile

Dawwerha System: Change password	
Actors	User.
Description	▪ The system allows the user to change his/her password from the profile page. ▪ The system allows the user to change his/her password by entering the email, previous password and the new one.
Data	▪ Email, password.
Stimulus	▪ Click on change password button.
Response	▪ Confirmation message that the change has been done, then the profile will be displayed.
comments	▪ The user not be able to change the password unless having an account. ▪ If the user clicks on the change password, the user must enter the data to be changed.

Table 3.5: Use Case Scenario for Change password

Dawwerha System: Taka a photo	
Actors	User.
Description	▪ The system allows the user to click the camera icon and take photo from it (by use image classification feature). ▪ Then the system analyzes the photo that taken by the user to recognize and classify it according to the types of materials categories (plastic, glass, metal, paper).
Data	▪ Photo taken.
Stimulus	▪ Click on the camera icon and take photo.
Response	▪ The photo has been taken successfully and recognize the materials categories.
comments	▪ there is a help icon for the user, shows how to avoid the errors that may appear.

Table 3.6: Use Case Scenario for Take photo

Dawwerha System: Select the recycling materials category	
Actors	User.
Description	▪ The system allows the user selects one of the categories of recycling materials that the user wants to recycle.
Data	▪ Categories of recycled materials of glass, plastic, paper or metal.
Stimulus	▪ Click on the recycling materials categories icon.
Response	▪ Recycling materials category has been Selected.
comments	▪ None.

Table 3.7: Use Case Scenario for Recycling materials categories

Dawwerha System: Add materials quantity	
Actors	User.
Description	▪ The system allows the user to add the quantity number of materials that was recognized by the system after taking photo of the material by using image classification feature. ▪ The system allows the user to add quantity number of materials that was specified by the user after choosing which category the material belongs to. ▪ The system allows the user to view each type quantity of collected materials in the collected materials quantity in the profile page, after the system updating the user collected materials from the database with every added process.
Data	▪ Quantity number of materials.
Stimulus	▪ Click the add quantity button.
Response	▪ Quantity number of materials have been stored in the database.
comments	▪ None.

Table 3.8: Use Case Scenario for Add material quantity

Dawwerha System: View companies contact information.	
Actors	User.
Description	▪ The system allows the user to view recycling companies contact information (name, phone number and address).
Data	▪ None.
Stimulus	▪ Click on recycling companies contact information icon.
Response	▪ View Recycling companies contact information from database.
comments	None.

Table 3.9: Use Case Scenario for View companies contact

Dawwerha System: View Recycling codes	
Actors	User.
Description	▪ The system allows the user to view (Symbol, Code, Examples) of recycling codes for each category.
Data	▪ None.
Stimulus	▪ Click on the Recycling codes icon.
Response	▪ View all the details about recycling codes for each category from database.
comments	▪ None.

Table 3.10: Use Case Scenario for Recycling codes

Dawwerha System: View the Suggested channels	
Actors	User.
Description	▪ The system allows the user to view suggested channels of ideas for reusing materials that can be recycled at their home
Data	▪ None.
Stimulus	▪ Click on the Recycling ideas icon.
Response	▪ View all the details about Suggested channels for each category from database.
comments	▪ None.

Table 3.11: Use Case Scenario for Suggested channels

Dawwerha System: Find out user's impact on the environment.	
Actors	User.
Description	▪ The system allows the user to find out his/her impact of waste on the environment through ten questions. ▪ The system allows the user to view his/her result which will appear as percent, and several lines showing the user impact on the environment for the long term according to his/her answers.
Data	▪ The user answers.
Stimulus	▪ Clicks the submit answers bouton.
Response	▪ View the result.
comments	▪ Each question consists of four options and each option is specified with certain points, then these points are calculated to determine the final result and convert it into a percentage, where if the user percentage is 70% and higher it will have a good impact on the environment for a long term, if it is between 70% to 40% its impact will be seme good on the environment for a long term, and if it is 39% and lower its impact will be not good on the environment for a long term.

Table 3.12: Use Case Scenario for Find out user environmental impact

Dawwerha System: View about us.	
Actors	User.
Description	▪ The system allows the user to view about us, which view from where the idea came, about dawwerha and information (name and email) about the developers of this application.
Data	▪ None.
Stimulus	▪ Click on the about us icon.
Response	▪ View the about us page.
comments	▪ None.

Table 3.13: Use Case Scenario for View about us

Dawwerha System: Log out.	
Actors	User.
Description	▪ The system allows the user to logout move the user to logout from his/her profile page, then move the user to the welcome page.
Data	▪ None.
Stimulus	▪ Click on logout button.
Response	▪ Log out successfully, move to welcome page.
comments	▪ None.

Table 3.14: Use Case Scenario for Log out

3.1.1.3. Sequence Diagram

A Sequence diagram is a type of interaction diagram because it shows object interactions arranged in time sequence, to understand the system requirements, and to document an existing process. [16]

Figures 3.2 – 3.15 represents all the sequence diagram for Dawwerha application.

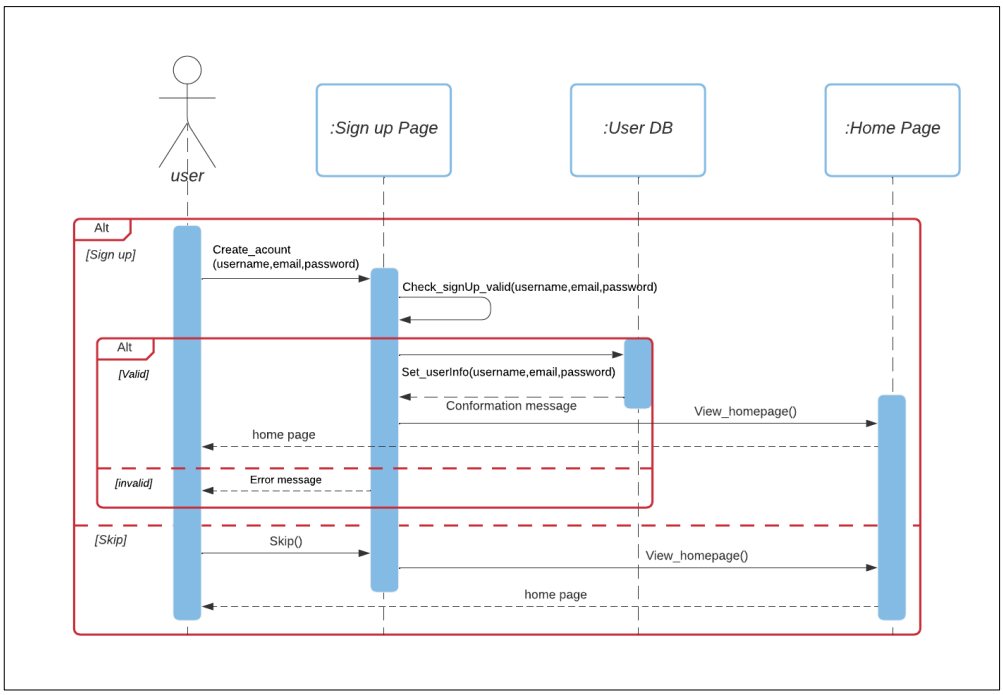


Figure 3.2: Sequence Diagram for Sign up

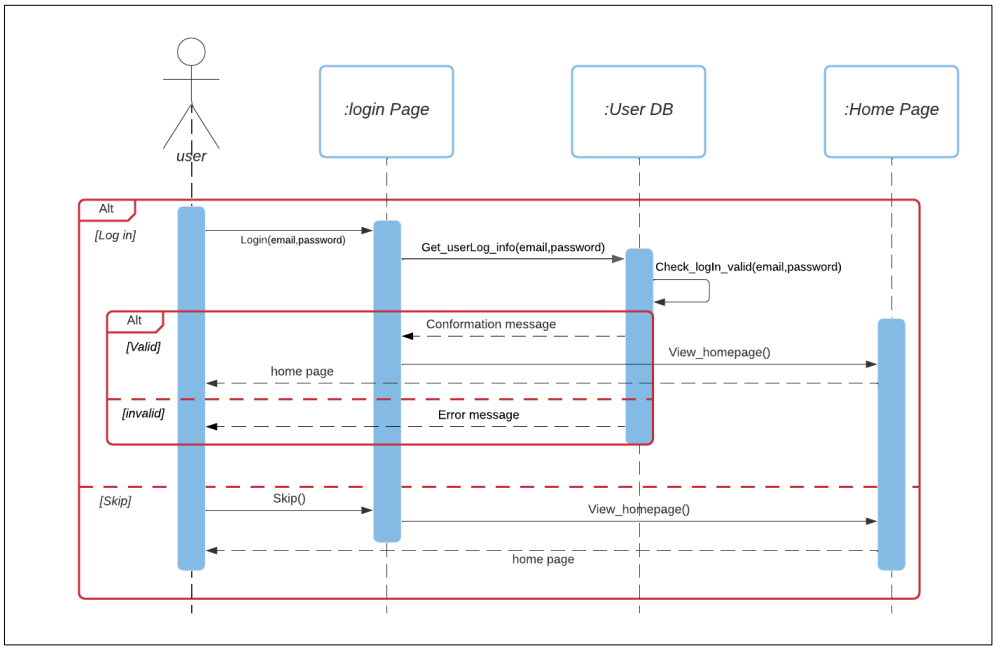


Figure 3.3: Sequence Diagram for Log in

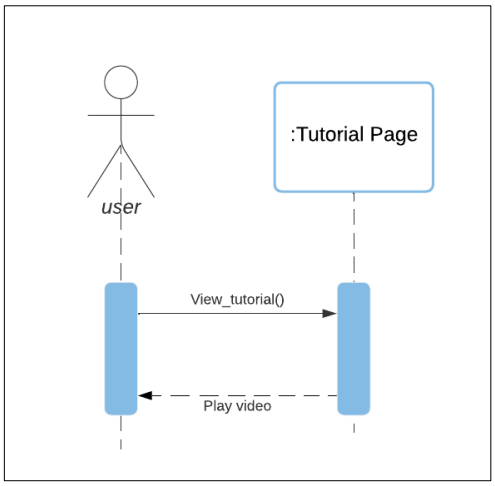


Figure 3.4: Sequence Diagram for Tutorial

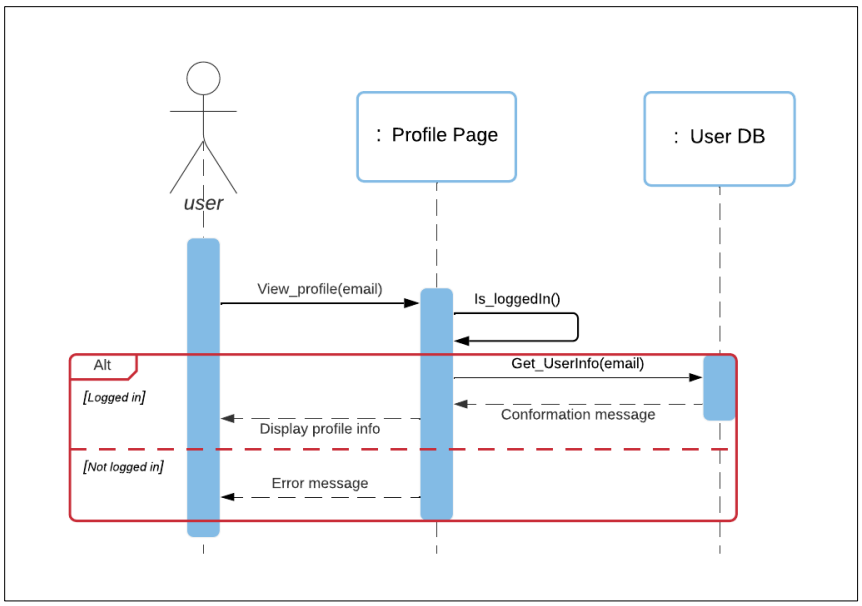


Figure 3.5: Sequence Diagram for View user's profile

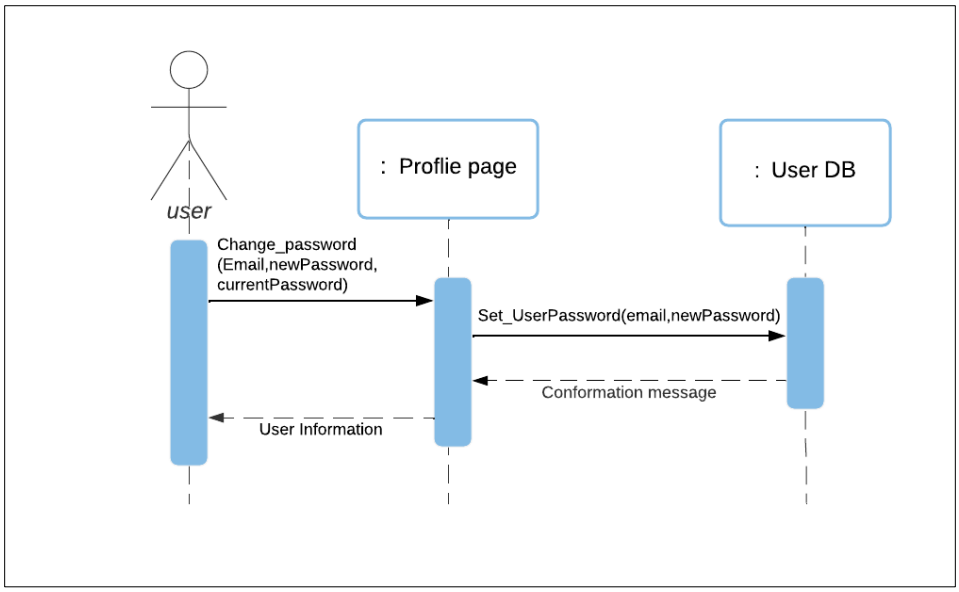


Figure 3.6: Sequence Diagram for Change password

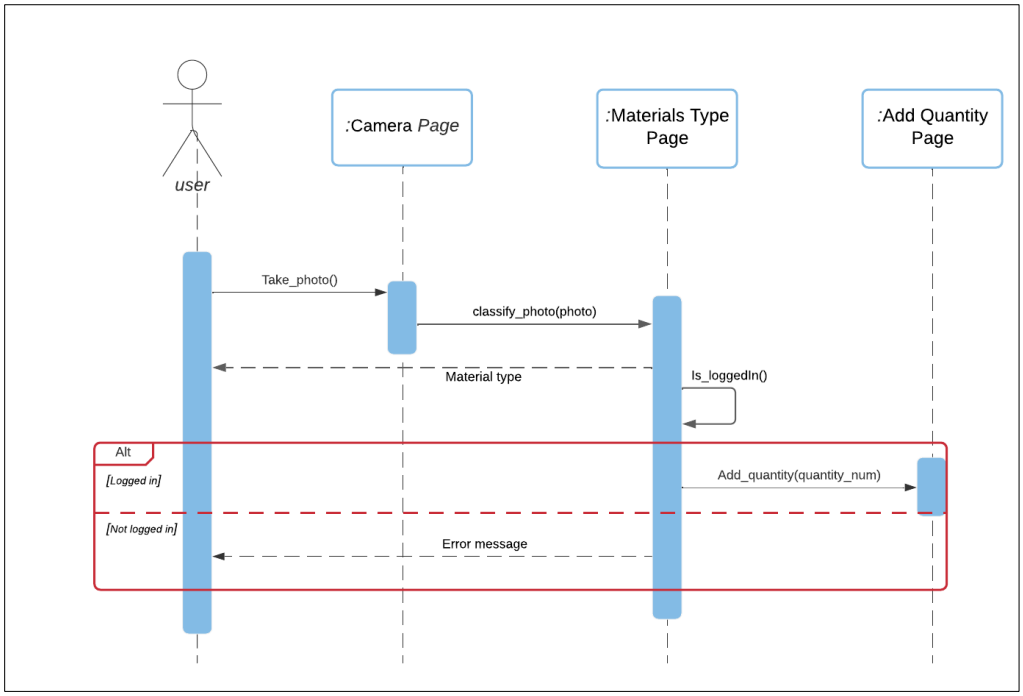


Figure 3.7: Sequence Diagram for Take photo

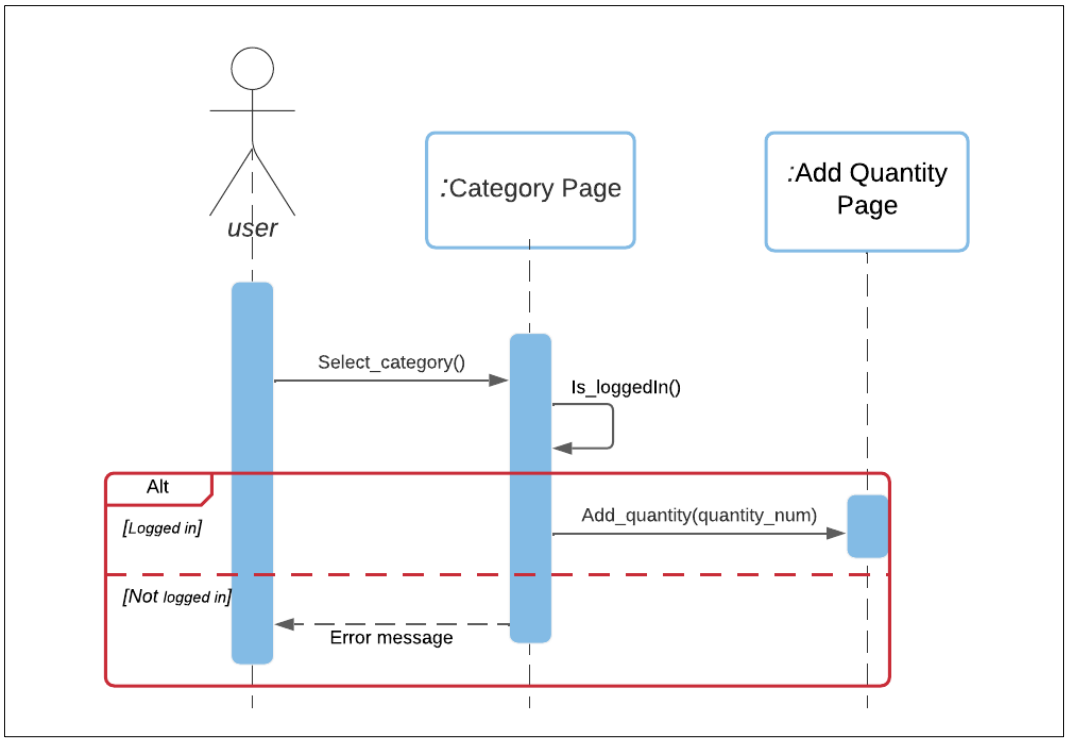


Figure 3.8: Sequence Diagram for Recycling materials categories

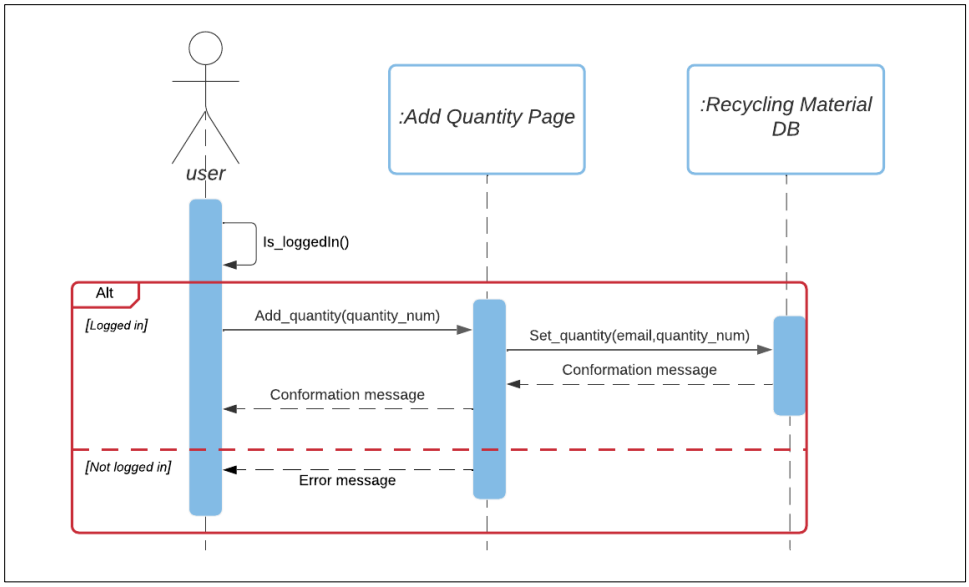


Figure 3.9: Sequence Diagram for Add material quantity

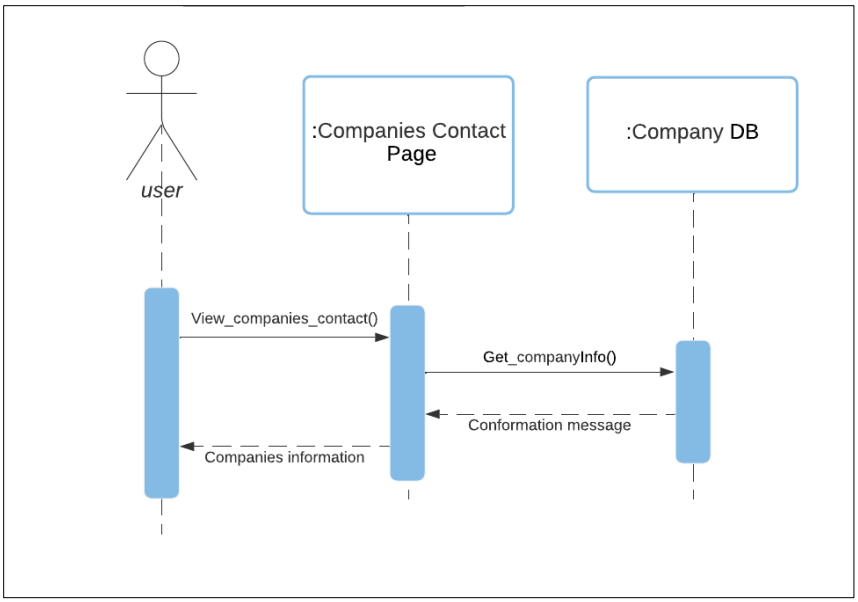


Figure 3.10: Sequence Diagram for View companies contact

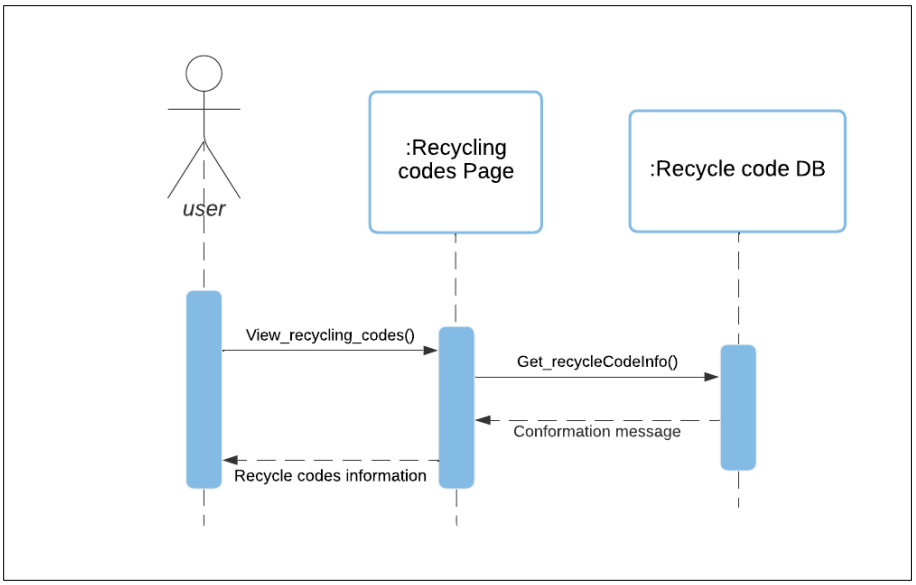


Figure 3.11: Sequence Diagram for Recycling codes

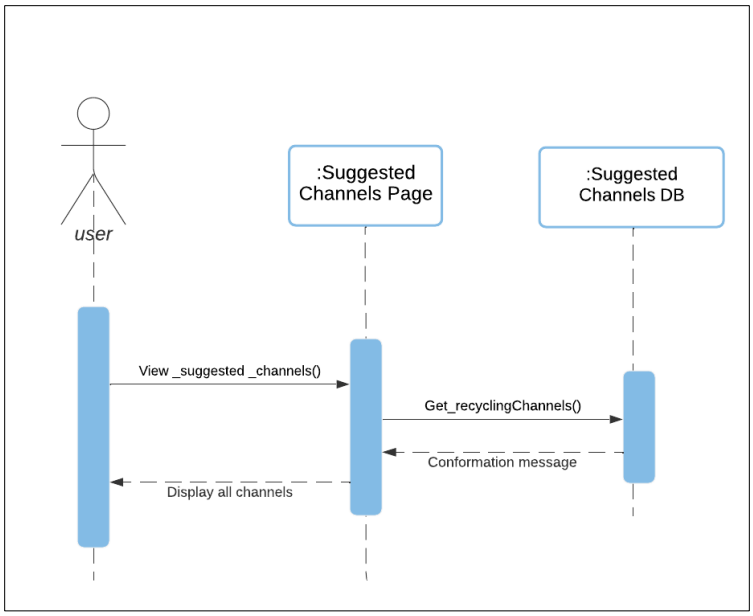


Figure 3.12: Sequence Diagram for Suggested channels

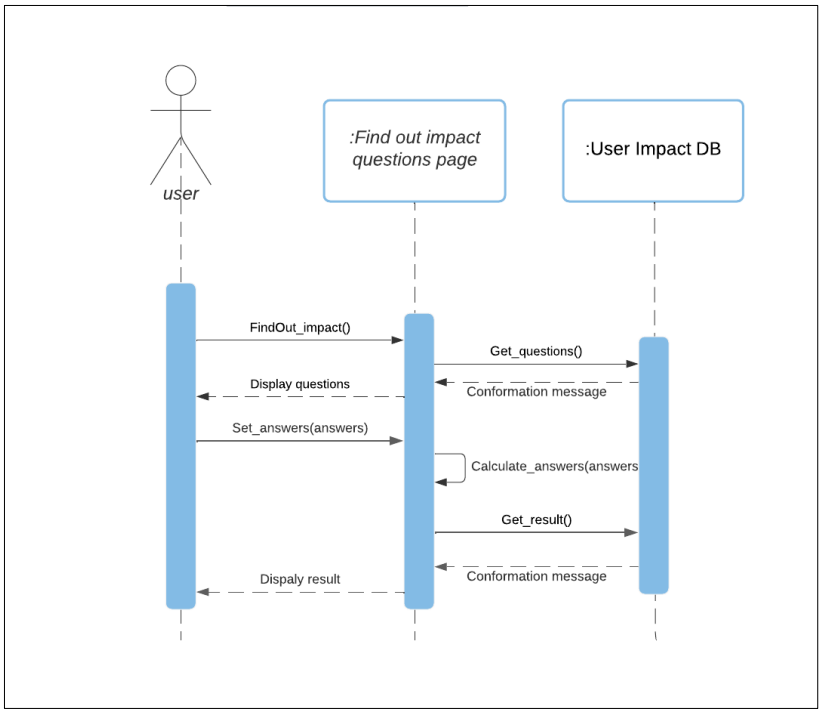


Figure 3.13: Sequence Diagram for Find out user environmental impact

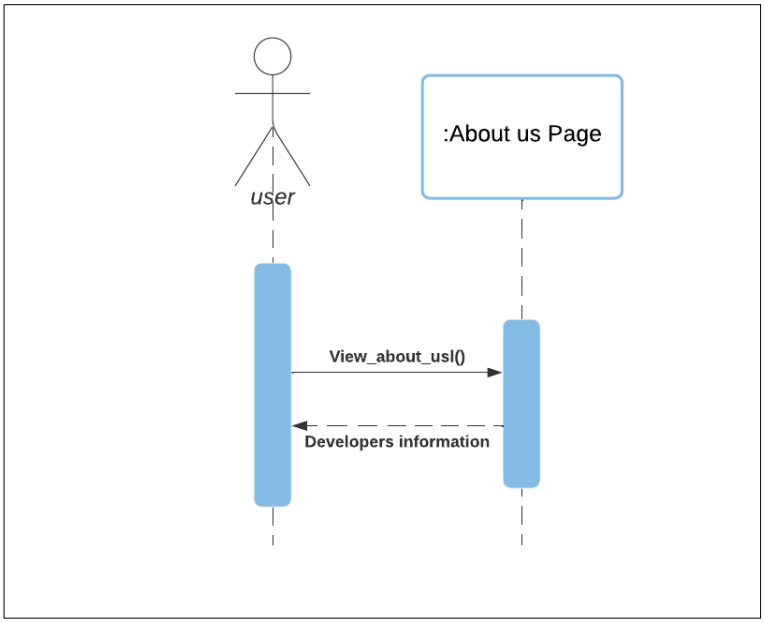


Figure 3.14: Sequence Diagram for View about us

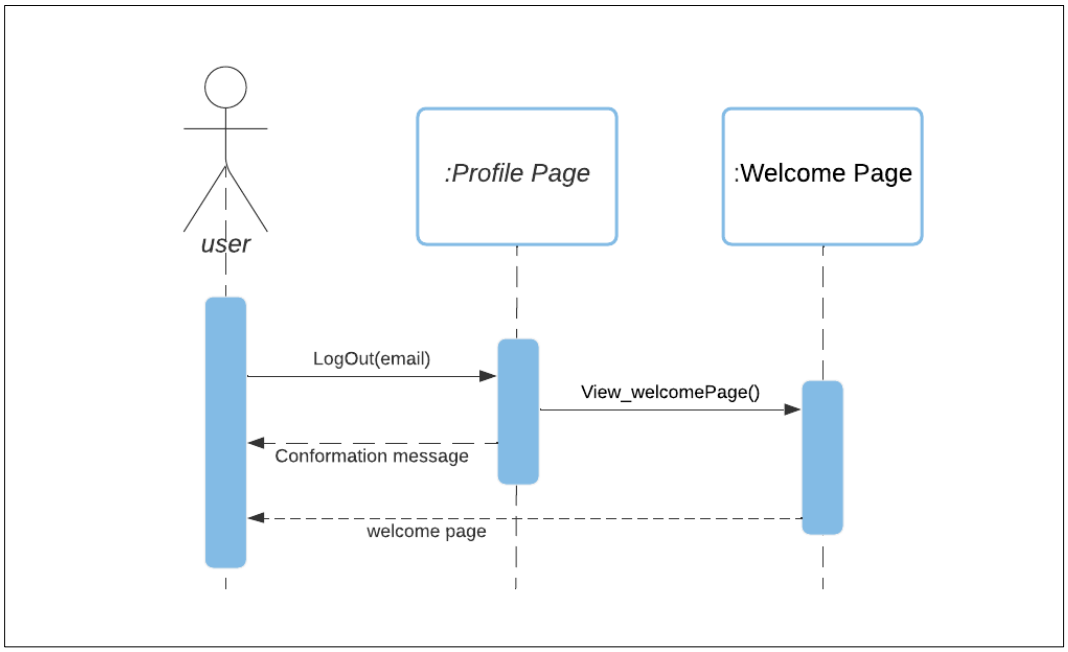


Figure 3.15: Sequence Diagram for Log out

3.1.2. Structural Models

Describes the structure of the objects that support the business processes in the organization. [18]

3.1.2.1. Class Diagram

A Class diagram is a graphical notation used to construct and visualize objects, it describes the structure of a system and it shows the classes and the relationships among the classes that remain constant in the system. [16]

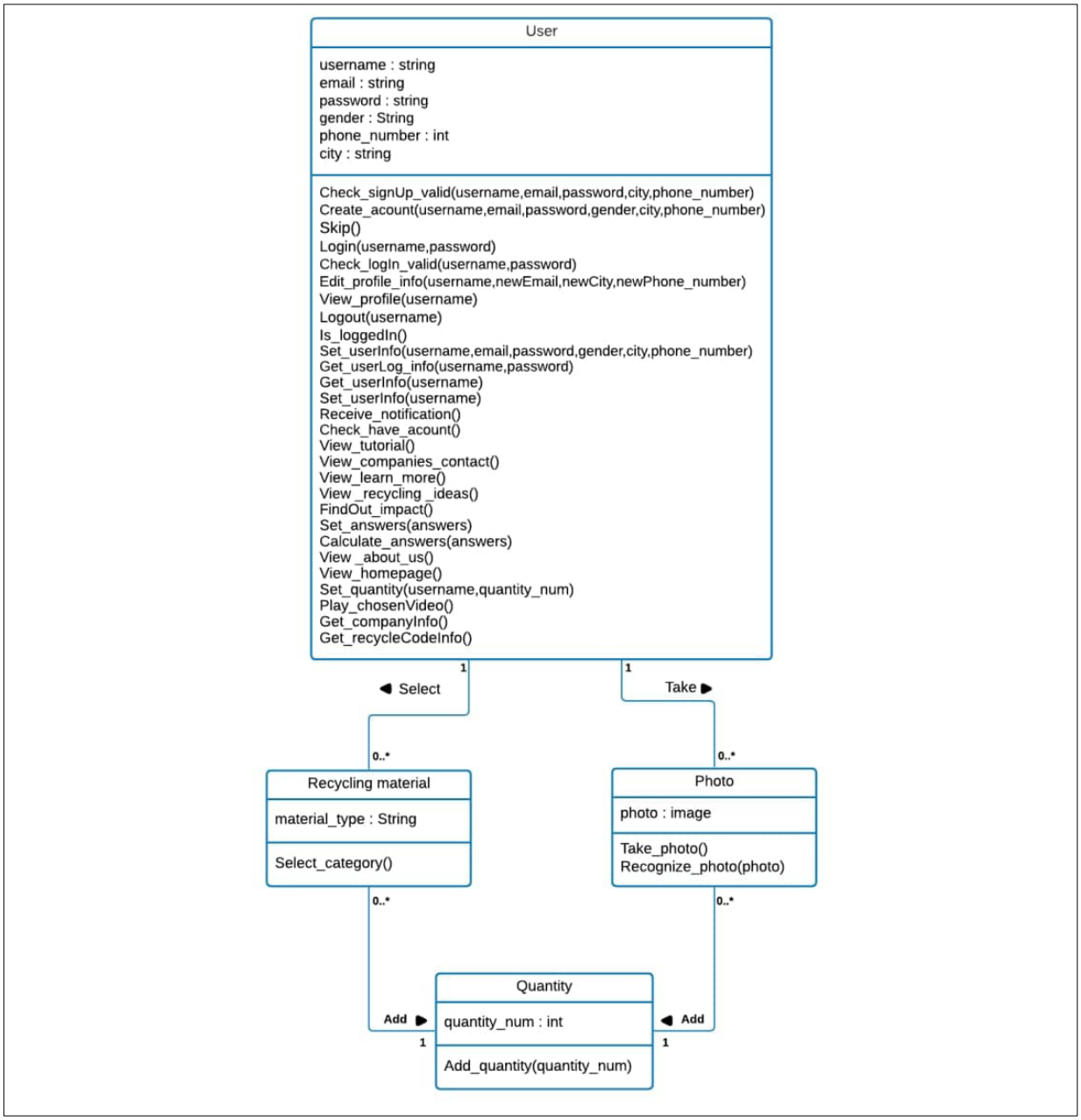


Figure 3.16: Class Diagram for Dawwerha application

3.1.2.2. ER Diagram

An Entity Relationship Diagram (ERD) is an image that shows information that has been generated and stored. The analyst can read the ERD to understand how the data is organized and related to each other. [16]

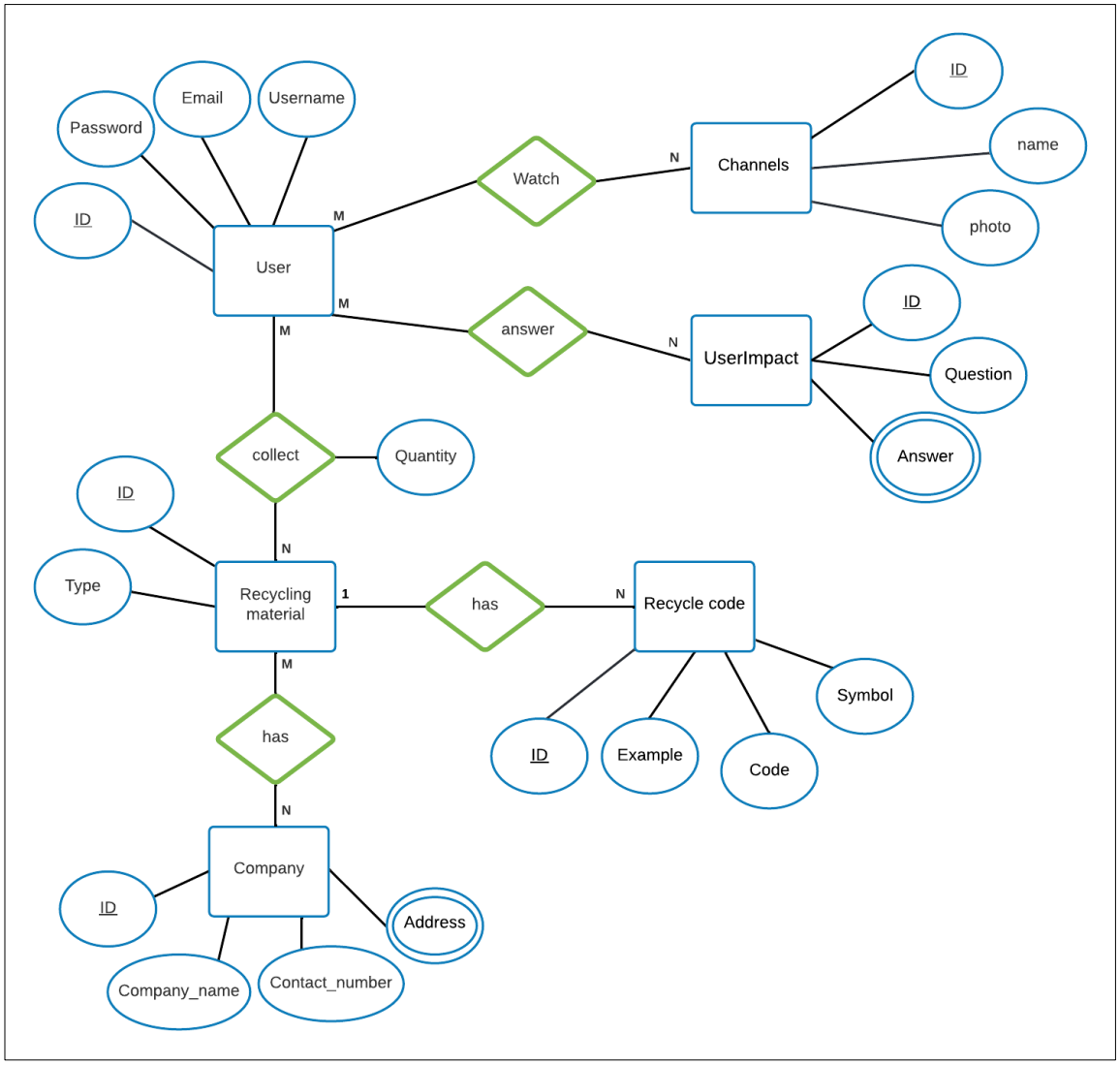


Figure 3.17: ER Diagram for Dawwerha application

3.1.2.3. Schema Relationships

The description of the database structure, data types and the constraints on the database. [19]

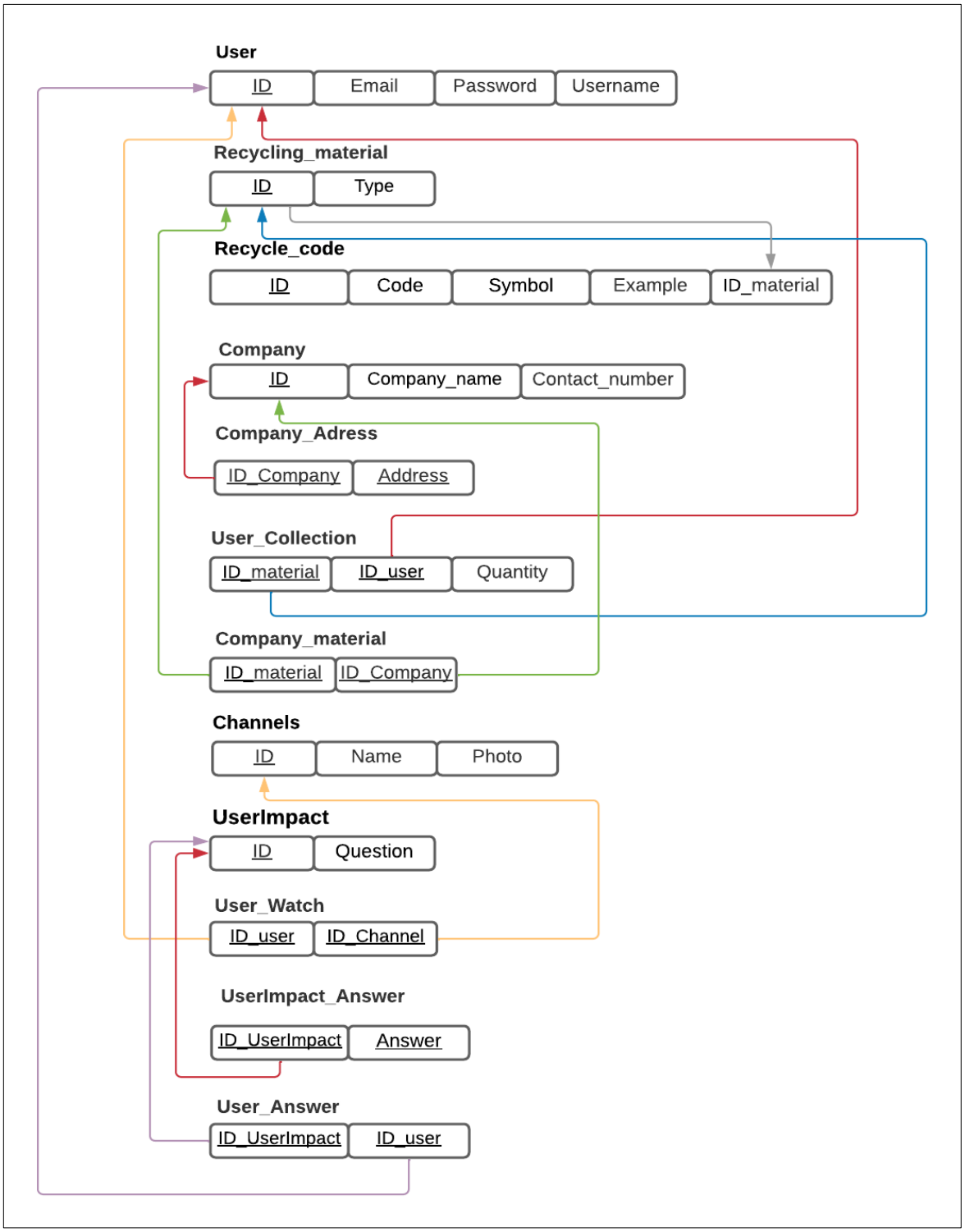


Figure 3.18: Schema relationship for Dawwerha application

3.1.2.4. Document Diagram

In the last semester, we had drawn the two models Schema and ER based on MySQL, but in this semester, we decided to change it to Firebase and because of that, we had to modify the models because they are express the relationships in SQL, but Firebase is considered to be unrelational or in another term NoSQL, so we searched for models that express this type which are 4 types Document, Graph, Key-Value, and Wide Column stores. Firebase is under the Document database.[38]

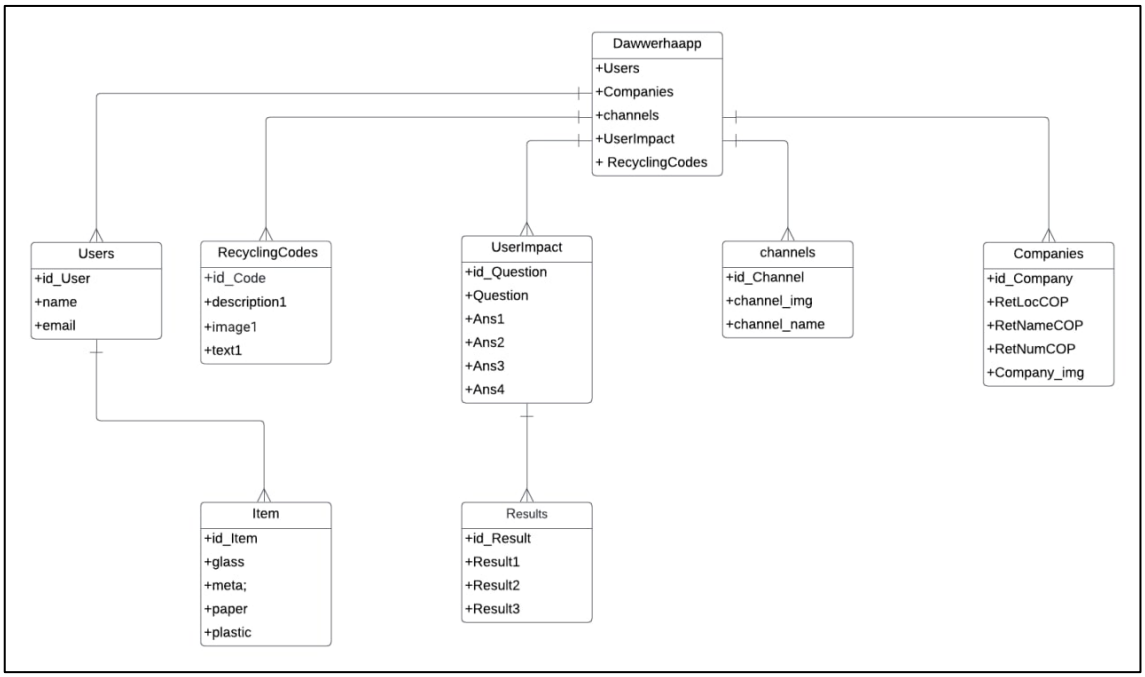


Figure 3.19: Document Diagram for Dawwerha application

3.1.3. Behavioral Models

Describes the internal dynamic aspects of the information system that supports the business processes in the organization. [18]

3.1.3.1. Activity Diagram

Activity diagram is a diagram that looks like a horizontal flow chart that shows the actions and events as they occur, additionally shows the order wherein actions take place and identify the out-come. [16]

Figures 3.19 – 3.32 represents all the activity diagram for Dawwerha application.

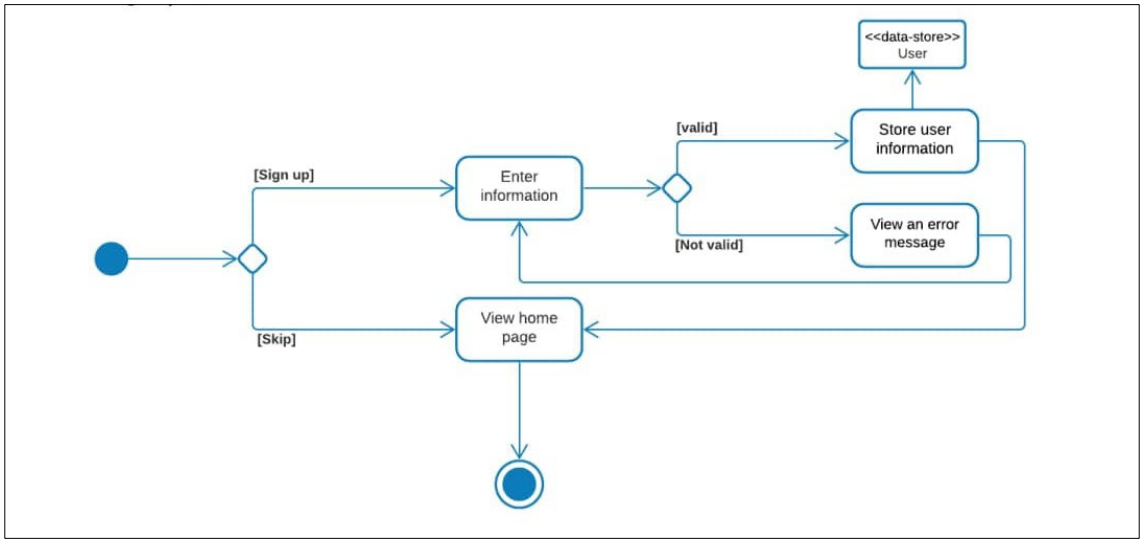


Figure 3.20: Activity Diagram for Sign up

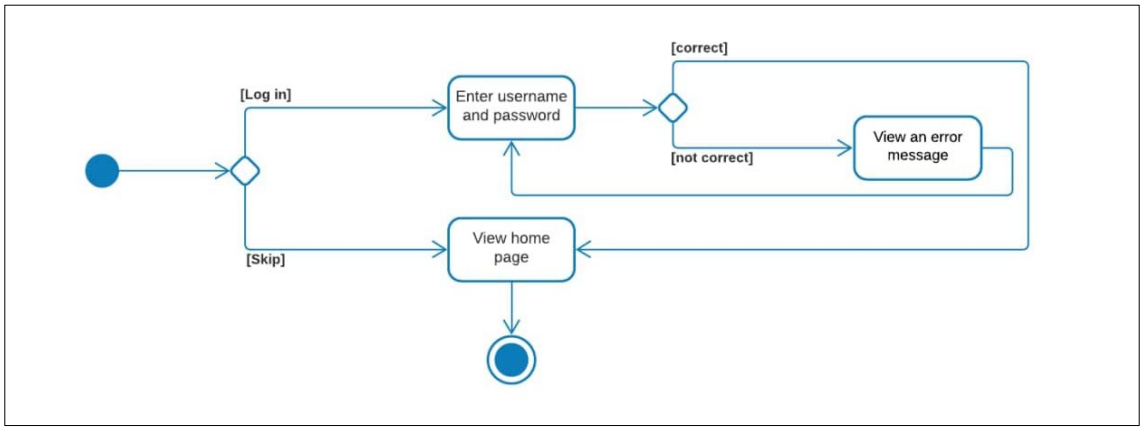


Figure 3.21: Activity Diagram for Log in

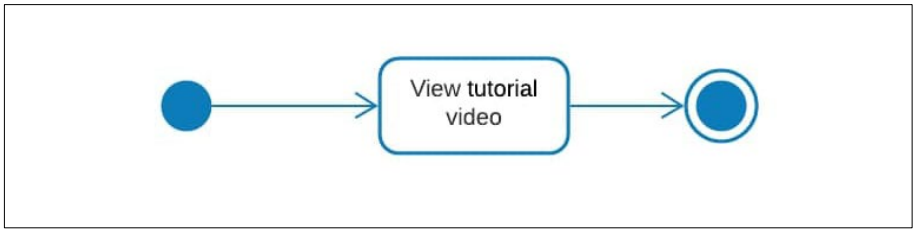


Figure 3.22: Activity Diagram for Tutorial

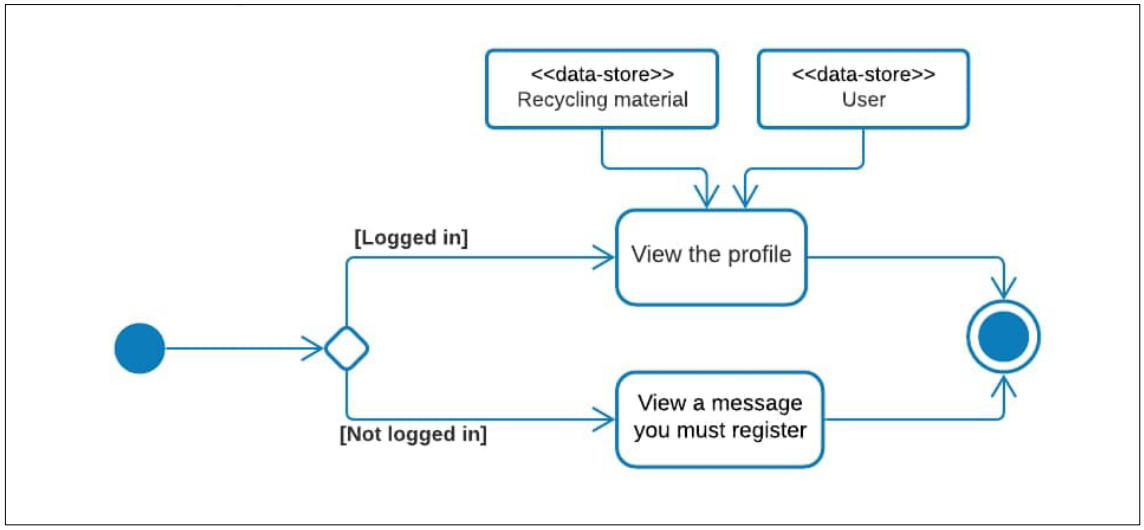


Figure 3.23: Activity Diagram for View user's profile

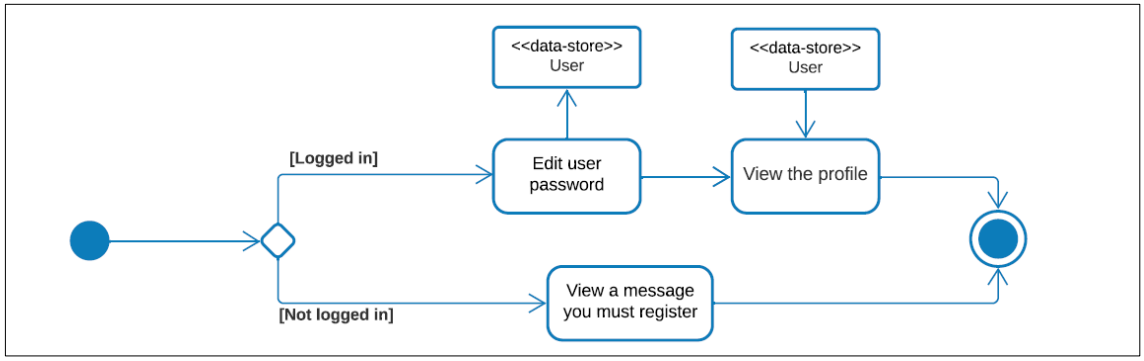


Figure 3.24: Activity Diagram for Change password

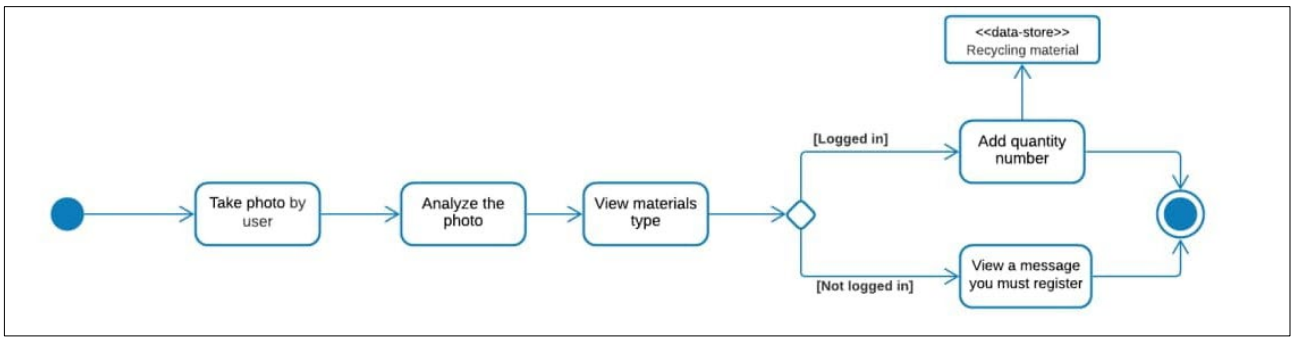


Figure 3.25: Activity Diagram for Take photo

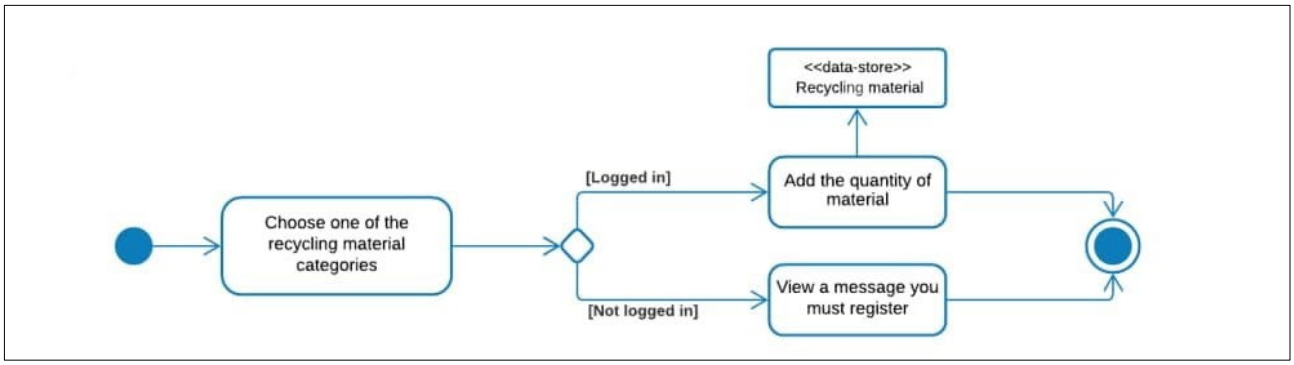


Figure 3.26: Activity Diagram for Recycling materials categories

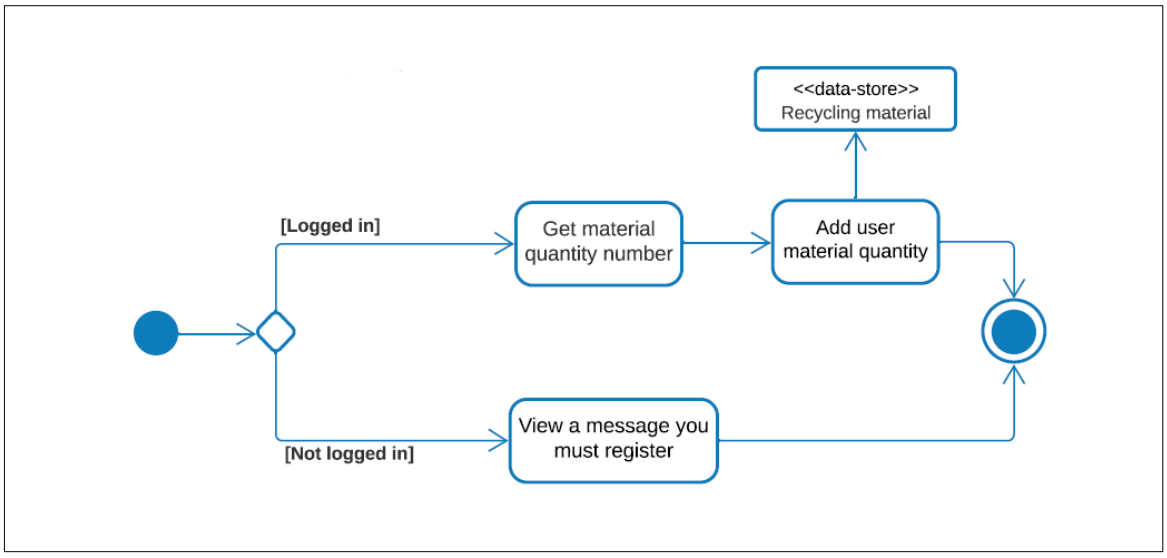


Figure 3.27: Activity Diagram for Add material quantity

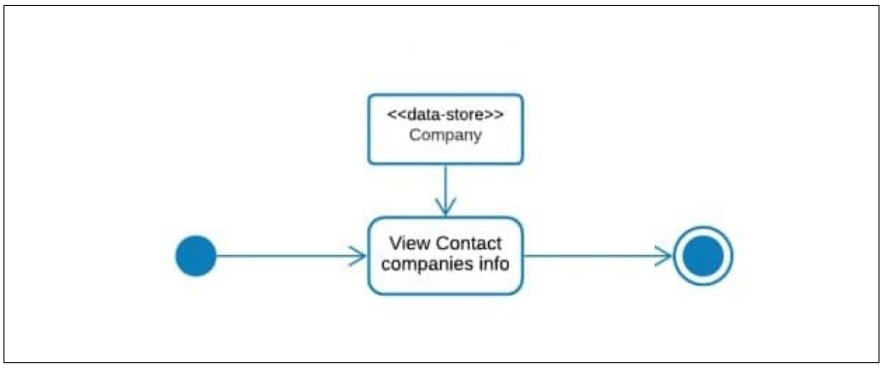


Figure 3.28: Activity Diagram for View companies contact

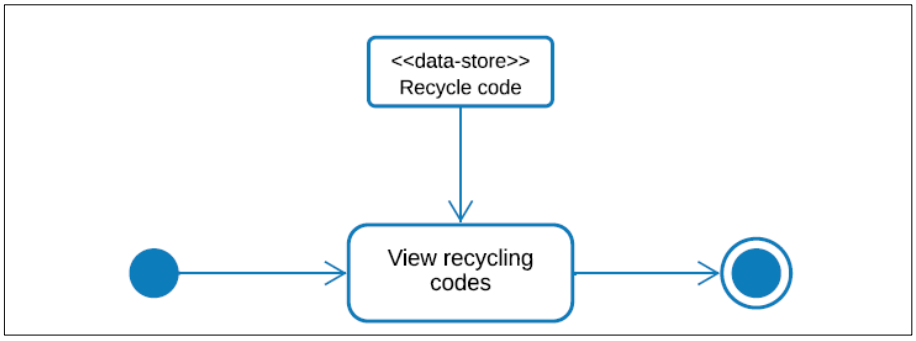


Figure 3.29: Activity Diagram for Recycling codes

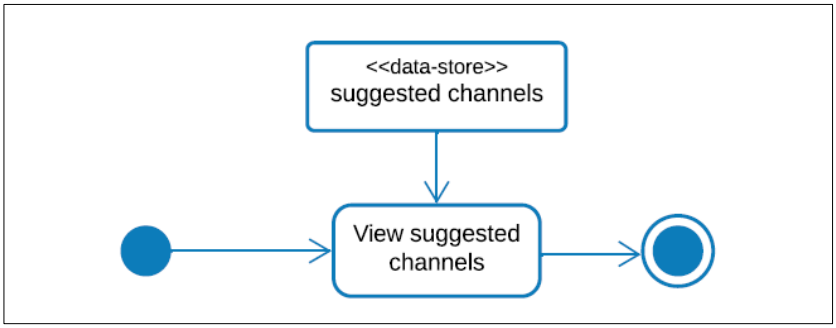


Figure 3.30: Activity Diagram for Suggested channels

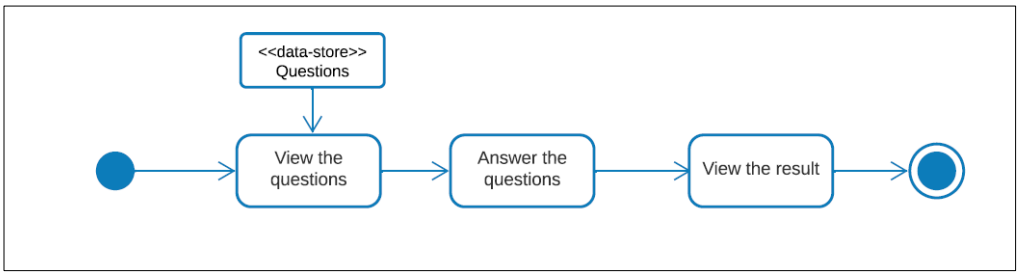


Figure 3.31: Activity Diagram for Find out user environmental impact

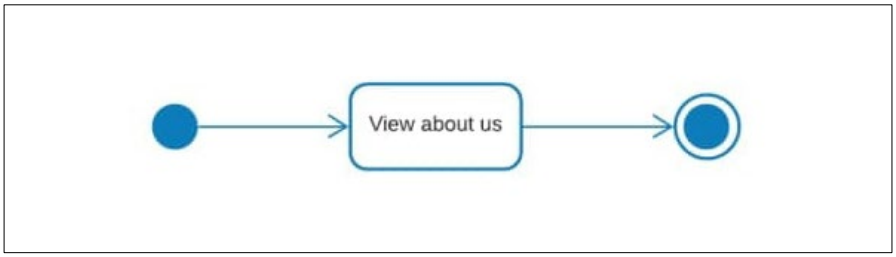


Figure 3.32: Activity Diagram for View about us

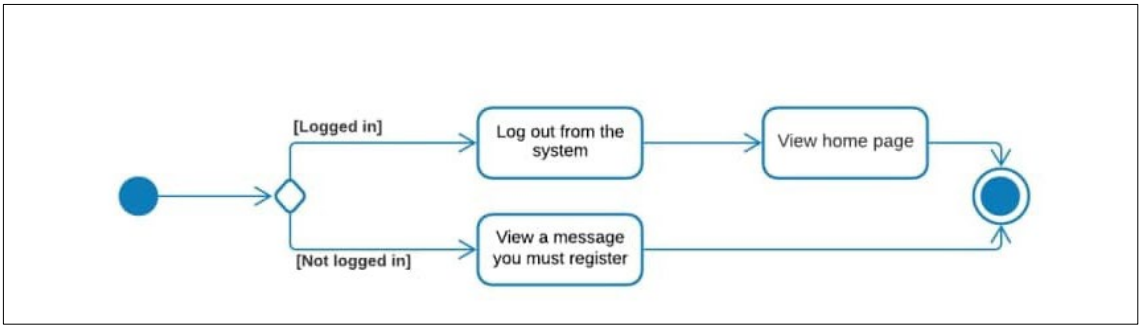


Figure 3.33: Activity Diagram for Log out

3.1.4. Architecture Design

Focuses on understanding how a software system should organize and design the overall structure of that system. [17]

3.1.4.1. MVC Pattern

Is a software design pattern for creating user interfaces, it divides the program into three interrelated component models, views and controller. [17]

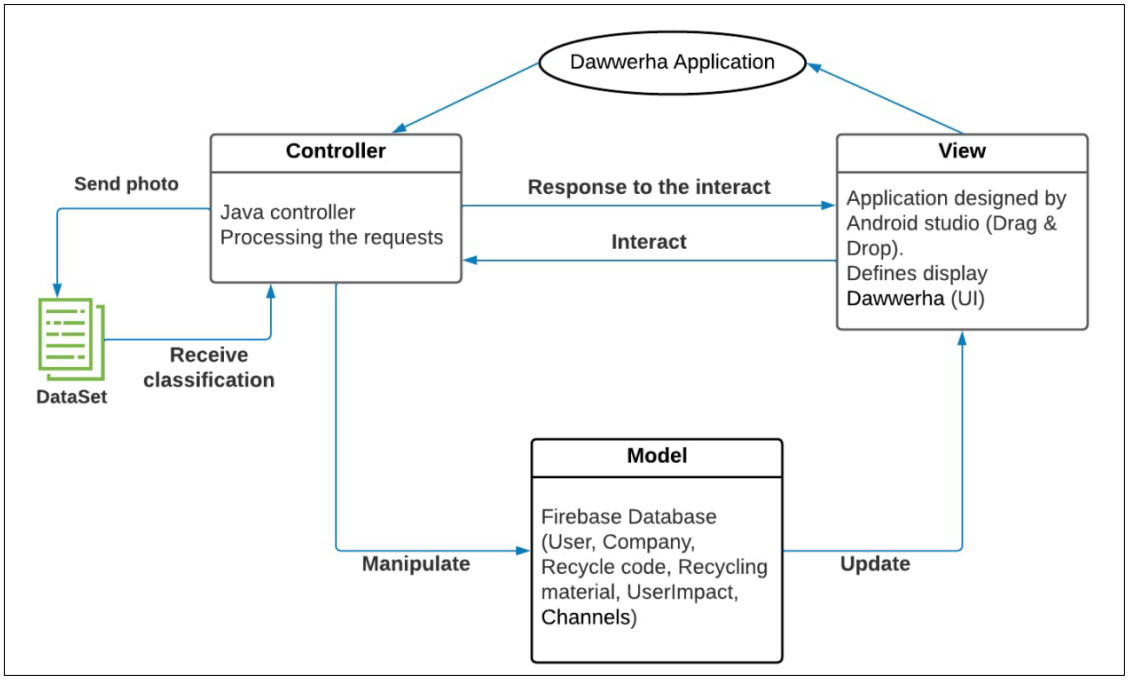


Figure 3.34: MVC Pattern for Dawwerha application

3.2. Design constraints

A summary of conditions that specify how to build and implement the required system.

3.2.1. Hardware and Software Environment

All necessary hardware and software environments relating to the customized product and the services.

3.2.1.1. Hardware Environment

The application will be available on a Smart phones with Android system version 7.0 Nougat and above.

3.2.1.2. Software Environment

To implement our application will use some programs and tools for:

- Application Type: Server client application.
- Implementing: Android Studio (Java), jupyter (python).
- Machine learning: Teachable Machine (tool), TensorFlow light (library).
- Designing diagrams: Lucidchart.
- Designing prototype for interfaces: proto.io.
- Designing application images: Adobe illustrator.
- Database management system: Firebase.

CHAPTER 4: Prototype

This chapter will cover the system prototype.

4.1. Prototype

Is an early version of a software system that is used to demonstrate concepts and can be used to help anticipate changes that may be required. [15]

Figures 4.1 – 4.19 represents all the Prototype interfaces work sequence in the application.

Figure 4.1: Opening Interface

Figure 4.2: Welcome Interface

Figure 4.3: Sign up Interface

Figure 4.6: Homepage Interface

Figure 4.4: Log in Interface

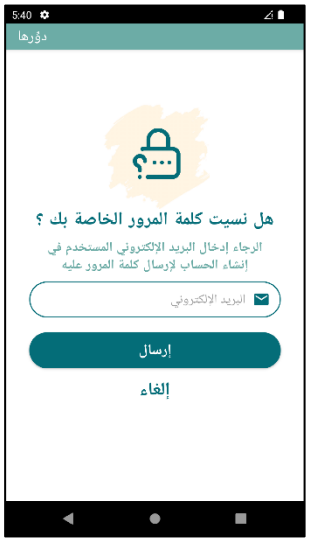


Figure 4.5: Forget password Interface

Figure 4.6: Homepage Interface

Figure 4.12: Recycling Companies Contacts Interface

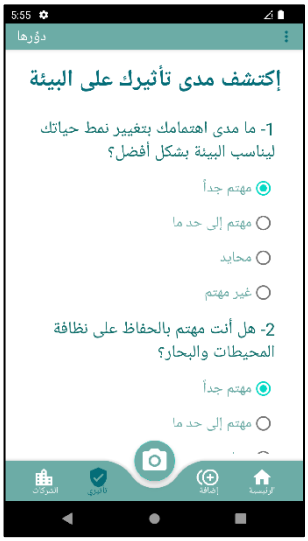


Figure 4.10: Impact on the environment Interface

Figure 4.8: Take a photo Interface

Figure 4.7: Select material Interface

Figure 4.11: Sample Result Interface

Figure 4.9: Add material Interface

Figure 4.6: Homepage Interface

Figure 4.13: Homepage list Interface

Figure 4.18: Tutorial Interface

Figure 4.17: Recycling codes Interface

Figure 4.16: Suggested channels Interface

Figure 4.14: View profile Interface

Figure 4.19: About us Interface

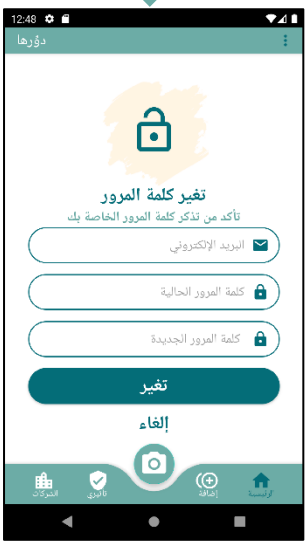


Figure 4.15: Change password Interface

CHAPTER 5: Implementation

This chapter will cover the implementation phase which including the application codes for the main functionality, database implementation and machine learning implementation.

5.1 Dawwerha App Codes

It describes the main functionalities which built the application.

5.1.1 Sign up

- This is the main function in the registration process. If the user wants to sign up to the application he/she must fills in the required data.
  Figures 5.1 – 5.2 shows the conditions for sign up.

```
// create account after everything is correct (entries)
BtnSginup.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        TxtUserName = EdtUserName.getText().toString();
        TxtUserEmail = EdtUserEmail.getText().toString().trim();
        TxtUserPassword = EdtUserPassword.getText().toString().trim();
        TxtUserConPassword = EdtUserConPassword.getText().toString().trim();
        if (!TextUtils.isEmpty(TxtUserName)) { // check username not empty
            if (TxtUserName.matches(namePattern)) { // check username matches the pattern
                if (!TextUtils.isEmpty(TxtUserEmail)) { // check email not empty
                    if (TxtUserEmail.matches(emailPattern)) { // check email matches the pattern
                        if (!TextUtils.isEmpty(TxtUserPassword)) { // check password not empty
                            if (TxtUserPassword.matches(passwordPattern)) { // check password matches the pattern
                                if (!TextUtils.isEmpty(TxtUserConPassword)) { // check re-password not empty
                                    if (TxtUserConPassword.equals(TxtUserPassword)) { // check that password and re-password are equals
                                        SignUpUser(); // everything is correct (entries), call create account
                                    } else {
                                        EdtUserConPassword.setError("الرجاء التحقق من مطابقة كلمة المرور بتأكيد كلمة المرور!");
                                    }
                                } else {
                                    EdtUserConPassword.setError("الرجاء تعبئة حقل النص الخاص بتأكيد كلمة المرور!");
                                }
                            } else {
                                EdtUserPassword.setError("الرجاء تعبئة حقل النص الخاص بكلمة المرور بشكل صحيح 8 خانات مكونة من حروف إنجليزية صغيرة وكبيرة وأرقام!");
                            }
                            } else {
                        EdtUserPassword.setError("الرجاء تعبئة حقل النص الخاص بكلمة المرور!");
```

Figure 5.1: Dawwerha App Code for Sign up Conditions

```
                                } else {
                                    EdtUserPassword.setError("الرجاء تعبئة حقل النص الخاص بكلمة المرور!");
                                }
                                    } else {
                                EdtUserEmail.setError("الرجاء إدخال البريد الإلكتروني بشكل صحيح!");
                            }
                        } else {
                            EdtUserEmail.setError("الرجاء تعبئة حقل النص الخاص بالبريد الإلكتروني!");
                        }
                    } else {
                        EdtUserName.setError("الرجاء تعبئة حقل النص الخاص باسم المستخدم بحروف فقط!");
                    }
                    } else {
                        EdtUserName.setError("الرجاء تعبئة حقل النص الخاص باسم المستخدم!");
                    }
                }
            });
        }
```

Figure 5.2: Dawwerha App Code for Sign up Conditions

- If the user tried to sign up with a email that already exists, an error message will appear that the email is already exists as shown In Figure 5.3.

```
        // if the account is already exist
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            Toast.makeText( context: Sginup.this,  text: "هذا البريد الإلكتروني مستخدم مسبقاً,قم بتسجيل الدخول أو استخدم بريد الكتروني آخر", Toast.LENGTH_SHORT).show();
            LodProgressBar.setVisibility(View.INVISIBLE);
            BtnSginup.setVisibility(View.VISIBLE);
        }
    });
}
```

Figure 5.3: Dawwerha App Code for email already existed

- After the user enters the required data correctly, the material quantity nodes are created for the user in the Database, then the user will be moved to the home page.
  Figures 5.4 – 5.5 shows the sign up process.

```
    // create account method
    private void SignUpUser() {
        //progress bar
        LodProgressBar.setVisibility(View.VISIBLE);
        BtnSginup.setVisibility(View.INVISIBLE);
        mAuth.createUserWithEmailAndPassword(TxtUserEmail, TxtUserPassword).addOnSuccessListener(new OnSuccessListener<AuthResult>() {
            @Override
            public void onSuccess(AuthResult authResult) {
                //get the entries
                EdtUserName = findViewById(R.id.UserName_Sginup);
                EdtUserEmail = findViewById(R.id.UserEmail_Sginup);
                //make it string
                String name=EdtUserName.getText().toString();
                String email=EdtUserEmail.getText().toString();
                // save the entries to database
                firebaseDatabase = FirebaseDatabase.getInstance();
                String uid = FirebaseAuth.getInstance().getCurrentUser().getUid();
                databaseReference = firebaseDatabase.getReference( path: "Users").child(uid);
                databaseReferencetxt =  firebaseDatabasetxt.getInstance().getReferenceFromUrl("https://dawwerhaapp-default-rtdb.firebaseio.com/");
                databaseReferencename = databaseReferencetxt.child("Users").child(uid).child("name");
                databaseReference.child("name").setValue(name);
                databaseReferenceemail = databaseReferencetxt.child("Users").child(uid).child("email");
                databaseReference.child("email").setValue(email);
                // defaults quantities value
                databaseReference = firebaseDatabase.getReference( path: "Users").child(uid).child("Item");
                databaseReferencetxt =  firebaseDatabasetxt.getInstance().getReferenceFromUrl("https://dawwerhaapp-default-rtdb.firebaseio.com/");
                databaseReferencepaper = databaseReferencetxt.child("Users").child(uid).child("Item").child("paper");
                databaseReference.child("paper").setValue("0");
                databaseReferenceplastic = databaseReferencetxt.child("Users").child(uid).child("Item").child("plastic");
                databaseReference.child("plastic").setValue("0");
```

Figure 5.4: Dawwerha App Code for Sign up process

```
                databaseReferenceglass = databaseReferencetxt.child("Users").child(uid).child("Item").child("glass");
                databaseReference.child("glass").setValue("0");
                databaseReferencemetal = databaseReferencetxt.child("Users").child(uid).child("Item").child("metal");
                databaseReference.child("metal").setValue("0");
                Toast.makeText( context: Sginup.this,  text: "لقد تم إنشاء الحساب بنجاح!", Toast.LENGTH_SHORT).show();
                Intent intent = new Intent( packageContext: Sginup.this, MainActivity2.class);
                startActivity(intent);
                finish();
            }
```

Figure 5.5: Dawwerha App Code for Sign up process

- This code sends notifications to users every 10 days as shown in Figure 5.6.

```
    //Method makes the user receive notifications from the application
    @Override
    public void onMessageReceived(@NonNull RemoteMessage remoteMessage) {
        String title = remoteMessage.getNotification().getTitle();
        String text = remoteMessage.getNotification().getBody();
        final String CHANNEL_ID = "HEADS_UP_NOTIFICATION";
        NotificationChannel channel = new NotificationChannel(
                CHANNEL_ID,
                name: "Heads Up Notification",
                NotificationManager.IMPORTANCE_HIGH
        );
        getSystemService(NotificationManager.class).createNotificationChannel(channel);
        Notification.Builder notification =
                new Notification.Builder( context: this, CHANNEL_ID)
                        .setContentTitle(title)
                        .setContentText(text)
                        .setSmallIcon(R.drawable.ic_launcher_background)
                        .setAutoCancel(true);
        NotificationManagerCompat.from(this).notify( id: 1, notification.build());
        super.onMessageReceived(remoteMessage);
    }
}
```

Figure 5.6: Dawwerha App Code for sends notifications

5.1.2 Login

- This is the main function in the login process. If the user wants to login into the application he/she must fills in his/her email and password, then the user will be moved to the home page. Figures 5.7 – 5.8 shows the login process.

```
// Initialize Firebase Auth
mAuth = FirebaseAuth.getInstance();
BtnLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        TxtUserEmail = EdtUserEmail.getText().toString().trim();
        TxtUserPassword = EdtUserPassword.getText().toString().trim();

        if (!TextUtils.isEmpty(TxtUserEmail)) {
            if (TxtUserEmail.matches(emailPattern)) {
                if (!TextUtils.isEmpty(TxtUserPassword)) {
                    SignInUser();
                } else {
                    EdtUserPassword.setError("! الرجاء تعبئة حقل النص الخاص بكلمة المرور");
                }
            } else {
                EdtUserEmail.setError("! الرجاء إدخال البريد الإلكتروني بشكل صحيح");
            }
        } else {
            EdtUserEmail.setError("! الرجاء تعبئة حقل النص الخاص بالبريد الإلكتروني");
        }
    }
});
```

Figure 5.7: Dawwerha App Code for login process

```
//method for database
private void SignInUser() {
    LodProgressBar.setVisibility(View.VISIBLE);
    BtnLogin.setVisibility(View.INVISIBLE);

    mAuth.signInWithEmailAndPassword(TxtUserEmail, TxtUserPassword).addOnSuccessListener(new OnSuccessListener<AuthResult>() {
        @Override
        public void onSuccess(AuthResult authResult) {
            Toast.makeText( context: Login.this,  text: "! لقد تم تسجيل الدخول بنجاح", Toast.LENGTH_SHORT).show();
            Intent intent = new Intent( packageContext: Login.this, MainActivity2.class);
            startActivity(intent);
            finish();
        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            Toast.makeText( context: Login.this,  text: "! الرجاء التحقق من صحة البريد الإلكتروني أو كلمة المرور", Toast.LENGTH_SHORT).show();
            LodProgressBar.setVisibility(View.INVISIBLE);
            BtnLogin.setVisibility(View.VISIBLE);
        }
    });
}
////////////////////////////////////
```

Figure 5.8: Dawwerha App Code for login process

- If the user wants to explore the application without logging in or sign up, the user will click on the "تخطي"button, to be registered as a guest in the application, then the user will be moved to the home page as shown in Figure 5.9.

```
// If the "تخطي" button is clicked, the user will be taken to the home page
skip= findViewById(R.id.skip);
skip.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        TxtUserEmail = "guset@gmail.com";
        TxtUserPassword = "guset123";
        SignInUserr();

    }});
```

Figure 5.9: Dawwerha App Code for skip

5.1.3 Forget password

- If the user clicks on the " نسيت كلمة المرور " text, the user will be moved to the Forgot Password page as shown in Figure 5.10.

```
//If the " نسيت كلمة المرور " button is clicked, the user will be taken to the Forgot Password page
forget= findViewById(R.id.forget);
forget.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intentloadNewActivite =new Intent( packageContext: Login.this, ForegetPassword.class);
        startActivity(intentloadNewActivite);
    }});
```

Figure 5.10: Dawwerha App Code for Forget password

- After that, the user enters his/her email in the forgot password page to send a link to his/her email to create a new password as shown in Figure 5.11.

```
// click on this to reset password
private void resetPassword(){
  String email= forgetEmail.getText().toString().trim();
  //conditions of write the email and password
   if (!email.isEmpty()) {
       if (email.matches(emailPattern)) {
       } else {
          forgetEmail.setError("!الرجاء إدخال البريد الإلكتروني بشكل صحيح");
          forgetEmail.requestFocus();
          return;
       }
   } else {
      forgetEmail.setError("!الرجاء تعبئة حقل النص الخاص بالبريد الإلكتروني");
      forgetEmail.requestFocus();
      return;
   }
   //click on this to send Password Reset Email
  auth.sendPasswordResetEmail(email).addOnCompleteListener(new OnCompleteListener<Void>() {
     @Override
     public void onComplete(@NonNull Task<Void> task) {
       if(task.isSuccessful()){
           Toast.makeText( context: ForegetPassword.this, text: " !تم إرسال رابط الإسترجاع إلى البريد الإلكتروني بنجاح" , Toast.LENGTH_SHORT).show();
           LodProgressBar.setVisibility(View.VISIBLE);
           resetPasswordBut.setVisibility(View.INVISIBLE);
           Intent intent = new Intent( packageContext: ForegetPassword.this, Login.class);
           startActivity(intent);
           finish();
       } else {
           Toast.makeText( context: ForegetPassword.this, text: "!لا يوجد بريد إلكتروني مسجل في البرنامج يطابق هذا البريد" , Toast.LENGTH_SHORT).show();
       }
     }
```

Figure 5.11: Dawwerha App Code for Forget password

5.1.4 Tutorial

- This is the main function of the tutorial process, once the user click on the top-drop-down menu from the top bar and choose " كيف تستخدم تطبيق دورها " there is a video will demonstrate how to use the application as shown in Figure 5.12.

```
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        //to play the tutorial video
        View rootView = inflater.inflate(R.layout.fragment_tutorial, container,  attachToRoot: false);
        VideoView videoview = rootView.findViewById(R.id.videoView);
        String videoPath = "android.resource://" + getActivity().getPackageName()+ "/" + R.raw.logo;
        Uri uri =Uri.parse(videoPath);
        videoview.setVideoURI(uri);
        MediaController mediaController = new MediaController(getActivity());
        videoview.setMediaController(mediaController);
        mediaController.setAnchorView(videoview);
        videoview.start();
        ;
         return rootView;
    }
}
```

Figure 5.12: Dawwerha App Code for Tutorial

5.1.5 Profile

- This is the main function for viewing the profile, once the user clicks on the top-drop-down menu from the top bar and choose "الحساب الشخصي", the user information is retrieved from the database and shown to the user if the user is logged in, but if the user is not logged in or don't have an account, the user will be moved to the Welcome page
  Figures 5.13 – 5.16 shows the profile page.

```
// check if there is user account logged in
FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
if ((user.getUid()).equals("ZTfA2GzePrYydnRbBaFrqJXMlIF3"))  {
    Toast.makeText(getActivity(), text: "الرجاء إنشاء حساب للتمكن من استخدام الميزة!", Toast.LENGTH_SHORT).show();
    Intent intent = new Intent(getActivity(), Welcome.class);
    startActivity(intent);
} else {
    String uid =user.getUid();
    firebaseDatabasetxt = FirebaseDatabase.getInstance();
    databaseReferencename = firebaseDatabasetxt.getReference( path: "Users").child(uid).child("name");
    databaseReferenceemail= firebaseDatabasetxt.getReference( path: "Users").child(uid).child("email");
    //get user information
    get_Email();
    get_Name();
    get_name();
    databaseReferencePaper = firebaseDatabasetxt.getReference( path: "Users").child(uid).child("Item").child("paper");
    databaseReferenceGlass = firebaseDatabasetxt.getReference( path: "Users").child(uid).child("Item").child("glass");
    databaseReferenceMetal = firebaseDatabasetxt.getReference( path: "Users").child(uid).child("Item").child("metal");
    databaseReferencePlastic = firebaseDatabasetxt.getReference( path: "Users").child(uid).child("Item").child("plastic");
    //get user quantities
    get_paper();
    get_glass();
    get_metal();
    get_plastic();
}
userResult.setOnClickListener(this);
```

Figure 5.13: Dawwerha App Code for Profile

```
// display username1
private void get_Name() {
    databaseReferencename.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String value = snapshot.getValue(String.class);
            Name.setText(value);
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {}
    }); }

// display email
private void get_Email() {
    databaseReferenceemail.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String value = snapshot.getValue(String.class);
            Email.setText(value);
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {}
    }); }
```

Figure 5.14: Dawwerha App Code for Profile

```
// display paper quantities
private void get_paper() {
    databaseReferencePaper.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String value = snapshot.getValue(String.class);
            Paper.setText(value);
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {}
    }); }
// display glass quantities
private void get_glass() {
    databaseReferenceGlass.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String value = snapshot.getValue(String.class);
            Glass.setText(value);
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {}
    }); }
```

Figure 5.15: Dawwerha App Code for Profile

```
// display metal quantities
private void get_metal() {
    databaseReferenceMetal.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String value = snapshot.getValue(String.class);
            Metal.setText(value);
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {}
    }); }
// display plastic quantities
private void get_plastic() {
    databaseReferencePlastic.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String value = snapshot.getValue(String.class);
            Plastic.setText(value);
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {}
    }); }
```

Figure 5.16: Dawwerha App Code for Profile

5.1.6 Change password

- If the user clicks on the " تغيير كلمة المرور " button, the user will be moved to the change Password page as shown in Figure 5.17.

```java
// move to change password page
@Override
public void onClick(View view) {
    Fragment fragment = null;
    switch (view.getId()) {
        case R.id.change_password_id:
            fragment = new ChangePassword();
            replaceFragment(fragment);
            break;
    }
}
```

Figure 5.17: Dawwerha App Code for change Password

- After that, the user enters his/her email and the old and new password on the change password page, then a message appears that the password has been changed successfully as shown in Figure 5.18.

```java
// change  password using firebase re-authentication
private void changepassword(final String email,  String password) {
    LodProgressBar.setVisibility(View.VISIBLE);
    changepassword.setVisibility(View.INVISIBLE);
    FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
    // Get auth credentials from the user for re-authentication
    AuthCredential credential = EmailAuthProvider.getCredential(email, password); // Current Login Credentials
    // Prompt the user to re-provide their sign-in credentials
    user.reauthenticate(credential).addOnCompleteListener(new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> task) {
            Log.d( tag: "value",  msg: "User re-authenticated.");
            // Now change your password
            FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
            user.updatePassword(Newpassword.getText().toString()).addOnCompleteListener(new OnCompleteListener<Void>() {
                @Override
                public void onComplete(@NonNull Task<Void> task) {
                    if (task.isSuccessful()) {
                        Toast.makeText(getActivity(), text: "تمت تغيير كلمة المرور بنجاح!", Toast.LENGTH_LONG).show();
                        Intent intent = new Intent(getActivity(), MainActivity2.class);
                        startActivity(intent);
                    } else {
                        Toast.makeText(getActivity(), text: "لا يوجد ايميل مسجل في البرنامج يطابق هذا الايميل!", Toast.LENGTH_LONG).show();
                    }
                }
            });
        }
    });
```

Figure 5.18: Dawwerha App Code for change Password

5.1.7 Take photo

- This is the main function in the Take photo process. When the user clicks the camera icon, it will show the user the permissions for the application's use of the camera, after the photo is taken by the user then displayed on the camera page as shown in Figure 5.19.

```
//open the camera
ImageView open = (ImageView) findViewById(R.id.open);
open.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Launch camera if we have permission
        if (checkSelfPermission(Manifest.permission.CAMERA) == PackageManager.PERMISSION_GRANTED) {
            Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
            startActivityForResult(cameraIntent, requestCode: 1);
        } else {
            //Request camera permission if we don't have it.
            requestPermissions(new String[]{Manifest.permission.CAMERA}, requestCode: 100);
        }
    }

});
```

Figure 5.19: Dawwerha App Code for Take photo

- This code classifies the image taken by the user based on the analysis of the image. Figures 5.20 – 5.21 shows the classification process in the take phot page.

```
// classify image using model
public void classifyImage(Bitmap image){
    try {
        Model model = Model.newInstance(getApplicationContext());
        // Creates inputs for reference.
        TensorBuffer inputFeature0 = TensorBuffer.createFixedSize(new int[]{1, 224, 224, 3}, DataType.FLOAT32);
        ByteBuffer byteBuffer = ByteBuffer.allocateDirect(4 * imageSize * imageSize * 3);
        byteBuffer.order(ByteOrder.nativeOrder());
        // get 1D array of 224 * 224 pixels in image
        int [] intValues = new int[imageSize * imageSize];
        image.getPixels(intValues, offset: 0, image.getWidth(), x: 0, y: 0, image.getWidth(), image.getHeight());
        // iterate over pixels and extract R, G, and B values. Add to bytebuffer.
        int pixel = 0;
        for(int i = 0; i < imageSize; i++){
            for(int j = 0; j < imageSize; j++){
                int val = intValues[pixel++]; // RGB
                byteBuffer.putFloat(((val >> 16) & 0xFF) * (1.f / 255.f));
                byteBuffer.putFloat(((val >> 8) & 0xFF) * (1.f / 255.f));
                byteBuffer.putFloat((val & 0xFF) * (1.f / 255.f));
            } }
        inputFeature0.loadBuffer(byteBuffer);
        // Runs model inference and gets result.
        Model.Outputs outputs = model.process(inputFeature0);
        TensorBuffer outputFeature0 = outputs.getOutputFeature0AsTensorBuffer();
        float[] confidences = outputFeature0.getFloatArray();
        // find the index of the class with the biggest confidence.
        int maxPos = 0;
        float maxConfidence = 0;
        for(int i = 0; i < confidences.length; i++){
            if(confidences[i] > maxConfidence){
```

Figure 5.20: Dawwerha App Code for classify

```
        for(int i = 0; i < confidences.length; i++){
            if(confidences[i] > maxConfidence){
                maxConfidence = confidences[i];
                maxPos = i;
            }
        }
        // Model classification sections
        String[] classes = {"زجاج", "معدن", "ورق", "بلاستيك"};
        System.out.print(": التصنيف هو");
        result.setText(classes[maxPos]);
        // Releases model resources if no longer used.
        model.close();
    } catch (IOException e) {
        // TODO Handle the exception
    }
}
```

Figure 5.21: Dawwerha App Code for classify

- This code check if the user is logged in then allows the user to add the quantity of the classified item, then the item quantity is updated and stored in the database, but if the user is not logged in or don't have an account, the user will be moved to the welcome page.
  Figures 5.22 – 5.24 shows the processes after classify the photo.

```
// add quantity per user in Firebase
ADDitem.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
        if ((user.getUid()).equals("ZTfA2GzePrYydnRbBaFrqJXMlIF3")) {
            Toast.makeText( context: Camera.this, text: "الرجاء إنشاء حساب للتمكن من استخدام الميزة!", Toast.LENGTH_SHORT).show();
            Intent intent = new Intent( packageContext: Camera.this, Welcome.class);
            startActivity(intent);
            /* Retrieve the quantity from Firebase and add
             the new quantity to the previous quantity and update it in Firebase*/
        }else if (result.getText().toString() == "ورق"){
            //paper
            String paperdata = value.getText().toString();
            int paperint = parseInt(paperdata);
            paperint = paperint + getvaluepaper;
            paperdata = Integer.toString(paperint);
            databaseReference.child("paper").setValue(paperdata);
            Toast.makeText( context: Camera.this, text: "تم إضافة الكمية بنجاح!", Toast.LENGTH_SHORT).show();
        }
        else if (result.getText().toString() == "بلاستيك"){
            //plastic
            String plasticdata = value.getText().toString();
            int plasticint = parseInt(plasticdata);
            plasticint = plasticint + getvalueplastic;
            plasticdata = Integer.toString(plasticint);
            databaseReference.child("plastic").setValue(plasticdata);
            Toast.makeText( context: Camera.this, text: "تم إضافة الكمية بنجاح!", Toast.LENGTH_SHORT).show();
        }
```

Figure 5.22: Dawwerha App Code for add quantity after take photo

```
        else if (result.getText().toString() == "زجاج"){
            //glass
            String glassdata = value.getText().toString();
            int glassint = parseInt(glassdata);
            glassint = glassint + getvalueglass;
            glassdata = Integer.toString(glassint);
            databaseReference.child("glass").setValue(glassdata);
            Toast.makeText( context: Camera.this,  text: "!تم إضافة الكمية بنجاح", Toast.LENGTH_SHORT).show();
        }
        else if (result.getText().toString() == "معدن"){
            //metal
            String metaldata = value.getText().toString();
            int metalint = parseInt(metaldata);
            metalint = metalint + getvaluemetal;
            metaldata = Integer.toString(metalint);
            databaseReference.child("metal").setValue(metaldata);
            Toast.makeText( context: Camera.this,  text: "!تم إضافة الكمية بنجاح", Toast.LENGTH_SHORT).show();
        }
        else {
            Toast.makeText( context: Camera.this,  text: "!يجب عليك إلتقاط صورة أولا", Toast.LENGTH_SHORT).show();
        }
    }
});
```

Figure 5.23: Dawwerha App Code for add quantity after take photo

```
//retrieve the glass from firebase
private void getdataglass() {
    databaseReferenceglass.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String value = snapshot.getValue(String.class);
            getvalueglass = parseInt(value);
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {
        }
    });
}
```

Figure 5.24: Dawwerha App Code for retrieve quantity

- If the user face a problem in classifying the image, he/she will click on the "مساعدة"icon, then instructions will appear to help the user on how to take photo in right way as shown in Figure5.25.

```
//###Start pop page###//
HintCamera = findViewById(R.id.Camera_hint);

//Create the Dialog here
dialog = new Dialog( context: this);
dialog.setContentView(R.layout.camera_dialog);
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
    dialog.getWindow().setBackgroundDrawable(getDrawable(R.drawable.hint_camera));
}
dialog.getWindow().setLayout(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT);
dialog.setCancelable(false); //Optional
dialog.getWindow().getAttributes().windowAnimations = R.style.DialogAnimation; //Setting the animations to dialog

Button Okay = dialog.findViewById(R.id.btn_okay);

Okay.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) { dialog.dismiss(); }
});

HintCamera.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) { dialog.show(); // Showing the dialog here }
});
//###End pop page###//
```

Figure 5.25: Dawwerha App Code for help

5.1.8 Add material quantity

- This is the main function of the add material quantity, when the user clicks "إضافة" from the bottom bar of the application, it checks if the user is logged in then allows the user to add the pieces of materials, updated and stored it in the database, but if the user is not logged in or don't have an account, the user will be moved to the welcome page.
  Figures 5.26 – 5.28 shows the add material quantity process.

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
    View rootView = inflater.inflate(R.layout.fragment_additem, container,  attachToRoot: false);


    paper=(TextView) rootView.findViewById(R.id.paper);
    plastic=(TextView) rootView.findViewById(R.id.plastic);
    glass=(TextView) rootView.findViewById(R.id.glass);
    metal=(TextView) rootView.findViewById(R.id.metal);
    ADDitem =rootView.findViewById(R.id.AddBtn_itms);
    FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
    if ((user.getUid()).equals("ZTfA2GzePrYydnRbBaFrqJXMlIF3"))  {
        Toast.makeText(getActivity(),  text: "الرجاء إنشاء حساب للتمكن من استخدام العيزة!", Toast.LENGTH_SHORT).show();
        Intent intent = new Intent(getActivity(), Welcome.class);
        startActivity(intent);
    }else {
        String uid = FirebaseAuth.getInstance().getCurrentUser().getUid();
        firebaseDatabase = FirebaseDatabase.getInstance();
        databaseReference = firebaseDatabase.getReference( path: "Users").child(uid).child("Item");
        databaseReferencetxt = firebaseDatabasetxt.getInstance().getReferenceFromUrl("https://dawwerhaapp-default-rtdb.firebaseio.com/");
        databaseReferencepaper = databaseReferencetxt.child("Users").child(uid).child("Item").child("paper");
        databaseReferenceplastic = databaseReferencetxt.child("Users").child(uid).child("Item").child("plastic");
        databaseReferenceglass = databaseReferencetxt.child("Users").child(uid).child("Item").child("glass");
        databaseReferencemetal = databaseReferencetxt.child("Users").child(uid).child("Item").child("metal");
        getdatapaper();
        getdataplastic();
        getdataglass();
        getdatametal();
    }
```

Figure 5.26: Dawwerha App Code for add material quantity

```
//
ADDitem.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //add paper in firebase
        String paperdata = paper.getText().toString();
        int paperint = parseInt(paperdata);
        paperint = paperint + getvaluepaper;
        paperdata = Integer.toString(paperint);
        databaseReference.child("paper").setValue(paperdata);
        //add plastic in firebase
        String plasticdata = plastic.getText().toString();
        int plasticint = parseInt(plasticdata);
        plasticint = plasticint + getvalueplastic;
        plasticdata = Integer.toString(plasticint);
        databaseReference.child("plastic").setValue(plasticdata);
        //add glass in firebase
        String glassdata = glass.getText().toString();
        int glassint = parseInt(glassdata);
        glassint = glassint + getvalueglass;
        glassdata = Integer.toString(glassint);
        databaseReference.child("glass").setValue(glassdata);
        //add metal in firebase
        String metaldata = metal.getText().toString();
        int metalint = parseInt(metaldata);
        metalint = metalint + getvaluemetal;
        metaldata = Integer.toString(metalint);
        databaseReference.child("metal").setValue(metaldata);
        Toast.makeText(getActivity(), text: "تمت إضافة الكمية بنجاح!", Toast.LENGTH_LONG).show();
    }
});
```

Figure 5.27: Dawwerha App Code for add material quantity

```
//retrieve the plastic from firebase
private void getdataplastic() {
    databaseReferenceplastic.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String value = snapshot.getValue(String.class);
            getvalueplastic = parseInt(value);
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {
            Toast.makeText(getActivity(), text: "خطأ في استرجاع المعلومات.", Toast.LENGTH_SHORT).show();
        }
    });
}
```

Figure 5.28: Dawwerha App Code for retrieve quantity

5.1.9 Companies contact information

- This is the main function of the companies contact information process, when the user clicks "الشركات" from the bottom bar of the application, the name, number and location of the recycling companies will be appears.
  Figures 5.29 – 5.30 shows the companies contact information page.

```
//Retrieve image Company 1 from firebase
getImage1.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        String link = dataSnapshot.getValue(String.class);
        Picasso.get().load(link).into(Companyimg1);
    }
    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {
    }
});
```

Figure 5.29: Dawwerha App Code for companies contact information

```
////Retrieve the name of the company 1 from the firebase
private void getdata1() {
    databaseReferencetxt1.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String value = snapshot.getValue(String.class);
            NameCOP1.setText(value); }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {
        }}); }
//Retrieve the number of the company 1 from the firebase
private void getdata11() {
    databaseReferenceNum1.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String value = snapshot.getValue(String.class);
            NumCOP1.setText(value); }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {
        }}); }
//Retrieve the location of the company 1 from the firebase
private void getdata111() {
    databaseReferenceLoc1.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String value = snapshot.getValue(String.class);
            LocCOP1.setText(value); }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {
        }}); }
```

Figure 5.30: Dawwerha App Code for companies contact information

5.1.10 Recycling codes

- This is the main function of the recycling codes process, once the user clicks on the top-drop-down menu from the top bar and clicks "رموز اعادة التدوير", the recycling codes are displayed with a description and example of each code.
  Figures 5.31 – 5.32 shows the recycling code page.

```
//get image1
getImage1.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        String link = dataSnapshot.getValue(String.class);
        Picasso.get().load(link).into(papercode1);
    }
    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {}
});
```

Figure 5.31: Dawwerha App Code for Recycling codes

```
//Retrieve code 1 and Description
private void getdatatext1() {
    databaseReferencetextpapercode1.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String value = snapshot.getValue(String.class);
            textpapercode1.setText(value);
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {}
    });
}
private void getdatadep1() {
    databaseReferencedescode1.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String value = snapshot.getValue(String.class);
            descode1.setText(value);
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {}
    });
}
```

Figure 5.32: Dawwerha App Code for Recycling codes

5.1.11 Suggested channels

- This is the main function of the suggested channels process, once the user clicks on the top-drop-down menu from the top bar and clicks "قنوات مقترحة لإعادة التدوير", suggested YouTube channels are displayed that help in the process of reusing materials at home.
  Figures 5.33 – 5.34 shows the suggested channels page.

```
//Retrieve the image of the first channel
getImage1.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        String link = dataSnapshot.getValue(String.class);
        Picasso.get().load(link).into(channel1_img);
    }
    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {
    }
});
```

Figure 5.33: Dawwerha App Code for Suggested channels

```
//Retrieve the name of the first channel
private void getdata1() {
    databaseReferenceChannel1.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String value = snapshot.getValue(String.class);
            channel1_name.setText(value);
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {
        }
    });
}
```

Figure 5.34: Dawwerha App Code for Suggested channels

5.1.12 User impact

- This is the main function of the user impact calculation, when the user clicks "تأثيري" from the bottom bar of the application, several questions will be display to the user then the user will answer all of them then the result of the user impact will be calculated.
  Figures 5.35 – 5.36 shows the user impact calculate prosses.

```
// calculate the result for the user after click result button
showResultButton = rootView.findViewById(R.id.showResultButton);
showResultButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (A11.isChecked()) {
            final_result = final_result + 100;
        } else if (A12.isChecked()) {
            final_result = final_result + 60;
        } else if (A13.isChecked()) {
            final_result = final_result + 20;
        } else if (A14.isChecked()) {
            final_result = final_result + 0;
        }
```

Figure 5.35: Dawwerha App Code for user impact calculation

```
//////////////////////Q10/////////////////
if (A101.isChecked()) {
    final_result = final_result + 100;
} else if (A102.isChecked()) {
    final_result = final_result + 60;
} else if (A103.isChecked()) {
    final_result = final_result + 20;
} else if (A104.isChecked()) {
    final_result = final_result + 0;
}
final_result = final_result / 10;
```

Figure 5.36: Dawwerha App Code for user impact calculation

- This code check that the user has answered all the questions, then the user will be moved to the result page based on his/her result.
  Figures 5.37 – 5.38 shows the result page.

```
//check that the user answer all the questions
if ((A11.isChecked() || A12.isChecked() || A13.isChecked() || A14.isChecked()) &&
        (A21.isChecked() || A22.isChecked() || A23.isChecked() || A24.isChecked()) &&
        (A31.isChecked() || A32.isChecked() || A33.isChecked() || A34.isChecked()) &&
        (A41.isChecked() || A42.isChecked() || A43.isChecked() || A44.isChecked()) &&
        (A51.isChecked() || A52.isChecked() || A53.isChecked() || A54.isChecked()) &&
        (A61.isChecked() || A62.isChecked() || A63.isChecked() || A64.isChecked()) &&
        (A71.isChecked() || A72.isChecked() || A73.isChecked() || A74.isChecked()) &&
        (A81.isChecked() || A82.isChecked() || A83.isChecked() || A84.isChecked()) &&
        (A91.isChecked() || A92.isChecked() || A93.isChecked() || A94.isChecked()) &&
        (A101.isChecked() || A102.isChecked() || A103.isChecked() || A104.isChecked())) {
    // best User Impact result
    if (final_result >= 70) {
        Bundle bundle=new Bundle();
        bundle.putString("Key1",final_result+"%");
        resetResult();
        UserResult fragment =new UserResult();
        fragment.setArguments(bundle);
        getFragmentManager().beginTransaction().replace(R.id.container,fragment).commit();
    // middle User Impact result
    } else if (final_result < 70 && final_result >= 40) {
        Bundle bundle=new Bundle();
        bundle.putString("Key2",final_result+"%");
        resetResult();
        UserResult2 fragment =new UserResult2();
        fragment.setArguments(bundle);
        getFragmentManager().beginTransaction().replace(R.id.container,fragment).commit();
    // worst User Impact result
    } else if (final_result < 40) {
        Bundle bundle=new Bundle();
        bundle.putString("Key3",final_result+"%");
```

Figure 5.37: Dawwerha App Code for user impact

```
        } else if (final_result < 40) {
            Bundle bundle=new Bundle();
            bundle.putString("Key3",final_result+"%");
            resetResult();
            UserResult3 fragment =new UserResult3();
            fragment.setArguments(bundle);
            getFragmentManager().beginTransaction().replace(R.id.container,fragment).commit();
        }
    // the user not answer all the questions
    } else {
        Toast.makeText(getActivity(), text: "الرجاء التحقق من اختيار إجابة لجميع الأسئلة !", Toast.LENGTH_SHORT).show();
    }
}

});
```

Figure 5.38: Dawwerha App Code for user impact

- This code retrieves questions and their answers from the Database as shown in Figure 5.39.

```
//////////////////////////////////////Retrieve Question1 with its Answers//////////////////////////////////////
private void getdataQ1() {
    databaseReferenceQ1.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String value = snapshot.getValue(String.class);
            Q1.setText(value);
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {}
    });
}
private void getdataA11() {
    databaseReferenceA11.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String value = snapshot.getValue(String.class);
            A11.setText(value);
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {}
    });
}
private void getdataA12() {
    databaseReferenceA12.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String value = snapshot.getValue(String.class);
            A12.setText(value);
        }
```

Figure 5.39: Dawwerha App Code for retrieves questions

5.1.13 Logout

- This is the main function of the logout process. It is accessed from the profile page after clicking on the logout icon, the user will be moved to the welcome page as shown in Figure 5.40.

```
// log out
Exit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(getActivity(), text: "!لقد تم تسجيل الخروج بنجاح", Toast.LENGTH_SHORT).show();
        FirebaseAuth.getInstance().signOut();
        Intent intent = new Intent(getActivity(), Welcome.class);
        startActivity(intent);
    }
});
```

Figure 5.40: Dawwerha App Code for logout

5.1.14 Bottom bar

- This code displays the bottom bar that allows the user to navigate between the following application pages "الرئيسية - إضافة - تأثيري - الشركات" as shown in Figure 5.41.

```
//Bottom bar transfers
  @Override
  public boolean onNavigationItemSelected(@NonNull MenuItem item) {

      switch (item.getItemId()) {
          case R.id.impact:
              getSupportFragmentManager().beginTransaction().replace(R.id.container, userImpact).commit();
              return true;

          case R.id.home:
              getSupportFragmentManager().beginTransaction().replace(R.id.container, homePage).commit();
              return true;

          case R.id.select:
              getSupportFragmentManager().beginTransaction().replace(R.id.container, additems).commit();
              return true;

          case R.id.compny:
              getSupportFragmentManager().beginTransaction().replace(R.id.container, companyInfo).commit();
              return true;


      }
      return false;
  }
```

Figure 5.41: Dawwerha App Code for bottom bar

5.1.15 Top-drop-down menu

- This code displays the top-drop-down menu that allows the user to navigate between the following application pages: " الحساب الشخصي - قنوات مقترحة لإعادة التدوير - رموز اعادة التدوير - كيف تستخدم تطبيق دورها - من نحن " as shown in Figure 5.42.

```
//Top bar transfers
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.profile:
            getSupportFragmentManager().beginTransaction().replace(R.id.container, profile).commit();
            return true;

        case R.id.idea:
            getSupportFragmentManager().beginTransaction().replace(R.id.container, suggestedchannels).commit();
            return true;

        case R.id.learnmore:
            getSupportFragmentManager().beginTransaction().replace(R.id.container,recyclingcodes).commit();
            return true;

        case R.id.tutorial:
            getSupportFragmentManager().beginTransaction().replace(R.id.container, tutorial).commit();
            return true;

        case R.id.aboutus:
            getSupportFragmentManager().beginTransaction().replace(R.id.container, aboutUs).commit();
            return true;
    }
    return false;
}
```

Figure 5.42: Dawwerha App Code for top-drop-down menu

5.2 Database Implementation

Firebase is Google-backed application development software that allows developers to build Android, IOS, and web applications Firebase Database was chosen as the management system for our application because it easily and quickly exchanges the data, provide developers with a variety of tools and services to help them build high-quality applications and it is support storing images.

Firebase has a wide range of services, and in our application, we used four services, which are Authentication, Real-time, Storage and Cloud messaging.

- Firebase Authentication is a firebase service that provides libraries to authenticate users in the application. It is used to authenticate the user at sign-up/login in our application as shown in Figure 5.43.[35][36]

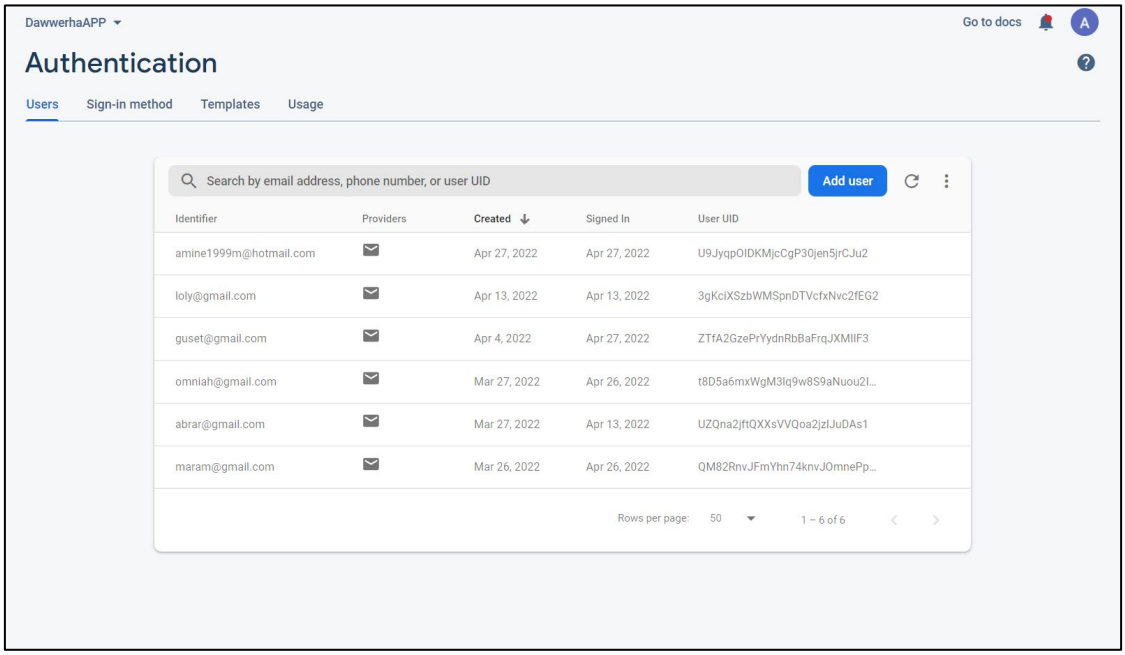


Figure 5.43: Database Firebase Authentication contains the email and password for every user

- Firebase Real-time Database that is synchronized in real time for all users, even when the application is offline. It is used to store and retrieve data in the application as shown in Figures 5.44 - 5.49. [35][36]

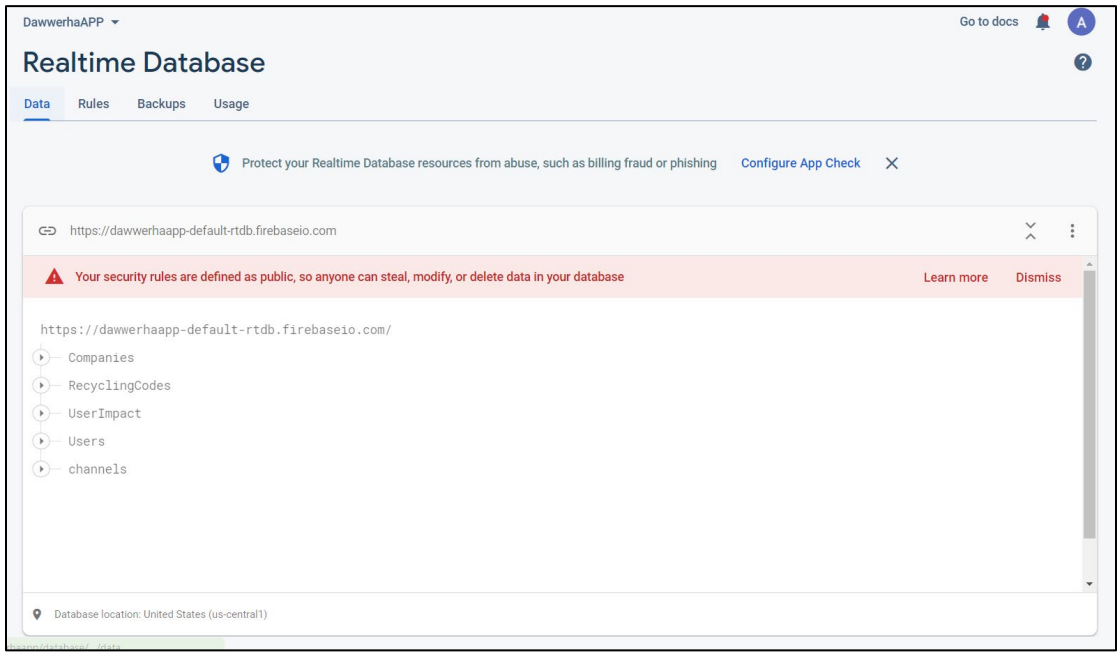


Figure 5.44: Firebase Real-time Database contains the parent five nodes

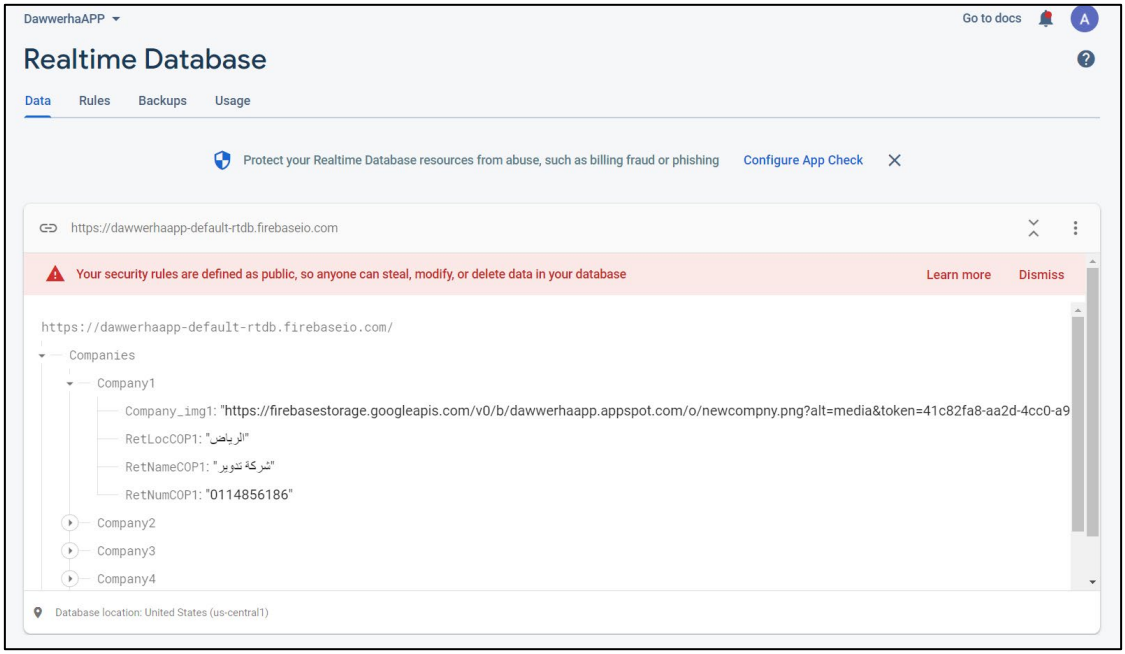


Figure 5.45: Firebase Real-time Database for the content of Companies node

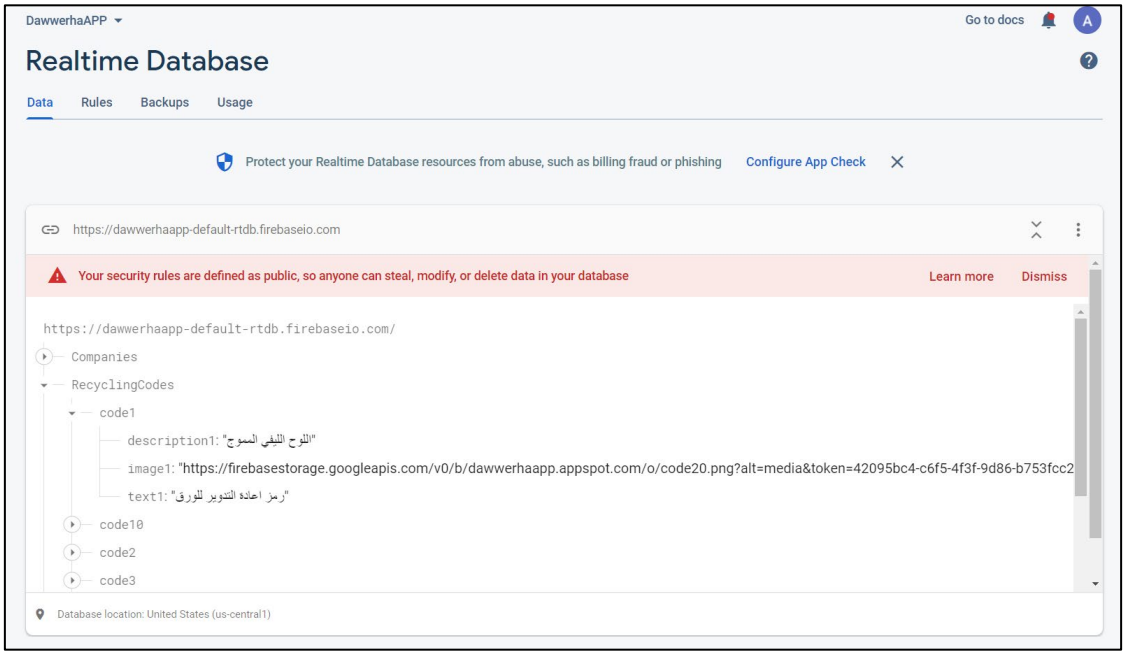


Figure 5.46: Firebase Real-time Database for the content of Recycling Codes node

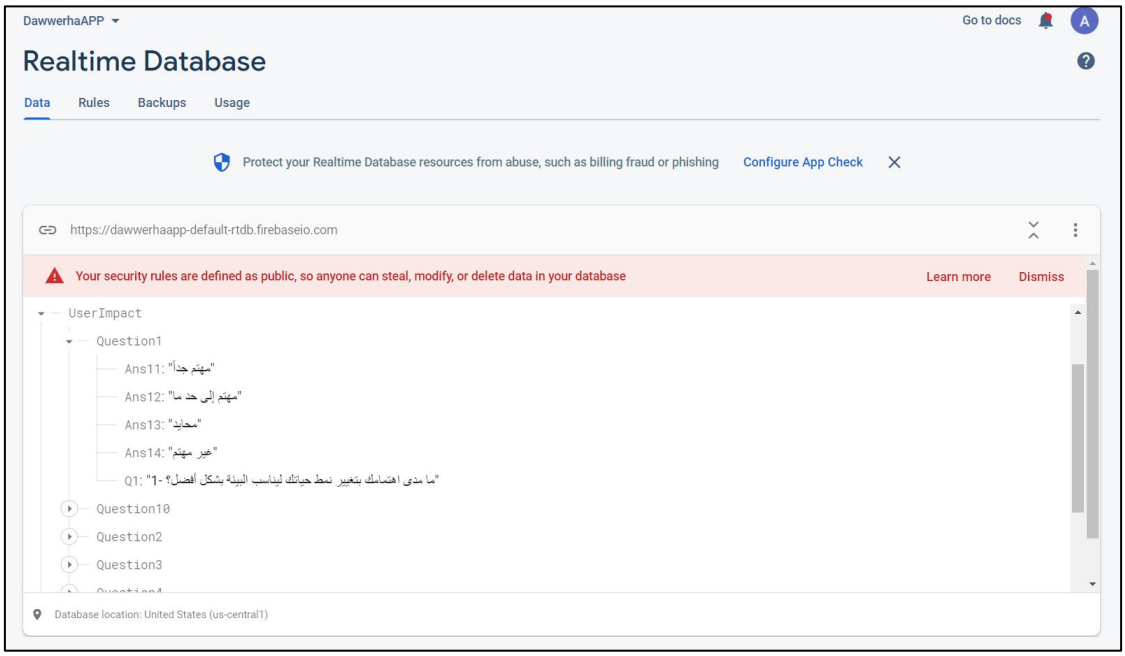


Figure 5.47: Firebase Real-time Database for the content of User Impact node

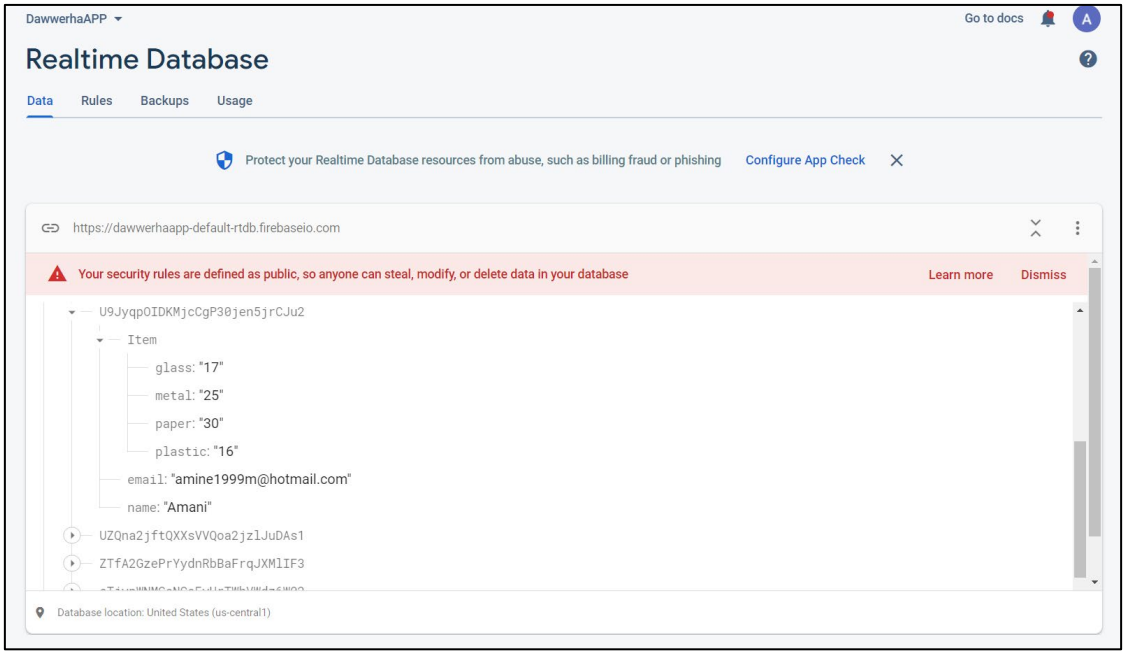


Figure 5.48: Firebase Real-time Database for the content of Users node

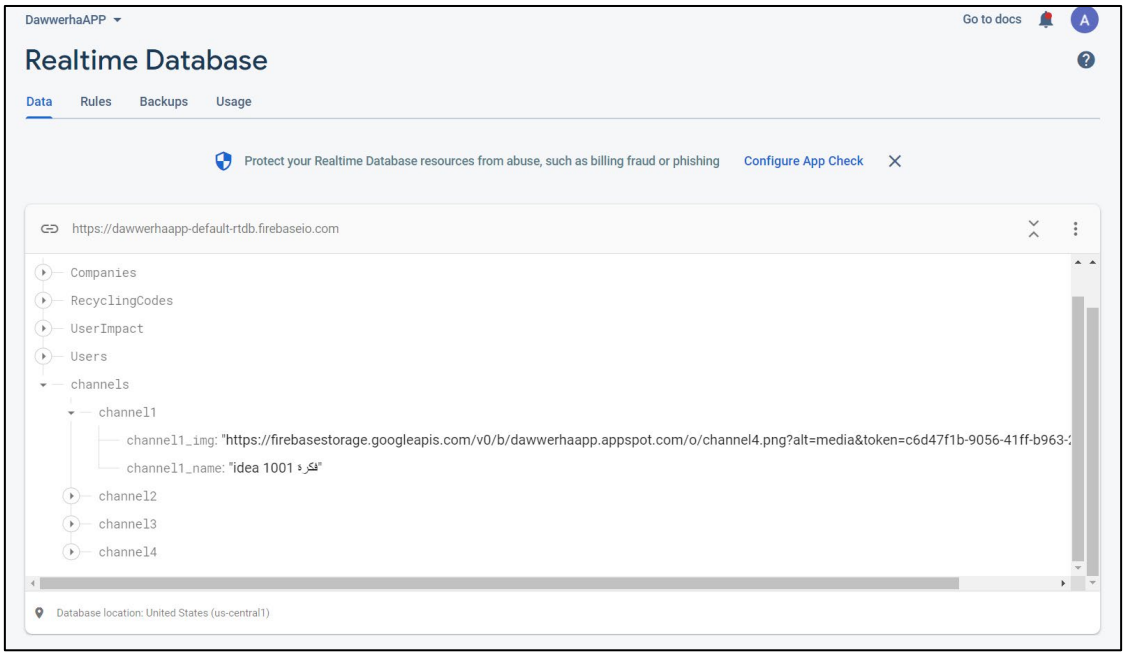


Figure 5.49: Firebase Real-time Database for the content of channels node

- Firebase Storage is using to uploading content like images that will used in our application as shown in Figure 5.50. [35][36][37]

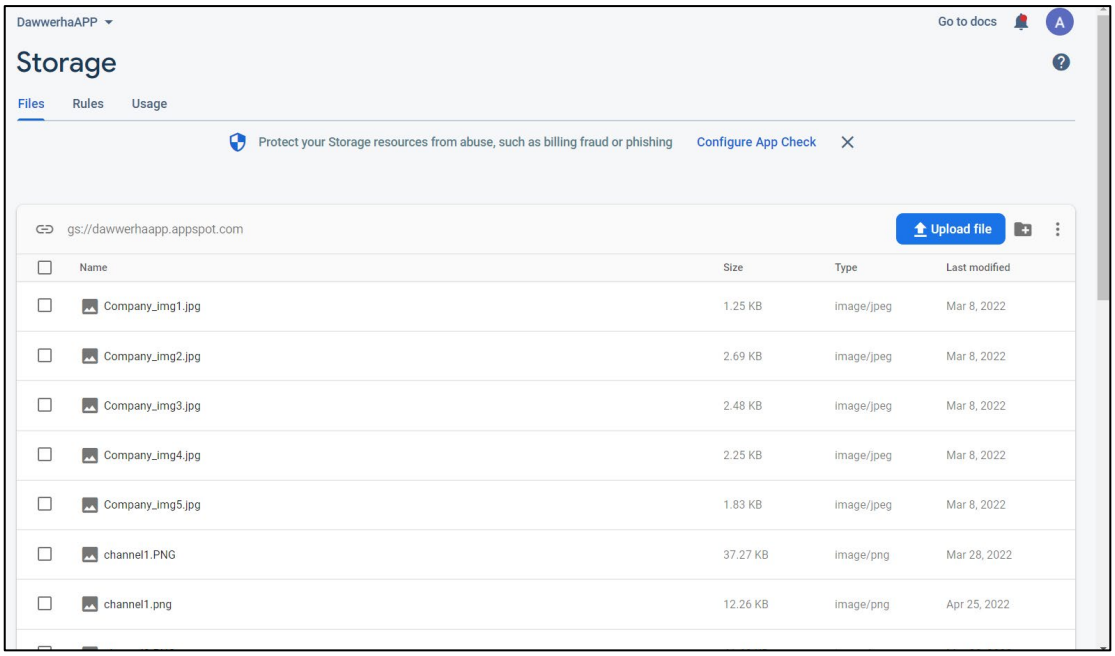


Figure 5.50: Firebase Storage for storage images

- Firebase Cloud massages used to send Notifications to users in specific time and date as shown in Figure 5.51. [35][36]

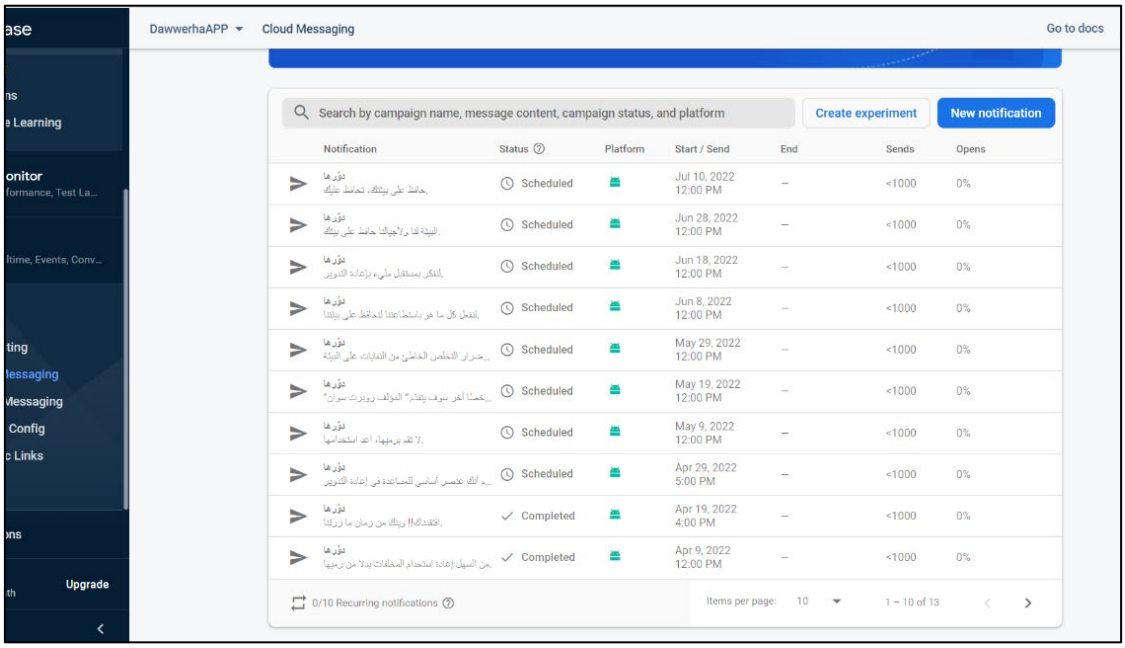


Figure 5.51: Firebase Cloud massages for Notifications

5.3 Machine learning Implementation

When we started working on the machine training part, we used The Jupyter Notebook environment to train the machine. We started searching for libraries and auxiliary codes to implement what we need to train. At first, we tested the codes on a small part of the dataset to make sure that it works and to know the accuracy of the results. The results were somewhat satisfying when it was a small dataset as shown Figures 5.52 - 5.54, but with the entire dataset, the experiment was not satisfying as we modified the codes, libraries, and method of work to obtain results, and it was not as satisfying as expected, as it was inaccurate, the error rate was high, and it took a lot of time to show the results, as the work environment.

Jupyter is training the machine from scratch based on the available data set, but it was not enough to achieve high accuracy. By searching for alternative solutions and experimenting with more than one method, codes and libraries, we found a machine learning work environment where it is pre-trained machine provides a focus on the data set that is entered with the presence of accuracy and ease of use.

```
Epoch 1/10
3/3 [==============================] - 5s 590ms/step - loss: 9.1101 - accurac
y: 0.7500 - val_loss: 0.6534 - val_accuracy: 0.7000
Epoch 2/10
3/3 [==============================] - 1s 485ms/step - loss: 4.1140 - accurac
y: 0.5000 - val_loss: 2.6397 - val_accuracy: 0.5000
Epoch 3/10
3/3 [==============================] - 2s 770ms/step - loss: 1.2468 - accurac
y: 0.7500 - val_loss: 1.3268 - val_accuracy: 0.5000
Epoch 4/10
3/3 [==============================] - 1s 516ms/step - loss: 0.7146 - accurac
y: 0.7500 - val_loss: 0.6127 - val_accuracy: 0.5000
Epoch 5/10
3/3 [==============================] - 1s 522ms/step - loss: 0.1741 - accurac
y: 1.0000 - val_loss: 0.6625 - val_accuracy: 0.5000
Epoch 6/10
3/3 [==============================] - 2s 648ms/step - loss: 0.0397 - accurac
y: 1.0000 - val_loss: 0.5312 - val_accuracy: 0.7000
Epoch 7/10
3/3 [==============================] - 2s 579ms/step - loss: 0.0138 - accurac
y: 1.0000 - val_loss: 0.5244 - val_accuracy: 0.7000
Epoch 8/10
3/3 [==============================] - 2s 584ms/step - loss: 0.0060 - accurac
y: 1.0000 - val_loss: 0.5205 - val_accuracy: 0.7000
Epoch 9/10
3/3 [==============================] - 2s 733ms/step - loss: 0.0023 - accurac
y: 1.0000 - val_loss: 0.5151 - val_accuracy: 0.7000
Epoch 10/10
3/3 [==============================] - 2s 474ms/step - loss: 8.6243e-04 - acc
uracy: 1.0000 - val_loss: 0.5167 - val_accuracy: 0.7000
```

Figure 5.52: Accuracy of machine training in jupyter

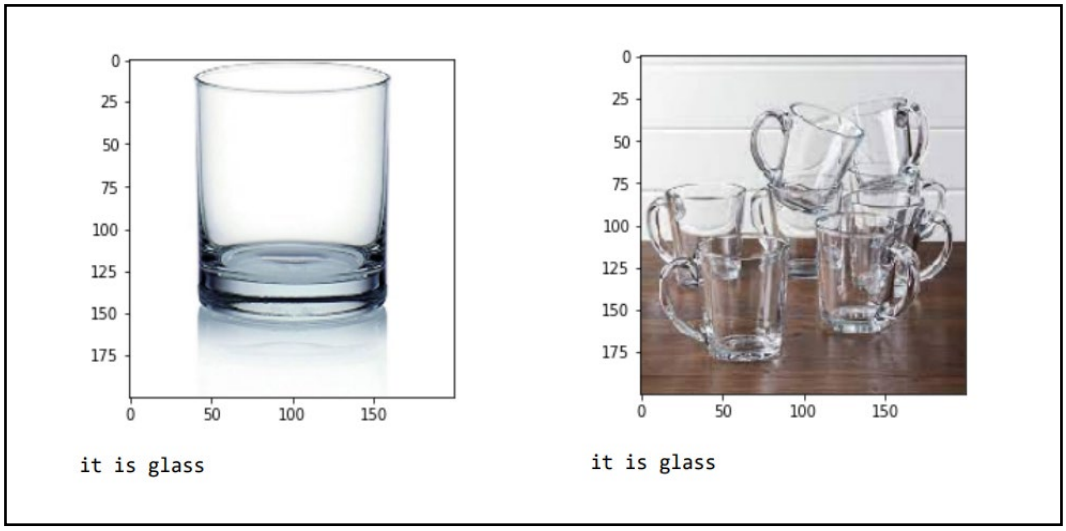


Figure 5.53: Machine training result (correct)

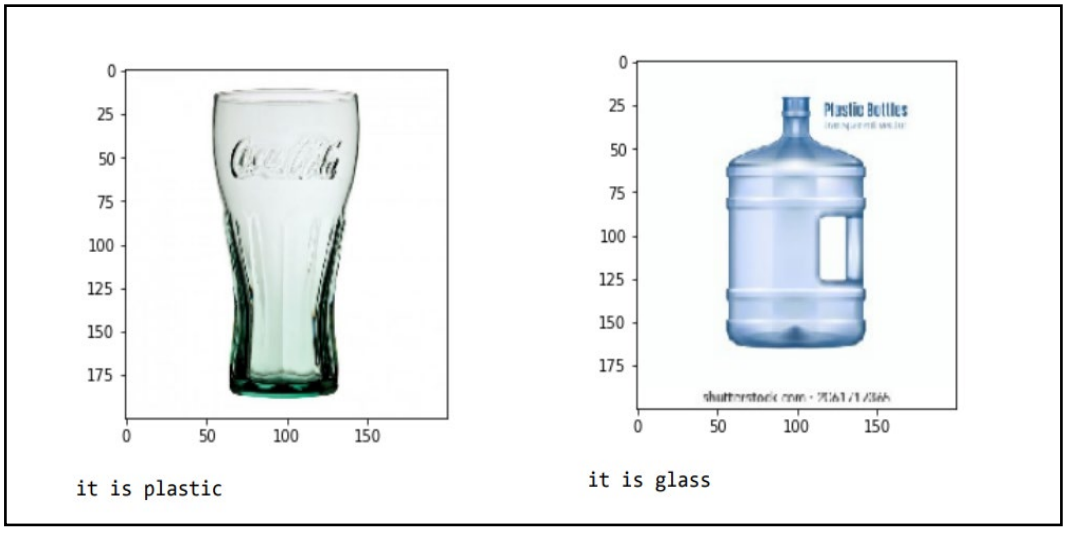


Figure 5.54: Machine training result (incorrect)

CHAPTER 6: Testing

This chapter will cover the testing phase including unit testing, machine learning testing and usability testing.

6.1 Development Testing

Is a process of software development strategy that aims to test the system during the development through get the development and testing phases together so that errors and bugs can be discovered.[18]

6.1.1 Unit testing

Is a process of software testing the individual components of program units or object classes. focusing on testing the functionality of objects or methods. [18]

6.1.1.1 Opening and welcome unit testing case

- When the user opens the app, the main logo appears as animation in the start page as shown in Figure 6.1.
- Then we will find the app welcome page, which contains three basic buttons sign up, log in or skip, which the user can use the app through them as shown in Figure 6.2.

Figure 6.1: Unit testing for Start page

Figure 6.2: Unit testing for Welcome page

6.1.1.2 Sign up unit testing case

- When the user chooses to create a new account by clicking sign up bouton, the user will be moved to sign up interface and the user will be asked to fill the required information as shown in Figure 6.3.
- If the user didn't fill out the information or fills it incorrectly, the user will be kept on the same interface with error messages appearing on the wrong required information to fills it correctly. Figures 6.4 – 6.9 represents all the errors that may appear in sign up interface.
- If the user fills in all the information correctly, the user will be moved to Homepage interface with messages shows that the account has been created successfully as shown in Figure 6.10.

Figure 6.3: Unit testing for Sign up

Figure 6.4: Unit testing for Sign up without filling user name

Figure 6.5: Unit testing for Sign up without filling user email

Figure 6.6: Unit testing for Sign up with writing wrong form email

Figure 6.7: Unit testing for Sign up without filling user password

Figure 6.8: Unit testing for Sign up without filling user confirmation password

Figure 6.9: Unit testing for Sign up with writing confirmation password doesn't match the password

Figure 6.10: Unit testing for Sign up successfully

6.1.1.3 Log in unit testing case

- When the user chooses to log in into the application by clicking log in bouton, the user will be moved to log in interface and the user will be asked to fill the required information as shown in Figure 6.11.
- If the user didn't fill out the information or fills it incorrectly, the user will be kept on the same interface with error messages appearing on the wrong required information to fills it correctly. Figures 6.12 – 6.14 represents all the errors that may appear in sign up interface.
- If the user fills in all the information correctly, the user will be moved to Homepage interface with messages shows that the logged in done successfully as shown in Figure 6.15.

Figure 6.11: Unit testing for Log in

Figure 6.12: Unit testing for Log in without filling user email

Figure 6.13: Unit testing for Log in without filling user password

Figure 6.14: Unit testing for Log in with writing wrong email or password

Figure 6.15: Unit testing for Log in successfully

6.1.1.4 Forgot password unit testing case

- When the user forgotten his/her password, he/she should click forgot password text, the user will be moved to forgot password interface and the user will be asked to fill the required information as shown in Figure 6.16.
- If the user didn't fill out the information or fills it incorrectly, the user will be kept on the same interface with error messages appearing on the wrong required information to fills it correctly. Figures 6.17 – 6.18 represents all the errors that may appear in sign up interface.
- If the user fills in all the information correctly, the user will be moved to Log in interface with messages shows that recovered link sent to the email as shown in Figure 6.19.
- When the user opens his/her email will find a password reset link, then the user can use the new one. Figures 6.20– 6.22 represents the reset password process.

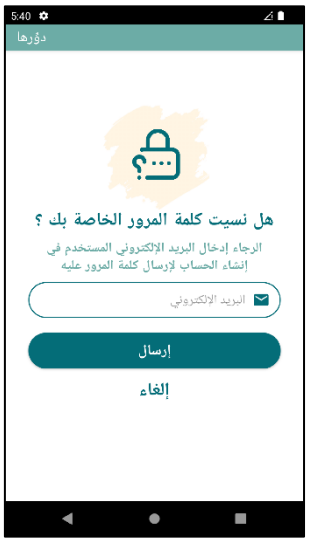


Figure 6.16: Unit testing for Forgot password

Figure 6.17: Unit testing for Forgot password without filling user email

Figure 6.18: Unit testing for Forgot password with writing email not found in database

Figure 6.19: Unit testing for Forgot password link sent successfully

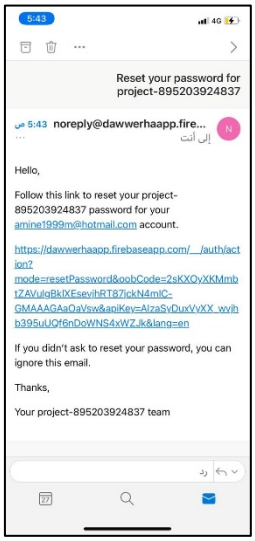


Figure 6.20: Unit testing for Forgot password reset link

Figure 6.21: Unit testing for Forgot password reset password

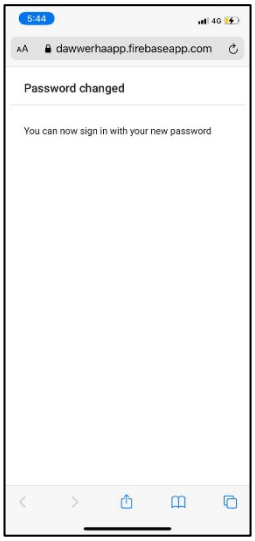


Figure 6.22: Unit testing for Forgot password password changed successfully

6.1.1.5 Homepage unit testing case

- After the user sign up or log in in the application, the user will be moved to Homepage interface as shown in Figure 6.23.
- The Homepage contains a drop-down menu and a bottom bar for quick access to the main functions of the application as shown in Figure 6.24.

Figure 6.23: Unit testing for Homepage

Figure 6.24: Unit testing for Homepage drop-down menu and a bottom bar

6.1.1.6 Add material quantity unit testing case

- After the user sign up or log in in the application, the user can use the add material quantity feature by click add button in the bottom bar, the user will be moved to Add material quantity interface as shown in Figure 6.25.
- The add material quantity contains the four materials categories (plastic, glass, metal, paper), the user chooses the number of pieces of materials that he/she wants to recycle then click the Add button to add to the new materials to the materials that was recycled as shown in Figure 6.26.

Figure 6.25: Unit testing for Add material quantity

Figure 6.26: Unit testing for Add material quantity successfully

6.1.1.7 Take photo unit testing case

- After the user sign up, log in or skip this in the application, the user can use the take photo feature by click camera button in the bottom bar, the user will be moved to take photo interface as shown in Figure 6.27.
- The take photo interface contains camera icon which must be clicked before adding the pieces of the material that the user has as shown in Figure 6.28, to classify the materials that the user would like to knew, then after the classify the user must sign up or log in to chooses the number of pieces of materials that he/she wants to recycle then click the Add button to add to the new materials to the materials that was recycled as shown in Figures 6.30-6.31.
- If the photo classified wrong, there is a button to help the user to get information to make the classification correctly as shown in Figure 6.29.

Figure 6.27: Unit testing for Take photo

Figure 6.28: Unit testing for Add material before taking photo

Figure 6.29: Unit testing for Take photo help button

Figure 6.30: Unit testing for Take photo classify

Figure 6.31: Unit testing for add material quantity successfully

6.1.1.8 User impact unit testing case

- After the user sign up, log in or skip this in the application, the user can use the user impact feature by click my impact button in the bottom bar, the user will be moved to user impact interface as shown in Figure 6.32, the page contains 10 questions the user must answer it to know his/her environmental impact.
  Figures 6.32 – 6.37 represents all the 10 questions in user impact interface.

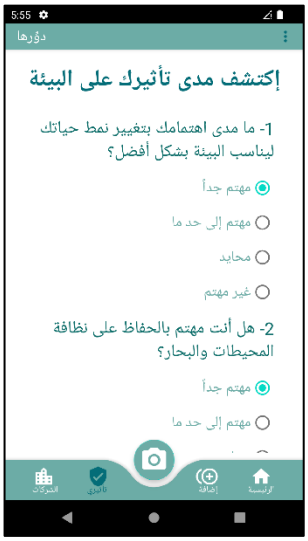


Figure 6.32: Unit testing for User impact page with questions 1,2

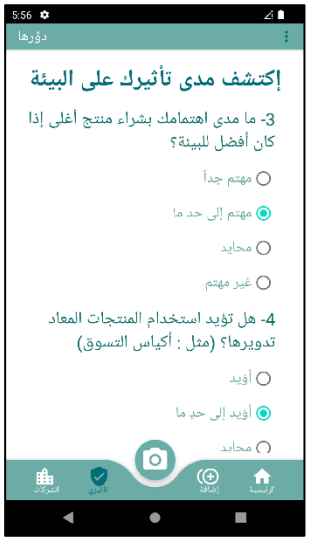


Figure 6.33: Unit testing for User impact page with questions 3,4

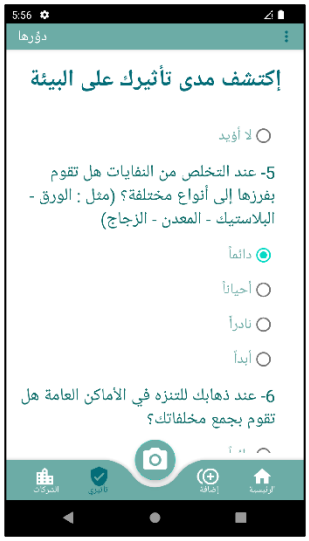


Figure 6.34: Unit testing for User impact page with question 5

Figure 6.35: Unit testing for User impact page with questions 6,7

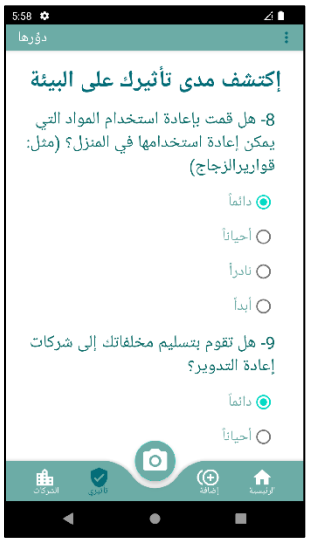


Figure 6.36: Unit testing for User impact page with questions 8,9

Figure 6.37: Unit testing for User impact page with question 10

- After the user answers the environmental impact questions, one of these results will appear to him/her shown in Figures 6.38-6.39-6.40, about whether his/her impact is excellent on the environment, good or bad, with some motivational phrases.

Figure 6.38: Unit testing for excellent result user impact

Figure 6.39: Unit testing for good result user impact

Figure 6.40: Unit testing for bad result user impact

- If the user does not answer all the questions, an error message will appear as shown in Figure 6.41.

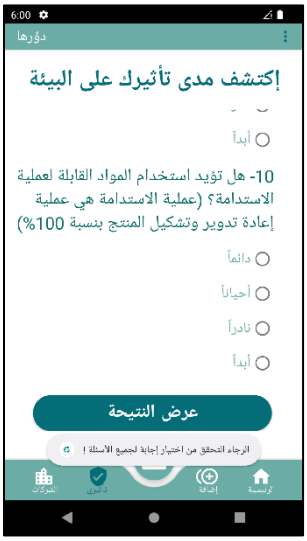


Figure 6.41: Unit testing for User impact with not answering all the questions

6.1.1.9 Recycling companies contact information unit testing case

- After the user sign up, log in or skip this in the application, the user can use the recycling companies contact information feature by click the companies button in the bottom bar, the user will be moved to the companies interface as shown in Figure 6.42, the page contains companies contact information name, number, address and logo.

Figure 6.42: Unit testing for Recycling companies contact information page

6.1.1.10 User profile unit testing case

- After the user sign up or log in in the application, the user can use the User profile feature by click my profile icon in the drop-down menu, the user will be moved to Profile page as shown in Figure 6.43, the page contains the user's name, email and the total pieces from each type of material that he/she was added before.
- The profile page contains log out icon in the top -left of the page so the user can log out from it, then the user will be moved to the welcome page as shown in Figure 6.44.

Figure 6.43: Unit testing for Profile page

Figure 6.44: Unit testing for Log out successfully

6.1.1.11 Change password unit testing case

- The profile page contains change password button in bottom of the page so the user can change his/her password from it by entering his/her uses email, old password and the new password as shown in Figure 6.45, the password have the same constrains like the one in the sign up page, then click on change button the user will be moved to the homepage as shown in Figure 6.46.

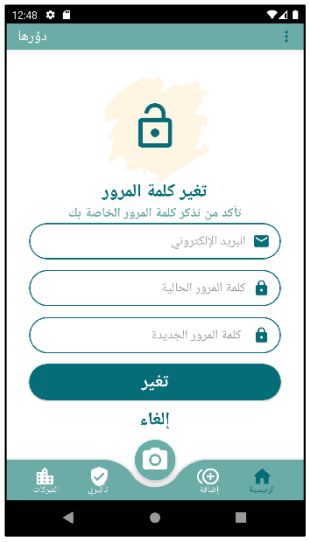


Figure 6.45: Unit testing for Change password page

Figure 6.46: Unit testing for Change password successfully

6.1.1.12 Suggested channels for recycling ideas unit testing case

- After the user sign up, skip log in in the application, the user can use the Suggested channels feature by click channels for recycling ideas icon in the drop-down menu, the user will be moved to Suggested channels page as shown in Figure 6.47, the page contains suggested channels from the YouTube to help the users to recycling reusing the things in their homes.

Figure 6.47: Unit testing for Suggested channels for recycling ideas page

6.1.1.13 Recycling codes unit testing case

- After the user sign up, skip log in in the application, the user can use the Recycling codes feature by click Recycling codes icon in the drop-down menu, the user will be moved to Recycling codes page as shown in Figure 6.48, the page contains the recycling codes photo with a description and example.

Figure 6.48: Unit testing for Recycling codes page

6.1.1.14 Tutorial unit testing case

- After the user sign up, skip log in in the application, the user can use the Tutorial feature by click How to use dawwerha icon in the drop-down menu, the user will be moved to How to use dawwerha page as shown in Figure 6.49, the page contains tutorial video of how to use the app.

Figure 6.49: Unit testing for Tutorial page

6.1.1.15 About us unit testing case

- After the user sign up, skip log in in the application, the user can use the About us feature by click About us icon in the drop-down menu, the user will be moved to About us page as shown in Figure 6.50, the page contains the aim of the application and the name and email for the developer's team.

Figure 6.50: Unit testing for About us page

6.1.1.16 Skip unit testing case

- When the user skips the sign up or log in to the application the user will not be able to use some features of the application, like add quantity and having profile, so a message will appear to the user that he/she must sign up or log in to be able to use these features.as shown in Figure 6.51.

Figure 6.51: Unit testing for Skip

6.1.1.17 Notifications unit testing case

- There are a number of notifications stored in the database that reach the user every ten days after starting to use the application to motivate the user as shown in Figure 6.52.

Figure 6.52: Unit testing for Notifications

6.1.2 Machine learning testing

It Contains ten pictures tested from each category that we have (paper, glass, metal, plastic) to determine the accuracy of the classification feature.

6.1.2.1 Paper classification testing case

- Here is a 10 pictures sample for taking photo classification for paper class as shown in Figures 6.53-6.64.
- Figures 6.53-6.60 show the correct results for paper classification.
- Figures 6.61-6.62 show the wrong result for paper classification, but after help icon and retaking the photo, it goes correct.
- Figures 6.63-6.64 show the wrong result for paper classification even after retaking the photo it still wrong.

Figure 6.53: Machine learning testing for correct paper classification

Figure 6.54: Machine learning testing for correct paper classification

Figure 6.55: Machine learning testing for correct paper classification

Figure 6.56: Machine learning testing for correct paper classification

Figure 6.57: Machine learning testing for correct paper classification

Figure 6.58: Machine learning testing for correct paper classification

Figure 6.59: Machine learning testing for correct paper classification

Figure 6.60: Machine learning testing for correct paper classification

Figure 6.61: Machine learning testing for wrong paper classification

Figure 6.62: Machine learning testing for correct paper classification

Figure 6.63: Machine learning testing for wrong paper classification

Figure 6.64: Machine learning testing for wrong paper classification

6.1.2.2 Glass classification testing case

- Here is a 10 pictures sample for taking photo classification for glass class as shown in Figures 6.65-6.75.
- Figures 6.65-6.73 show the correct results for glass classification.
- Figures 6.74-6.75 show the wrong result for glass classification even after retaking the photo it still wrong.

Figure 6.65: Machine learning testing for correct glass classification

Figure 6.66: Machine learning testing for correct glass classification

Figure 6.67: Machine learning testing for correct glass classification

Figure 6.68: Machine learning testing for correct glass classification

Figure 6.69: Machine learning testing for correct glass classification

Figure 6.70: Machine learning testing for correct glass classification

Figure 6.71: Machine learning testing for correct glass classification

Figure 6.72: Machine learning testing for correct glass classification

Figure 6.73: Machine learning testing for correct glass classification

Figure 6.74: Machine learning testing for wrong glass classification

Figure 6.75: Machine learning testing for wrong glass classification

6.1.2.3 Metal classification testing case

- Here is a 10 pictures sample for taking photo classification for metal class as shown in Figures 6.76-6.86.
- Figures 6.76-6.82 show the correct results for metal classification.
- Figures 6.83-6.84 show the wrong result for metal classification, but after help icon and retaking the photo, it goes correct.
- Figures 6.85-6.86 show the wrong result for metal classification even after retaking the photo it still wrong.

Figure 6.76: Machine learning testing for correct metal classification

Figure 6.77: Machine learning testing for correct metal classification

Figure 6.78: Machine learning testing for correct metal classification

Figure 6.79: Machine learning testing for correct metal classification

Figure 6.80: Machine learning testing for correct metal classification

Figure 6.81: Machine learning testing for correct metal classification

Figure 6.82: Machine learning testing for correct metal classification

Figure 6.83: Machine learning testing for wrong metal classification

Figure 6.84: Machine learning testing for correct metal classification

Figure 6.85: Machine learning testing for wrong metal classification

Figure 6.86: Machine learning testing for wrong metal classification

6.1.2.4 Plastic classification testing case

- Here is a 10 pictures sample for taking photo classification for plastic class as shown in Figures 6.87-6.97.
- Figures 6.87-6.95 show the correct results for plastic classification.
- Figures 6.96-6.97 show the wrong result for plastic classification, but after help icon and retaking the photo, it goes correct.

Figure 6.87: Machine learning testing for correct plastic classification

Figure 6.88: Machine learning testing for correct plastic classification

Figure 6.89: Machine learning testing for correct plastic classification

Figure 6.90: Machine learning testing for correct plastic classification

Figure 6.91: Machine learning testing for correct plastic classification

Figure 6.92: Machine learning testing for correct plastic classification

Figure 6.93: Machine learning testing for correct plastic classification

Figure 6.94: Machine learning testing for correct plastic classification

Figure 6.95: Machine learning testing for correct plastic classification

Figure 6.96: Machine learning testing for wrong plastic classification

Figure 6.97: Machine learning testing for correct plastic classification

6.1.3 Usability testing

The table below is a list of 5 questions asked to 10 users about their experience with trying the application.

- Have you face a problem while taking photo to your materials?
- Was the classification result after taking your photo, correct?
- Dose the application running fast enough?
- Dose the application design consistent?
- What are the difficulties that you faced while using the application?

users	Have you face a problem while taking photo to your materials?	Was the classification result after taking your photo, correct?	Dose the application running fast enough?	Dose the application design consistent?	What are the difficulties that you faced while using the application?
User1	No	Yes	no	Yes	-There are many requirements in the sign up, I prefer putting the mobile number only.
User2	No	No	Yes	Yes	-Click on camera icon in taking photo feature is ambiguous.
User3	No	Yes	No	Yes	-Click on camera icon in taking photo feature is ambiguous.
User4	No	Yes	Yes	Yes	-Click on camera icon in taking photo feature is ambiguous. -Last question in user impact is understandable and the answers is not consistent.
User5	No	Yes	Yes	Yes	-Click on logo icon to start using the application is ambiguous.
User6	No	Yes	Yes	Yes	-I haven't faced any difficulties
User7	No	No	No	Yes	-Click on camera icon in taking photo feature is ambiguous.
User8	No	Yes	No	Yes	-Last question in user impact is understandable and the answers is not consistent.
User9	No	No	Yes	Yes	-Click on logo icon to start using the application is ambiguous.
User10	No	Yes	Yes	Yes	-I haven't faced any difficulties

Table 6.1: Usability testing

Conclusion

Today we see that technology is present in our world more and better than before, so we as programmers must use the technology to serve, promote and develop our society.

In this project, we used some techniques that help increase society's awareness of recycling, which is one of the most required goals by the world, also to align with the vision of the Kingdom of Saudi Arabia 2030.

During the project development process, we faced many challenges. Some of these challenges are how to use the Android Studio environment in a short time, working on a new database concept Firebase (with nodes), the method of compiling and sorting the dataset, training the model using the Teachable machine, and also linking the Firebase and the model with Android Studio.

Our effort and work in this project earned us experience, skill and self-confidence and encouraged us to be innovative.

In the end, this project was a great and enjoyable experience, and we learned many lessons throughout it. Finally, we thank our supervisor, Dr. Reem Saleh Al Ashaikh, and everyone who helped us complete this project, and we hope that we have presented a work that contributes to the service of our community.

References

[1] "Environmental Protection in KSA," 2021. [Online]. Available: https://www.my.gov.sa/wps/portal/snp/aboutksa/environmentalProtection. [Accessed 15 november 2021].

[2] D. Achilias, Recycling Materials Based on Environmentally Friendly Techniques, 2015.

[3] I. El Naqa, R. Li and M. J. Murphy, What Is Machine Learning?, 2015.

[4] M. Mohri, A. Rostamizadeh and A. Talwalkar, Foundations of Machine Learning, second edition, 2018.

[5] B. Javidi, Image Recognition and Classification, 2001.

[6] J. D. Kothari , "A CASE STUDY OF IMAGE CLASSIFICATION BASED ON DEEP LEARNING USING," 2018.

[7] N. Shukla and K. Fricklas, Machine learning with TensorFlow2018 ,.

[8] H. Oinas-Kukkonen and M. Harjumaa, "Persuasive Systems Design: Key Issues, Process," 2009.

[9] B. J. Fogg, "Persuasive technology: using computers to change what we think and do," 2002.

[10] "qkons," [Online]. Available: http://www.qkons.com. [Accessed 8 september 2021].

[11] "trashly," [Online]. Available: https://trashly.io/about-us.html. [Accessed 8 september 2021].

[12] "tadweer," [Online]. Available: https://www.tadweer.gov.ae/ar/ESPAcc/Pages/default.aspx. [Accessed 8 september 2021].

[13] "live love recycle," [Online]. Available: https://apps.apple.com/lb/app/live-love-recycle/id1373708798. [Accessed 8 september 2021].

[14] B.-A. Andrei, A.-C. Casu-Pop, S.-C. Gheorghe and C.-A. Boiangiu, "A study on using waterfall and agile methods in software project management," 2019.

[15] I. Sommerville, Software Engineering(TENTH EDITION), 2016.

[16] A. DENNIS, B. HALEY WIXOM and R. M. ROTH, Systems Analysis and Design, 5th Edition, 2012.

[17] I. Sommerville, Software engineering 9th2011 ,.

[18] A. DENNIS, B. HALEY WIXOM and R. M. ROTH, Systems Analysis and Design with UML, 4th Edition, 2012.

[19] R. Elmasri and S. B. Navathe, FUNDAMENTALS OF Database Systems, 2016.

[20] A. Khan, B. Baharudin, L. H. Lee and K. Khan, "A review of machine learning algorithms for text-documents classification," 2010.

[21] "Machine Learning Datasets | Various Types of Datasets for Data Scientists," 2021. [Online]. Available: https://www.educba.com/machine-learning-datasets/. [Accessed 17 october 2021].

[22] "An introduction to Teachable Machine — AI for dummies," [Online]. Available: https://blog.etereo.io/an-introduction-to-teachable-machine-ai-for-dummies-61d1f97f5cf. [Accessed 16 april 2022].

[23] "How to use confidence scores in machine learning models," [Online]. Available: https://towardsdatascience.com/how-to-use-confidence-scores-in-machine-learning-models-abe9773306fa. [Accessed 16 april 2022].

[24] "github," [Online]. Available: https://github.com/garythung/trashnet. [Accessed 17 april 2022].

[25] "[PyTorch] Garbage Classification (~95% accuracy)," [Online]. Available: https://www.kaggle.com/aadhavvignesh/pytorch-garbage-classification-95-accuracy. [Accessed 17 april 2022].

[26] "drive recycling waste," [Online]. Available: https://drive.google.com/drive/folders/1CTvT_gnTvwlcKwJ8yz4jUOs0JYTKrplA. [Accessed 17 april 2022].

[27] "Why Do We Need a Validation Set in Addition to Training and Test Sets?," [Online]. Available: https://towardsdatascience.com/why-do-we-need-a-validation-set-in-addition-to-training-and-test-sets-5cf4a65550e0. [Accessed 20 april 2022].

[28] "How to Split a Dataset Into Training and Testing Sets with Python," [Online]. Available: https://towardsdatascience.com/how-to-split-a-dataset-into-training-and-testing-sets-b146b1649830. [Accessed 20 april 2022].

[29] M. Carney, B. Webster, I. Alvarado, K. Phillips, N. Howell, J. Griffith, J. Jongejan, A. Pitaru and A. Chen, "Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification," 2020.

[30] "Transfer learning from pre-trained models," [Online]. Available: https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751. [Accessed 20 april 2022].

[31] "Image Classification With MobileNet," [Online]. Available: https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470. [Accessed 20 april 2022].

[32] "Epoch," [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/epoch. [Accessed 21 april 2022].

[33] "Introducing Batch Size," [Online]. Available: https://deeplizard.com/learn/video/U4WB9p6ODjM#:~:text=Put%20simply%2C%20the%20batch%20size%20is%20the%20number,that%20are%20passed%20to%20the%20network%20at%20once. [Accessed 21 april 2022].

[34] "Understanding Learning Rates and How It Improves Performance in Deep Learning," [Online]. Available: https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10. [Accessed 21 april 2022].

[35] "java T point," [Online]. Available: Firebase Introduction - Javatpoint. [Accessed 25 april 2022].

[36] "What is Firebase?," [Online]. Available: https://www.educative.io/edpresso/what-is-firebase. [Accessed 25 april 2022].

[37] "5 tips for Firebase Storage," [Online]. Available: https://firebase.blog/posts/2016/07/5-tips-for-firebase-storage. [Accessed 25 april 2022].

[38] "Firebase – A Real Time Document Database," [Online]. Available: https://superdevelopment.com/2016/03/01/firebase-a-real-time-document-database/#:~:text=On%20the%20surface%2C%20Firebase%20is%20like%20any%20other,CouchDB%20all%20store%20data%20primarily%20the%20same%20way. [Accessed 7 may 2022].

Appendix

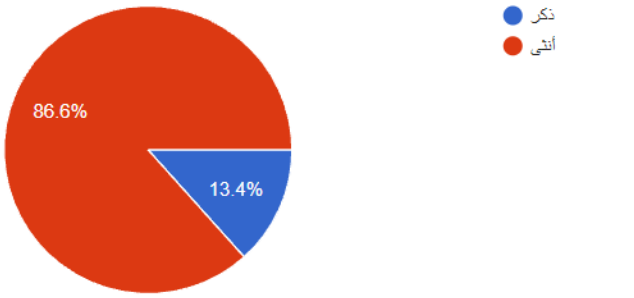
الأسئلة الردود 202 الإعدادات

استبيان حول إنشاء تطبيق لإعادة التدوير .

عزيزي المستخدم ..
نحن طالبات قسم علوم الحاسب الالي بجامعة ام القرى ..
نقدم بين يديك هذا الاستبيان والذي يهدف لمعرفة مدى احتياجك لتطبيق يساعدك في الاهتمام والمحافظة على البيئة , لذلك ليس هناك إجابات خاطئة وأخرى صحيحة, علماً بأن هذه الإجابات لن يتم مشاركتها مع أي جهة أخرى ولن تستخدم إلا لغرض الدراسة .
شكراً لك على حسن تعاونك

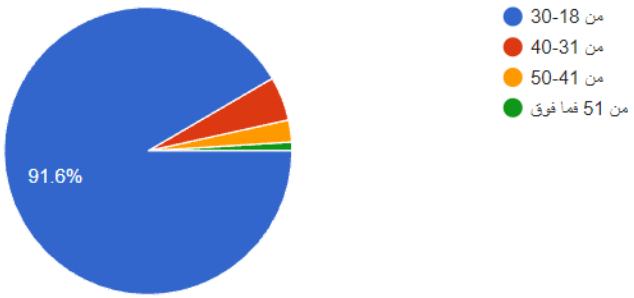
نوع الجنس ؟

202 رد



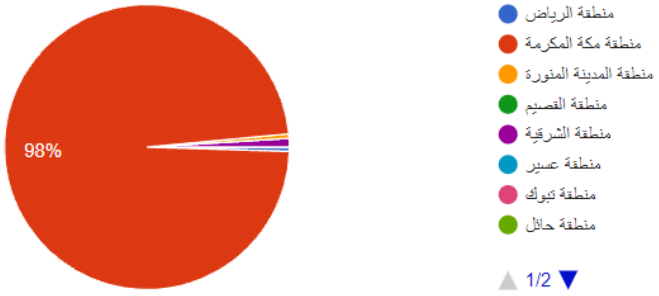
إلى أي فئة عمرية تنتمي ؟

202 رد



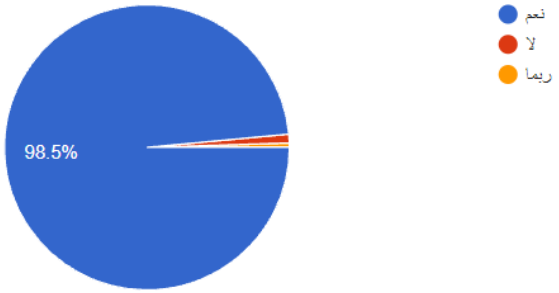
المنطقة التي تنتمي اليها ؟

202 رد



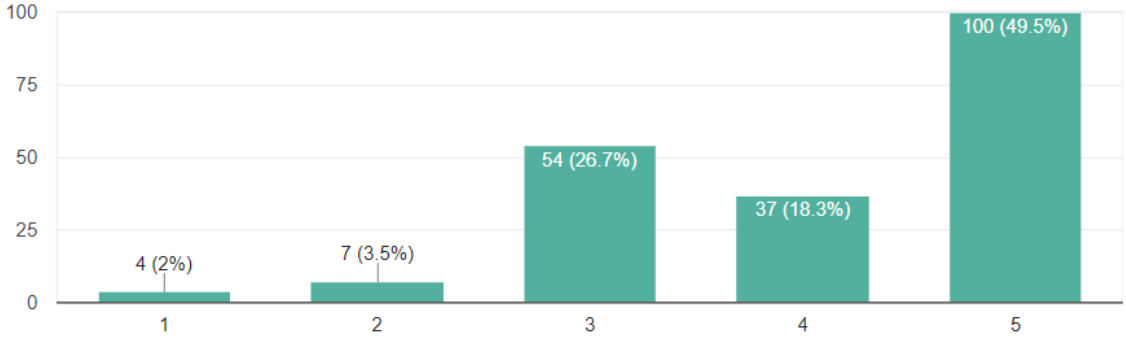
هل سمعت من قبل عن مصطلح إعادة التدوير ؟

202 رد



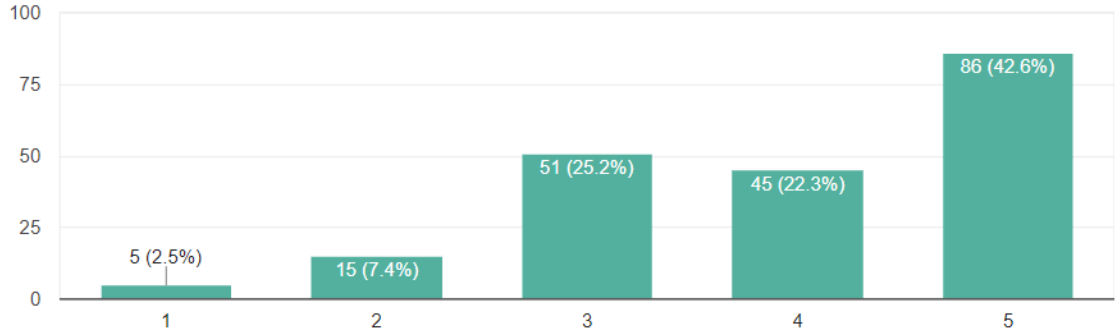
هل أنت من الأشخاص الذين يهتمون بالحفاظ على البيئة ؟

202 رد



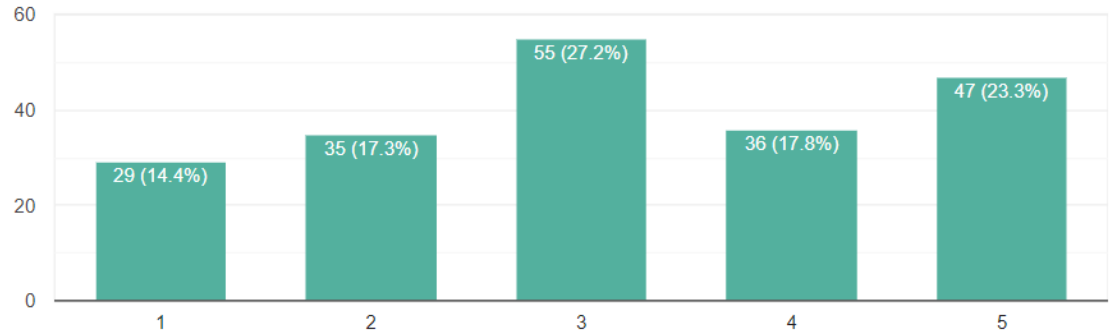
ما معدل استهلاكك للمنتجات التي يمكن إعادة تدويرها مثل علب المياه واكياس التسوق البلاستيكية ؟

202 رد



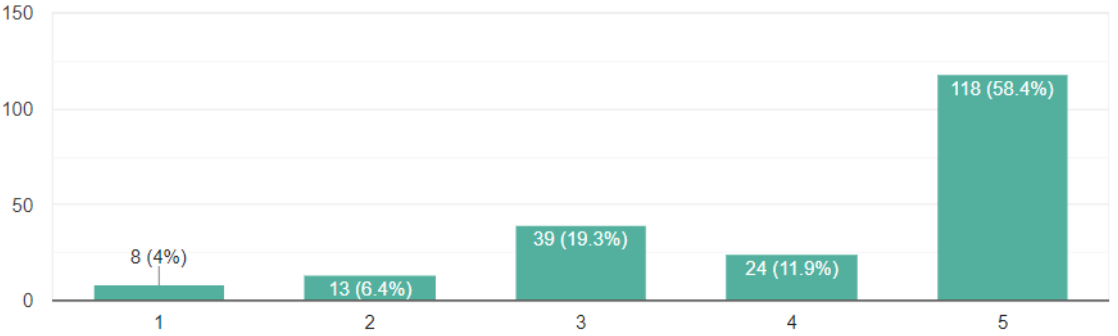
اواجه صعوبة في تحديد المواد المستهلكة المراد إعادة تدويرها ؟

202 رد



اواجه صعوبة في التواصل مع شركات إعادة تدوير ؟

202 رد



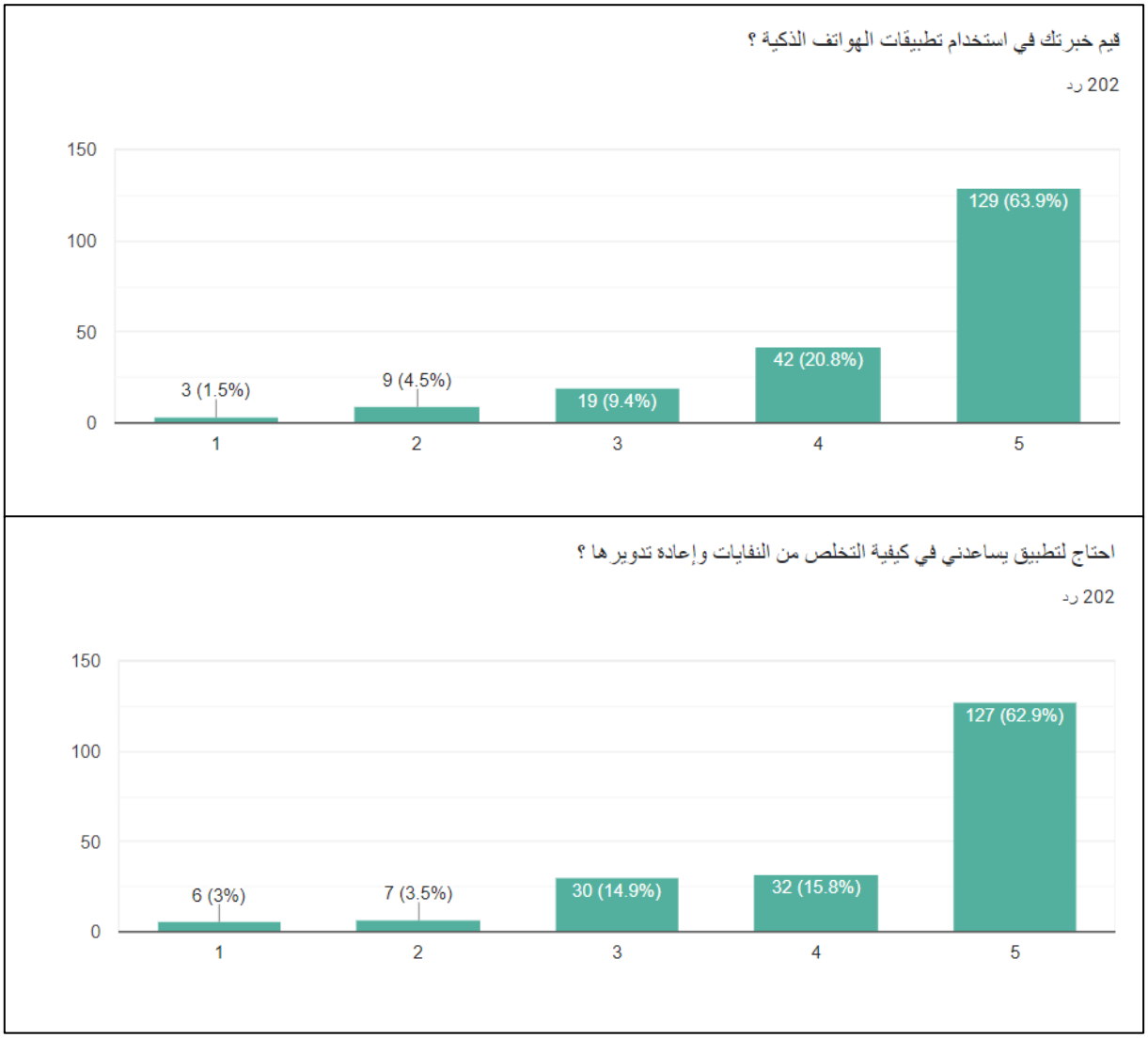


Figure 7: Questionnaire for Dawwerha application