

# MY POSCO.COM : NATURAL DISASTER INFORMATION SYSTEM USING WEB-BASED

**BIT304 FINAL YEAR PROJECT I**

E1400462 Tommy E1400459 Rizqi Daniya

E1400454 Ni Made Sinthia Pradnya Wedanthi

Submitted to the

FACULTY OF COMPUTING AND DIGITAL TECHNOLOGY

(SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY)

In partial fulfilment of the requirements for the degree of

Bachelor of Information Technology ( HONS )

HELP UNIVERSITY SEPTEMBER-DECEMBER 2017

# Abstract

The rapid and comprehensive development of information helps many people to know the latest news that is happening around them. One of the information that currently attracts a lot of attention is the news of eruption of Agung mountain in Bali. Status of the Agung mountain has entered the alert level and many residents who live close to the Agung mountain flee to safer places. The amount of information circulating in social media helps the process of evacuation and distribution of aid to be faster but not all information obtained can be trusted just like that. Invalid information has an impact on the uneven distribution of aid and the accumulation of unnecessary goods at refugee command post. Therefore, an information system on natural disasters is necessary to avoid invalid information spread across the community, facilitate the user to find out the assistance needed for each command post and to ensure the originality and validity of an information. Not only that, the natural disaster information system can be a place for volunteer recruitment or donation registration. Making natural disaster information system and how the result of information system can help user become the scope of our paper. Finally, a natural disaster information system has been built using web based platform, which has been adjusted to the project objectives, aims and requirements that we have set together. This information site has aim to help the people of Indonesia to channel their assistance to the natural disaster refugees in the registered posts appropriately and quickly. In our site, especially the command post section is given a get direction feature, which feature is very useful for users to track their trip to one of their chosen posts. So that the process of aid distribution can be done more efficiently.

# Declaration

We hereby declare that the report presented here as part of the requirement of BIT304 is original and no parts of this report had been plagiarised from any other resources unless those indicated with proper referencing. Resource from other people work has been entered as references without changing the name, date, title or anything else related to the resources we use for our project paper. We recognize that the act of plagiarism is an act that is not commendable and is the act of illegally stealing another's work. This report will be the property of HELP University and STIKOM Bali and cannot be distributed in any form without the written consent or permission from HELP University and STIKOM Bali.

Date:

Student Names:

Tommy Rizqi Daniya Ni Made Sinthia Pradnya Wedanthi

# Acknowledgement

This paper can be completed is not merely the result of our work but the encouragement and assistance given by various parties. Through this opportunity, we would like to express our deepest gratitude to Mr.Yudi Agusta who has devoted much of his time as our mentor who taught us about how the process of software project was made, help us when we have difficulty in understanding a necessary requirement in the project, providing suggestion, knowledge and guidance so that this paper can be completed. We are also grateful for the help and advice given by Ms. Anitha Velayutham and Mr Wirarama Wedhaswara in developing our project. Not to forget the support and motivation given by our beloved family and friends and also the senior who have completed previous projects so that we can discuss together about the making of the project. Last but not least, the grace of God who almighty has given us strength, health and fortune to prepare and finish this paper of final project 1 with all requirement that we have set together.

## Table of Contents

Abstract i Declaration ii Acknowledgment iii Table of Content iv List of Tables viii List of Figures xii **Chapter 1 Proposal 1**

* 1. Introduction 1
  2. Issues in Current System 2
  3. Benefits and Constraints of Proposed System 2
  4. Project Description 3
  5. Project Aims and Objectives 4
  6. Project Scope 5
  7. Software and Hardware Requirements 6
     1. Hardware 6
     2. Software 6
     3. Tools 8
     4. Resource Requirements 9
  8. Development Methodology 10
  9. Techniques 11
     1. Interview 11
     2. Black and White Box Testing 11
     3. Object Oriented Programming 12

### Chapter 2 Project Management Plan 13

* 1. Introduction 13
  2. Work Item List and Work Break Down Structure 13
  3. Risk Management 18
     1. Risk Identification 18
     2. Risk Register 19
     3. Risk Analysis 20
     4. SWOT Analysis 22
  4. Gantt Chart and Milestone 23
  5. References 26

### Chapter 3 Requirement Analysis 27

* 1. Introduction 27
  2. Requirement Summary 27
  3. Functional Requirement 28
  4. Non-functional Requirement 32
  5. Iteration Plan 33
     1. Feasibility Study Analysis 34
     2. Detailed Analysis 34
     3. Detailed Design 34
     4. Construction 35
     5. Testing 35
     6. System Delivery 35
     7. Operation & Maintenance 35
  6. Iteration 1 39
     1. Introduction 39
     2. Purpose 39
     3. Context 40
     4. Schedule of Iteration Workflows 40
     5. Iteration Schedule Breakdown 41
     6. Resource Summary 43
     7. Evaluation Criteria 43
     8. Iteration Review and Evaluation 44
  7. Analysis Artifacts 45
     1. Use Case Diagram 45
     2. High Level Use Case 47
     3. Expanded Use Case 56
     4. Analysis Class Diagram 78
  8. Design Artifacts 79
     1. Sequence Diagram 79
     2. Database Design 96
     3. Structural Design 103
     4. User Interface 103
        1. Designing Language 103
        2. Color Used 104
        3. Website User Interface for User 105
        4. Website User Interface for Admin 121
  9. Implementation/Testing 157
     1. Introduction 157
     2. Implementation Methodology 157
     3. Unit Testing 157
     4. System Testing 169
     5. Testing Phase Conclusion 173
  10. References 174

### Chapter 4 Conclusion 175

* 1. Introduction 175
  2. Shortcoming and Solutions 175
  3. What Went Right 176
  4. What Went Wrong 177
  5. Lesson Learned 177
  6. Further Improvement 178
  7. Future Work 178
  8. Conclusion 179 Appendix A: Gantt Chart 180 Appendix B: Progress Report 181
  9. September Project Report 181
  10. October Project Report 182
  11. November Project Report 183
  12. December Project Report 184 Appendix C: Minutes Meeting 186 Appendix D: Team Contract 199 Appendix E: Vision 201
  13. Introduction 201
  14. Positioning 201
      1. Problem Statement 201
      2. Product Position Statement 202
  15. Stakeholder Description 203
      1. Stakeholder Summary 203
  16. User Environment 205 Appendix F: Project Evaluation 206

## List of Tables

1. Table 1.7.1.1 Hardware 6
2. Table 1.7.2.1 Software 7
3. Table 1.7.2.2 Application 8
4. Table 1.7.3.1 Tools 9
5. Table 2.2.2.1 Work Item List 17
6. Table 2.3.2.1 Risk Register 20
7. Table 2.3.3.1 Rank of Risk 21
8. Table 2.4.1.1 Milestone 23
9. Table 3.4.1 Non-Functional Requirement 33
10. Table 3.5.8 Iteration Plan 38
11. Table 3.6.4.1 Iteration Workflow Schedule 41
12. Table 3.6.5.1 Iteration Plan Task Breakdown by Workflow 42
13. Table 2.6.6.1 Iteration Workflow Schedule 43
14. Table 3.7.2.1 High Level Use Case of Register 47
15. Table 3.7.2.2 High Level Use Case of Log in User 47
16. Table 3.7.2.3 High Level Use Case of Log in Admin 47
17. Table 3.7.2.4 High Level Use Case of Log Out 47
18. Table 3.7.2.5 High Level Use Case of Edit Profile 48
19. Table 3.7.2.6 High Level Use Case of View List of Province 48
20. Table 3.7.2.7 High Level Use Case of View List of Natural Disaster 48
21. Table 3.7.2.8 High Level Use Case of View List of Command Post 48
22. Table 3.7.2.9 High Level Use Case of Donate Money 49
23. Table 3.7.2.10 High Level Use Case of Donate Event 49
24. Table 3.7.2.11 High Level Use Case of Donate Logistical Aids 49
25. Table 3.7.2.12 High Level Use Case of Become a Volunteer 49
26. Table 3.7.2.13 High Level Use Case of Update Details of Command Post 50
27. Table 3.7.2.14 High Level Use Case of Add Stock in Command Post 50
28. Table 3.7.2.15 High Level Use Case of Add Needs in Command Post 50
29. Table 3.7.2.16 High Level Use Case of Update Stock in Command Post 50
30. Table 3.7.2.17 High Level Use Case of Update Needs in Command Post 51
31. Table 3.7.2.18 High Level Use Case of Delete Stock in Command Post 51
32. Table 3.7.2.19 High Level Use Case of Delete Needs in Command Post 51
33. Table 3.7.2.20 High Level Use Case of Event Approval 51
34. Table 3.7.2.21 High Level Use Case of Volunteer Approval 52
35. Table 3.7.2.22 High Level Use Case of Add Admin of Command Post 52
36. Table 3.7.2.23 High Level Use Case of Edit Admin of Command Post 52
37. Table 3.7.2.24 High Level Use Case of Delete Admin of Command Post 52
38. Table 3.7.2.25 High Level Use Case of Add Natural Disaster 53
39. Table 3.7.2.26 High Level Use Case of Edit Natural Disaster 53
40. Table 3.7.2.27 High Level Use Case of Delete Natural Disaster 53
41. Table 3.7.2.28 High Level Use Case of View Admin of Natural Disaster 53
42. Table 3.7.2.29 High Level Use Case of Add Admin of Natural Disaster 54
43. Table 3.7.2.30 High Level Use Case of Edit Admin of Natural Disaster 54
44. Table 3.7.2.31 High Level Use Case of Delete Admin of Natural Disaster 54
45. Table 3.7.2.32 High Level Use Case of Add Admin of Province 54
46. Table 3.7.2.33 High Level Use Case of Edit Admin of Province 55
47. Table 3.7.2.34 High Level Use Case of Delete Admin of Province 55
48. Table 3.7.2.35 High Level Use Case of Submit Testimonials 55
49. Table 3.7.2.36 High Level Use Case of Approve Testimonials 55
50. Table 3.7.2.37 High Level Use Case of Payment Approval 56
51. Table 3.7.3.1 Expanded Use Case of Register 56
52. Table 3.7.3.2 Expanded Use Case of Log in User 57
53. Table 3.7.3.3 Expanded Use Case of Log in Admin 58
54. Table 3.7.3.4 Expanded Use Case of Log Out 58
55. Table 3.7.3.5 Expanded Use Case of Edit Profile 58
56. Table 3.7.3.6 Expanded Use Case of View List of Provinces 59
57. Table 3.7.3.7 Expanded Use Case of View List of Natural Disaster 59
58. Table 3.7.3.8 Expanded Use Case of View List of Command Post 60
59. Table 3.7.3.9 Expanded Use Case of Donate Money 61
60. Table 3.7.3.10 Expanded Use Case of Donate Event 62
61. Table 3.7.3.11 Expanded Use Case of Donate Logistical Aids 63
62. Table 3.7.3.12 Expanded Use Case of Become a Volunteer 64
63. Table 3.7.3.13 Expanded Use Case of Update Details of Command Post 64
64. Table 3.7.3.14 Expanded Use Case of Add Stock in Command Post 65
65. Table 3.7.3.15 Expanded Use Case of Add Needs in Command Post 66
66. Table 3.7.3.16 Expanded Use Case of Update Stock in Command Post 66
67. Table 3.7.3.17 Expanded Use Case of Delete Stock in Command Post 67
68. Table 3.7.3.18 Expanded Use Case of Update Needs in Command Post 67
69. Table 3.7.3.19 Expanded Use Case of Delete Needs in Command Post 68
70. Table 3.7.3.20 Expanded Use Case of Event Approval 68
71. Table 3.7.3.21 Expanded Use Case of Volunteer Approval 69
72. Table 3.7.3.22 Expanded Use Case of Add Admin of Command Post 69
73. Table 3.7.3.23 Expanded Use Case of Edit Admin of Command Post 70
74. Table 3.7.3.24 Expanded Use Case of Add Natural Disaster 70
75. Table 3.7.3.25 Expanded Use Case of Edit Natural Disaster 71
76. Table 3.7.3.26 Expanded Use Case of Delete Natural Disaster 71
77. Table 3.7.3.27 Expanded Use Case of Add Admin of Natural Disaster 72
78. Table 3.7.3.28 Expanded Use Case of Delete Admin of Command Post 72
79. Table 3.7.3.29 Expanded Use Case of View Admin of Natural Disaster 73
80. Table 3.7.3.30 Expanded Use Case of Edit Admin of Natural Disaster 74
81. Table 3.7.3.31 Expanded Use Case of Delete Admin of Natural Disaster 74
82. Table 3.7.3.32 Expanded Use Case of Add Admin of Province 75
83. Table 3.7.3.33 Expanded Use Case of Edit Admin of Province 75
84. Table 3.7.3.34 Expanded Use Case of Delete Admin of Province 76
85. Table 3.7.3.35 Expanded Use Case of Submit Testimonials 76
86. Table 3.7.3.36 Expanded Use Case of Approve Testimonials 77
87. Table 3.7.3.37 Expanded Use Case of Payment Approval 77
88. Table 3.8.2.1 Database Design Super Admin 96
89. Table 3.8.2.2 Database Design User 97
90. Table 3.8.2.3 Database Design Volunteer 97
91. Table 3.8.2.4 Database Design Donation Event 98
92. Table 3.8.2.5 Database Design Donation Money 98
93. Table 3.8.2.6 Database Design Testimonial 98
94. Table 3.8.2.7 Database Design Admin of Province 99
95. Table 3.8.2.8 Database Design Province 99
96. Table 3.8.2.9 Database Design Admin of Natural Disaster 100
97. Table 3.8.2.10 Database Design Natural Disaster 100
98. Table 3.8.2.11 Database Design Admin of Command Post 101
99. Table 3.8.2.12 Database Design Command Post 102
100. Table 3.8.2.13 Database Logistical Stock 102
101. Table 3.9.3.1 Unit Testing: Register User 162
102. Table 3.9.3.2 Unit Testing: Login User 165
103. Table 3.9.3.3 Unit Testing: Login Admin 168
104. Table 3.9.4.1 Website Platform: Super Admin Login 169
105. Table 3.9.4.2 Website Platform: Admin of Province Login 170
106. Table 3.9.4.3 Website Platform: Admin of Natural Disaster Login 171
107. Table 3.9.4.4 Website Platform: Admin of Command Post Login 171
108. Table 3.9.4.5 Website Platform: Register User 172
109. Table 3.9.4.6 Website Platform: Login User 173

## List of Figures

1. Figure 1.4.1.1 Hierarchy of Admins 3
2. Figure 1.8.1.1 Waterfall with Prototyping Methodology 10
3. Figure 2.2.2.2 Work Breakdown Structure 17
4. Figure 2.3.3.1 Probability and Impact Matrix 21
5. Figure 2.4.1.2 Gantt Chart/Schedule 25
6. Figure 3.7.1.1 Use Case Diagram of Registered User and Super Admin 45
7. Figure 3.7.1.2 Use Case Diagram of Registered User and Province Admin 45
8. Figure 3.7.1.3 Use Case Diagram of Registered User and Disaster Admin 46
9. Figure 3.7.1.4 Use Case Diagram of Registered User and Command Post Admin 46
10. Figure 3.7.4.1 Analysis Class Diagram 78
11. Figure 3.8.1.1 Register Sequence Diagram 79
12. Figure 3.8.1.2 Login User Sequence Diagram 79
13. Figure 3.8.1.3 Login Admin Sequence Diagram 80
14. Figure 3.8.1.4 Log Out Sequence Diagram 80
15. Figure 3.8.1.5 Edit Profile Sequence Diagram 81
16. Figure 3.8.1.6 View List of Province Sequence Diagram 81
17. Figure 3.8.1.7 View List of Natural Disaster Sequence Diagram 81
18. Figure 3.8.1.8 View List of Command Post Sequence Diagram 82
19. Figure 3.8.1.9 Donate Money Sequence Diagram 82
20. Figure 3.8.1.10 Donate Event Sequence Diagram 83
21. Figure 3.8.1.11 Donate Logistical Aid Sequence Diagram 83
22. Figure 3.8.1.12 Become a Volunteer Sequence Diagram 84
23. Figure 3.8.1.13 Update Details of Command Post Sequence Diagram 84
24. Figure 3.8.1.14 Add Stock in Command Post Sequence Diagram 85
25. Figure 3.8.1.15 Add Needs in Command Post Sequence Diagram 85
26. Figure 3.8.1.16 Update Stock in Command Post Sequence Diagram 85
27. Figure 3.8.1.17 Update Needs in Command Post Sequence Diagram 86
28. Figure 3.8.1.18 Delete Stock in Command Post Sequence Diagram 86
29. Figure 3.8.1.19 Delete Needs in Command Post Sequence Diagram 87
30. Figure 3.8.1.20 Event Approval Sequence Diagram 87
31. Figure 3.8.1.21 Volunteer Approval Sequence Diagram 88
32. Figure 3.8.1.22 Add Admin Command Post Sequence Diagram 88
33. Figure 3.8.1.23 Edit Admin Command Post Sequence Diagram 89
34. Figure 3.8.1.24 Delete Admin Command Post Sequence Diagram 89
35. Figure 3.8.1.25 Add Natural Disaster Sequence Diagram 90
36. Figure 3.8.1.26 Edit Natural Disaster Sequence Diagram 90
37. Figure 3.8.1.27 Delete Natural Disaster Sequence Diagram 91
38. Figure 3.8.1.28 View Admin of Natural Disaster Sequence Diagram 91
39. Figure 3.8.1.29 Add Admin of Natural Disaster Sequence Diagram 92
40. Figure 3.8.1.30 Edit Admin of Natural Disaster Sequence Diagram 92
41. Figure 3.8.1.31 Delete Admin of Natural Disaster Sequence Diagram 93
42. Figure 3.8.1.32 Add Admin of Province Sequence Diagram 93
43. Figure 3.8.1.33 Edit Admin of Province Sequence Diagram 94
44. Figure 3.8.1.34 Delete Admin of Province Sequence Diagram 94
45. Figure 3.8.1.35 Submit Testimonials Sequence Diagram 95
46. Figure 3.8.1.36 Approve Testimonials Sequence Diagram 95
47. Figure 3.8.1.37 Payment Approval Sequence Diagram 96
48. Figure 3.8.3.1 Structural Design 103
49. Figure 3.8.4.2.1 Color for UI of User 104
50. Figure 3.8.4.2.2 Color for UI of Admin 104
51. Figure 3.8.4.3.1 Web UI for User: Homepage Header 105
52. Figure 3.8.4.3.2 Web UI for User: Homepage Body 106
53. Figure 3.8.4.3.3 Web UI for User: Province Page 107
54. Figure 3.8.4.3.4 Web UI for User: Natural Disaster Page 108
55. Figure 3.8.4.3.5 Web UI for User: Command Post List Page 108
56. Figure 3.8.4.3.6 Web UI for User: A Command Post Page 109
57. Figure 3.8.4.3.7 Web UI for User: A Command Post Page 111
58. Figure 3.8.4.3.8 Web UI for User: Unregistered User 112
59. Figure 3.8.4.3.9 Web UI for User: Donation Report 113
60. Figure 3.8.4.3.10 Web UI for User: Contact Us Page 114
61. Figure 3.8.4.3.11 Web UI for User: Give Your Testimonial Page 115
62. Figure 3.8.4.3.12 Web UI for User: Donate Fund 115
63. Figure 3.8.4.3.13 Web UI for User: Event Registration Form 116
64. Figure 3.8.4.3.14 Web UI for User: Logistical Aids 117
65. Figure 3.8.4.3.15 Web UI for User: Volunteer Registration form 118
66. Figure 3.8.4.3.16 Web UI for User: Login Page 119
67. Figure 3.8.4.3.17 Web UI for User: Registration Form Page 120
68. Figure 3.8.4.4.1 Web UI for Admin: Admin Login 121
69. Figure 3.8.4.4.2 Web UI for Admin: Super Admin Panel View Province 122
70. Figure 3.8.4.4.3 Web UI for Admin: Super Admin Panel View Admin Province 123
71. Figure 3.8.4.4.4 Web UI for Admin: Super Admin Panel Add Admin Province 124
72. Figure 3.8.4.4.5 Web UI for Admin: Super Admin Panel Edit Admin Province 125
73. Figure 3.8.4.4.6 Web UI for Admin: Super Admin Panel Approval Testimonial 126
74. Figure 3.8.4.4.7 Web UI for Admin: Super Admin Panel Approval Payment 127
75. Figure 3.8.4.4.8 Web UI for Admin: Super Admin Panel Layout Logo 128
76. Figure 3.8.4.4.9 Web UI for Admin: Super Admin Panel Layout Social Media 129
77. Figure 3.8.4.4.10 Web UI for Admin: Super Admin Panel Layout Banner 130
78. Figure 3.8.4.4.11 Web UI for Admin: Super Admin Panel Contact Us 131
79. Figure 3.8.4.4.12 Web UI for Admin: Super Admin Panel View Command Post 132
80. Figure 3.8.4.4.13 Web UI for Admin: Province Admin Panel View Disaster 132
81. Figure 3.8.4.4.14 Web UI for Admin: Province Admin Panel Add Disaster 134
82. Figure 3.8.4.4.15 Web UI for Admin: Province Admin Panel Edit Disaster 135
83. Figure 3.8.4.4.16 Web UI for Admin: Province Admin Panel View Admin 136
84. Figure 3.8.4.4.17 Web UI for Admin: Province Admin Panel Add Admin Disaster 137
85. Figure 3.8.4.4.18 Web UI for Admin: Province Admin Panel Edit Admin Disaster 138
86. Figure 3.8.4.4.19 Web UI for Admin: Province Admin Panel Testimonial 139
87. Figure 3.8.4.4.20 Web UI for Admin: Province Admin Panel View Command Post 140
88. Figure 3.8.4.4.21 Web UI for Admin: Province Admin Panel View Admin Co-Post 140
89. Figure 3.8.4.4.22 Web UI for Admin: Disaster Admin Panel Add Admin Co-Post 142
90. Figure 3.8.4.4.23 Web UI for Admin: Disaster Admin Panel Edit Admin Co-Post 143
91. Figure 3.8.4.4.24 Web UI for Admin: Disaster Admin Panel Testimonial 144
92. Figure 3.8.4.4.25 Web UI for Admin: Disaster Admin Panel View Co-Post 145
93. Figure 3.8.4.4.26 Web UI for Admin: Co-Post Admin Panel Evacuation Post Detail 146
94. Figure 3.8.4.4.27 Web UI for Admin: Co-Post Admin Panel Refugees Detail 147
95. Figure 3.8.4.4.28 Web UI for Admin: Co-Post Admin Panel Logistical Aids 148
96. Figure 3.8.4.4.29 Web UI for Admin: Co-Post Admin Panel Add Stock 149
97. Figure 3.8.4.4.30 Web UI for Admin: Co-Post Admin Panel Edit Stock 150
98. Figure 3.8.4.4.31 Web UI for Admin: Co-Post Admin Panel Add Item Needs 151
99. Figure 3.8.4.4.32 Web UI for Admin: Co-Post Admin Panel Approval 152
100. Figure 3.8.4.4.33 Web UI for Admin: Co-Post Admin Panel See Event Detail 153
101. Figure 3.8.4.4.34 Web UI for Admin: Co-Post Admin Panel See Volunteer Detail 153
102. Figure 3.8.4.4.35 Web UI for Admin: Co-Post Admin Panel Person Detail 154
103. Figure 3.8.4.4.36 Web UI for Admin: Co-Post Admin Panel Testimonials 155
104. Figure 3.8.4.4.37 Web UI for Admin: Co-Post Admin Panel View Co-Post Detail 156

## CHAPTER 1: PROPOSAL

### Introduction

Disaster is an event concentrated in time and space, in which a society or one of its subdivisions undergoes physical harm and social disruptions such that all or some essential functions of the society or subdivision are impaired (Fritz, 1961).Definition of disaster can be classified in three categories- classic, hazard, and socially focus (Perry, 2006).But according to bnbp.go.id, disaster is defined as events or series of events that threaten and disrupt people's lives and livelihoods caused by both natural factors and / or unnatural factors or human factors resulting in the occurrence of human casualties, environmental damage, property loss, and psychological impact. Thus, disaster could be caused by nature, human or other factors such as disease. Some of natural disaster that usually occurred in our country of Indonesia such as earthquake, tsunami, flood, volcanic eruption, landslide and drought.

Most of them had brought huge impact socially, geographically, economically even physiologically. It also needs longer time to recover from the effects of disaster. Hundreds and thousands house and building were damaged during disaster of Aceh Tsunami in 2004. More people died in seven different countries at the time. In 1883, 36.000 people were died due to Krakatau volcanic eruption (http://www.tandapagar.com). The world became darken for two days. The sun was dimming for a year and global climate change has taken place.

Long time ago, when the communication and information has not developed massively as nowadays, disaster recovery took such long time. But today, since the information has spread widely and fast. We can get faster information from the area or region which is thousand miles away from us. There were also more foundations which are able to conduct distribution aids to the victims in the disaster area. Helping and caring would become easier nowadays. But, the information we achieved could not be a valid information. The distribution of the aids also has not become inadequately for some region but becoming abundant for other region. The people also did not recognize clearly what are the needs of the refugees or victims in evacuation site due to the wrong understanding.

Therefore, we would like to solve these problems by using the development of technology. Then the people would accept the right information about the disaster and refugees, the evacuation

site or command post that needs more aids or help, and what are the refugees really need for themselves, in what kind of helps or aid should we give. We would like to create an information system which occupy these problems solving by the features and will be very useful for all kinds of user around the area in Indonesia.

### Issues in Current System

Bali volcano mount Agung alert level has raised since September 2017. People who lives near mount Agung have been asked to move to some evacuation sites which are safe from the eruption effects. According to tribunnews.com, the number of the refugees from 28 villages near mount Agung today is 185,865 people which is spread in 9 districts in Bali. Some of the command post or evacuation site have been built on behalf this situation. There are also some center evacuation sites in each district such as command post in Kompyang Sujana Field in Denpasar or Suwecapura center command post in Karangasem.

But some problems have appeared during this situation such as people do not know the command post which still needs logistics or helps while other command post is full of logistics. People do not know what are the needs of refugees, whether they needs food or clothes or other daily needs. The information which has been spread sometimes becoming invalid, because the information would be different for each media. Thus, we would like to build an information system in form of website based which would give the valid information about natural disaster occurred recently. It would also give the user information about each command post, the needs and the stocks in each post so the user can choose the post which still needs help and logistics. The user can also register as the volunteer and donator by completing some requirements. We hope through this information system will be very useful for each parties whether for the general users, volunteer, donator, refugees, and government.

### Benefits and Constraints of Proposed System

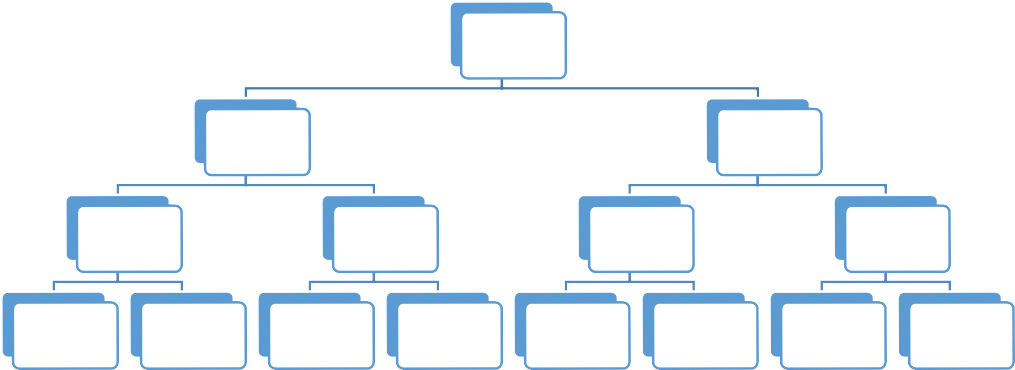
After seeing the problem of current issue above we planned to develop a web application that is able to solve those problems. By referring to our web application, people will know about what kind of logistical aids that is really need by particular command post (evacuation site),

whether they need additional volunteer or what kind of volunteer they are looking for. People also can get valid direction and information about this command post by using our application such as

: who is in charge of this command post, how many people are the refugees in this command post, and if they want to hold an event in this command post are the event are suitable with their ages. Even though this application might have some benefits there still some other things that should be solve such as the availability of internet connection and user acceptance. We know that in some area in Indonesia they were only limited and poor internet connection, it will make problem when the people want to use this application since this application need internet connection. The user acceptance is also important here, because we have to ensure that the interface of this website should be friendly to all of people from various kind of ages, whether it’s teenager, adult or elderly.

### Project Description

A web application that will be developed will included some functionalities such as authentication for admin and user. It means that the user will be able to register and login as member so that they can use the facilities of this web application. The admins would not be able to register themselves but should be added by the admin in the upper level, in this system the admin is differentiate by a hierarchical distribution.



Super Admin

Admin of

Province

Admin Of

Province

Admin Of

Disaster

Admin of

Disaster

Admin of

Disaster

Admin of

Disaster

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Admin of | Admin of | Admin of | Admin of | Admin of | Admin of | Admin of | Admin of |
| Command | Command | Command | Command | Command | Command | Command | Command |
| Post | Post | Post | Post | Post | Post | Post | Post |

*Figure 1.4.1.1 Hierarchy of Admins*

The upper level of admin has duty to add the lower level admin. The top-level admin is super admin, then comes to the lower level is admin of province, admin disaster and admin in each command post. One upper admin can approve or add the admin below them. As the example, the admin of disaster can approve or decline the candidates of admin of command post in their list or they can add the admin of command post by themselves.

The reason about creating this admin level because after doing the interview with BPBD Bali or Natural Disaster Countermeasure Department in Bali, we found that each province might be found different natural disaster because Indonesia is an archipelago that consists of many provinces and each province should be handled by an admin. If some disaster occurred and each kind of disaster will be handled by one or two admin that will add the admin of command post, the admin of command post here will be in charge of one command post and has duty to manage the stock, needs, volunteer and anything that is related with this command post.

The user will be able to participate to hold an event or being volunteer in particular command post by approval from the admin in that command post. The admin will manage them well.

### Project Aims and Objectives

Considering the high intensity of natural disasters occurring in Indonesia and the uneven distribution of logistic assistance to the post in the regions, MyPosko.com as a web-based information system of natural disaster has aim to provide the user (disaster recovery agency, society, volunteer or donator) information about the location of the available posts, post officers, details of the number of refugees and logistic aid that needed so that the logistical aid can be delivered appropriately. Not only to provide information, but also the user (society) able to contribute themselves as volunteer or donator in this social activities.

In other to achieve the project aims, there are some objectives that we set to develop MyPosko.com:

* + - Facilitate the user (disaster recovery agency, society, volunteer or donator) to determine which post is appropriate to be given logistical aid or still need some volunteer.

In the website, the user can view the detail of available post and they can decide which post that still need reinforcement in terms of logistical aid or volunteer.

* + - The admin of post can manage the data of logistical aid stock in and out.

In the application, the user (disaster recovery agency) can manage and update the current stock so they could estimate the availability of aid for per-day or per-week used to meet the needs.

* + - Facilitate the user (society) to participate in social activities as a volunteer or donator.

In the website, the user can apply their detail of identity to participate as a volunteer or donator. There are four types of volunteer such as kitchen, logistic, medical staff, and administration staff. Otherwise donator has three types such as donator of money, event and logistical aid.

* + - The administrator of post can easily giving out information related to posts.

In the website, the administrator can arrange the data of refugees and logistical aid that needed in each post so that the user can view the detail of post and distribute the aid evenly.

### Project Scope

This web application will handle five users, registered users and admins, that including super admin, admin of province, admin of disaster and command post admin. Here is the scope of our project:

* + - Authentication function for admin and users.
    - The user will be able to see the list of provinces, the list of disaster in particular province and the list of command post, and the detail of particular command post.
    - The user will be able to decide in which kind of logistical aid that they need to donate and which are need more volunteer or whether there has been an event or not in this particular command post.
      * The admin will have the duty to manage their own area, as the example, the admin of command post will be in charge to input the stock and needs, approving the candidate of volunteer and approving the event.
      * The admin of disaster will have duty to add admin of command post and manage them, each admin will have its own command post.
      * The admin of province will have duty to add admin of disaster, due to Indonesia consists of many provinces, we need to differentiate the disaster by their provinces.
      * Super admin will be able to add admin of province and approving testimonials from users about this application, and approving donation of money after receiving the receipt about some amount of money by user.

### Software and Hardware Requirements

* + 1. **Hardware**

This is the hardware which support us to finish the project.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Vendor** | **Hardware** | **Product Name** | **Usage** | **Cost** |
|  | Apple | Laptop | Macbook Pro Retina 2015 | Develop the project | Available |
|  | Apple | Smartphone | Iphone 7 | Web Testing | Available |
|  | Asus | Laptop | A456U | Develop the project | Available |
|  | Asus | Laptop | K45VD | Develop the project | Available |
|  | Asus | Smartphone | Zenfone 2 | Web Testing | Available |
|  | Canon | Printer | IP1800 | Print Document | Available |
|  | Samsung | Smartphone | J5 Pro | Web Testing | Available |
| **Hardware Cost** | | | | | **IDR 0,-** |

### Software

*Table 1.7.1.1 Hardware*

This is the operating system which we use for developing the project.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Developer** | **OS Name** | **Version** | **Usage** | **Cost** |
|  | Apple | Mac OS Sierra | 10.12.4 | Operating System | Available |
|  | Microsoft | Windows | 10 | Operating System | Available |
| **Operating System Cost** | | | | | **IDR 0,-** |

*Table 1.7.2.1 Operating System*

This is the list of software resource which we use when we create the project.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Developer** | **Application Name** | **Version** | **Usage** | **Cost** |
|  | Adobe System | Dreamweaver CC | 2015 | Text Editor and Designing the website | Available |
|  | Adobe System | Photoshop CC | 2015 | To create user interface design | Available |
|  | Apple | Safari | 10.1 | Web Testing | Available |
|  | Bitnami | Xampp | 7.1.4-0 | Localhost | Available |
|  | Gnome | Dia | 0.97.2 | Drawing Structured Diagrams | Available |
|  | Google | Chrome | 61.0.3163.100 | Web Testing | Available |
|  | Google | Drive | 1.31.2873.2759 | Data Transfer | Available |
|  | Microsoft | Word | 2013 | To create document | Available |
|  | Microsoft | Power Point | 2013 | To create presentation | Available |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Microsoft | Project | 2013 | To create project schedule | Available |
|  | Naver Corporation | Line | 4.10.0 | Communication | Available |
|  | Sublime | Text 3 | 3.0 | Text Editor for Programming | Available |
| **Software Cost** | | | | | **IDR 0,-** |

*Table 1.7.2.2 Application*

### Tools

This is the tools which help us to develop the website.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Developer** | **Application Name** | **Version** | **Usage** | **Cost** |
|  | Adobe System | Dreamweaver CC | 2015 | Text Editor and Designing the website | Available |
|  | Adobe System | Photoshop CC | 2015 | To create user interface design | Available |
|  | Apple | Safari | 10.1 | Web Testing | Available |
|  | Bitnami | Xampp | 7.1.4-0 | Localhost | Available |
|  | Gnome | Dia | 0.97.2 | Drawing Structured Diagrams | Available |
|  | Google | Chrome | 61.0.3163.100 | Web Testing | Available |
|  | Microsoft | Word | 2013 | To create document | Available |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Microsoft | Power Point | 2013 | To create presentation | Available |
|  | Microsoft | Project | 2013 | To create project schedule | Available |
|  | Sublime | Text 3 | 3.0 | Text Editor for Programming | Available |
| **Tools Cost** | | | | | **IDR 0,-** |

*Table 1.7.3.1 Tools*

### Resource Requirements

1. **Human Resource**

The qualified human resource is needed for developing a good application. We have a different ability to perform the tasks and we assign tasks according to our own ability. It is important to choose a person in the right position to do the task for making the human resource works best.

### Internet Resource

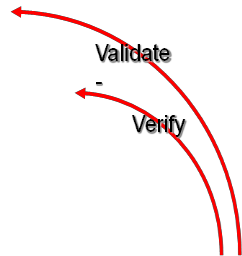
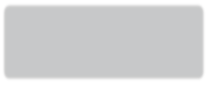
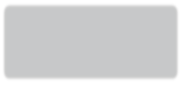
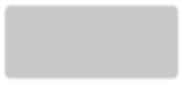
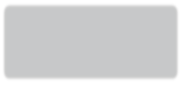
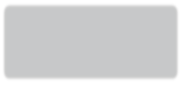
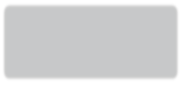
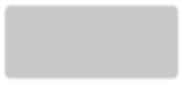
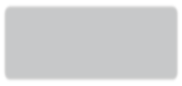
This is one of main resource which needed to access the information and analyze the disaster in Indonesia. We gather the information about disaster management, listing what is the problem in disaster management, learn what is the proses which they done in disaster management and learn all information which we need for doing this project.

### Stationary Resource

No matter what is the project is being done, we always need the stationary resources to support our documentation and description about the system. We need the stationary such as pen, pencil, eraser, paper, ruler, etc. we always write reminder, documentation all things that important for doing this project and we need to create a process overview about our system in the paper.

### Development Methodology

In developing a software product or application, choosing the right modelling life cycle process is very important. It is because based on model process selected, the development and testing process will be carried out. For the development of our project which has a title of MyPosko.com as an information system of logistical aid distribution using web-based, we are agreed to choose modelling process of **Waterfall with Prototyping**. Waterfall with prototyping model is a linear sequential design approach for software development and it is using verification and validation during the testing phase. The reason we choose waterfall with prototyping model is that this model has a sequential design approach which is easy to understand and manage. In addition, it is also applied the prototyping model which enable customers and developer to examine some aspect of the proposed system and decide if it is suitable or appropriate for the finished product.



Status Quo &

Problem Identification

Feasibility Study

Detailed Analysis

Detailed Design

Construction

Testing

**Prototyping**

System Delivery

Operation & Maintenance

*Figure 1.8.1.1 Waterfall with Prototyping Model Methodology*

There are two processes that conducted during the testing phase such as validation and verification. Validation ensures that the system has implemented all of the requirement, so that each system function can be traced back to a particular requirement in the specification. Then verification ensures that each function works correctly. (Charles P. Pfleeger, Shari Lawrence Pfleeger.2009).

### Techniques

There are several techniques that we use to gather information in order to develop MyPosko.com as information system of logistical aid distribution using web-based, such as:

### Interview

This technique allows us to gain information from interviewee about how the workflow that has been conducted in the event of natural disaster. To get the full information, we interview the people who is in charge of handling the natural disaster and also some volunteer and donator that have worked to help refugees. Based on the interview with volunteer and donator, we found many command post do not get their logistical aid evenly. It is caused by the lack of information to the public about command post that still need some reinforcement, for instances are logistical aid or volunteer. Those information from this technique becomes our consideration in developing our information system.

### Black and White Box Testing

Black and White box testing are used to test the software product or application from different view of user. Black box testing is used by the user who do not know about internal structure of code or program, mostly the user only test from the visual of software. They tends to find different kind of error such as:

1. Missing functions
2. Usability problem
3. Performance problem
4. Concurrency and timing errors
5. Initialization, termination errors and et cetera.

Then white box testing is used by the user who know about the internal structure of code or program. They are mainly focus on control flow or data flow of programs.

### Object Oriented Programming

Object Oriented Programming (OOP) is a programming paradigm which using the concept of "objects" that have data (attributes that describe objects) and procedures (functions) known as methods. The concept of OOP technique is to separate the program problems by using the object. Objects can be compared with 'stand-alone' special functions. To create an application, various objects will exchange data to achieve the final result. It means that each object is intended to work on a task, and produce the final value which can be displayed or used by other objects.

### CHAPTER 2: PROJECT MANAGEMENT PLAN

* 1. **Introduction**

In managing a project, it is important for us to create a work plan. Through a good work plan, the members will know the jobs for themselves, and when it should be started or ended. It is also important to know how long the project will be. The project manager also will be easier to control the works for other members. Due to this is a project which is used to complete a subject of BIT 304, the start time and end time is already decided by the college. We have 14 weeks to complete this project. The details of the schedules will be written in Gantt Chart and Work Item List.

### Work Item List and Work Break Down Structure

A work breakdown structure (WBS) outlines project work by decomposing the work activities into different levels of tasks (Kathy Schwalbe,2010). We have divided the work activities in our project into six different activities. Below are the detail of work breakdown structure and work item list.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task** | **Duration** | **Start Date** | **Finish Date** | **Task** | **Task Owner** |
| **1.0 Initiating** | **6 days** | **Friday, September 29, 2017** | **Friday, October 6,**  **2017** | **Task** | |
| 1.1 Kick off meeting | 1 day | Friday, September 29, 2017 | Friday, September 29,  2017 | Sub Task | Tommy,Sinthia, Daniya |
| 1.2 Inquire Idea | 6 days | Friday, September 29, 2017 | Friday, October 6, 2017 | Sub Task | Tommy,Sinthia, Daniya |
| 1.3 Consultation | 1 day | Monday, October 2,  2017 | Monday, October 2, 2017 | Sub Task | Tommy,Sinthia, Daniya |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1.4 Final Project Theme and Description Submission | 0 days | Friday, October 6, 2017 | | Milestone | |
| **2.0 Feasibility Study** | **10 days** | **Tuesday, October 3,**  **2017** | **Monday, October 16,**  **2017** | **Task** | |
| 2.1 Develop Project Background | 3 days | Monday, October 9,  2017 | Wednesday, October 11,  2017 | Sub Task | Daniya |
| 2.2 Develop Project Aims and Objectives | 3 days | Monday, October 9,  2017 | Wednesday, October 11,  2017 | Sub Task | Sinthia |
| 2.3 Develop Resource Plans | 3 days | Monday, October 9,  2017 | Wednesday, October 11,  2017 | Sub Task | Tommy |
| 2.4 Interview and Observation | 10 days | Tuesday, October 3,  2017 | Monday, October 16,  2017 | Sub Task | Tommy,Sinthia, Daniya |
| **3.0 Detailed Analysis** | **28 days** | **Thursday, October 12, 2017** | **Monday, November 20,**  **2017** | **Task** | |
| 3.1 Risk Management | 3 days | Thursday, October 12,  2017 | Monday, October 16,  2017 | Sub Task | Sinthia |
| 3.2 Develop Methodology and Techniques | 4 days | Thursday, October 12,  2017 | Tuesday, October 17,  2017 | Sub Task | Sinthia |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 3.3 Develop Project Schedule | 5 days | Thursday, October 12,  2017 | Wednesday, October 18,  2017 | Sub Task | Daniya |
| 3.4 Arranging Proposal | 7 days | Thursday, October 19,  2017 | Friday, October 27, 2017 | Sub Task | Daniya |
| 3.5 Proposal Submission | 0 days | Friday, October 27, 2017 | | Milestone | |
| 3.6 Literature Review | 6 days | Monday, October 30,  2017 | Monday, November 6,  2017 | Sub Task | Tommy,Sinthia, Daniya |
| 3.7 Requirements Gathering | 3 days | Tuesday, November 7, 2017 | Thursday, November 9,  2017 | Sub Task | Tommy,Sinthia, Daniya |
| 3.8 Preparing Interim Presentation | 16 days | Monday, October 30,  2017 | Monday, November 20,  2017 | Sub Task | Tommy,Sinthia, Daniya |
| 3.9 Interim Presentation | 0 days | Monday, November 20, 2017 | | Milestone | |
| **4.0 Detailed Design** | **4 days** | **Tuesday, November 21, 2017** | **Friday, November 24,**  **2017** | **Task** | |
| 4.1 Defining System Design | 4 days | Tuesday, November 21, 2017 | Friday, November 24,  2017 | Sub Task | Sinthia |
| 4.2 Designing Database | 3 days | Tuesday, November 21, 2017 | Thursday, November 23,  2017 | Sub Task | Daniya |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4.3 Designing Interface Website | 2 days | Tuesday, November 21, 2017 | Wednesday, November 22,  2017 | Sub Task | Tommy,Sinthia, Daniya |
| 4.4 System Design Defined | 0 days | Friday, November 24, 2017 | | Milestone | |
| 4.5 Database Defined | 0 days | Thursday, November 23, 2017 | | Milestone | |
| 4.6 Interface Website Defined | 0 days | Wednesday, November 22,  2017 | | Milestone | |
| **5.0 Construction** | **7 days** | **Monday, November 27, 2017** | **Tuesday, December 5,**  **2017** | **Task** | |
| 5.1 Coding | 7 days | Monday, November 27, 2017 | Tuesday, December 5,  2017 | Sub Task | Tommy,Sinthia, Daniya |
| 5.2 Develop Prototype | 3 days | Monday, November 27, 2017 | Wednesday, November 29,  2017 | Sub Task | Tommy,Sinthia, Daniya |
| **6.0 Closing** | **18 days** | **Thursday, November 23, 2017** | **Monday, December 18,**  **2017** | **Task** | |
| 6.1 Writing Final Report | 9 days | Wednesday, December 6, 2017 | Monday, December 18,  2017 | Sub Task | Tommy,Sinthia, Daniya |
| 6.2 Preparing Final Presentation | 9 days | Thursday, November 23, 2017 | Tuesday, December 5,  2017 | Sub Task | Tommy,Sinthia, Daniya |
| 6.3 Final Presentation | 0 days | Monday, December 18, 2017 | | Milestone | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 6.4 Submit Final Report | 0 days | Monday, December 18, 2017 | Milestone |

*Table 2.2.2.1 Work Item List*

6.1 Writing Final Report

6.2 Preparing

Final Presentation

6.3 Final Presentation

6.4 Submit Final Report

Final Project 1

1.0 Initiating

2.0

Feasibility Study

3.0 Detailed Analysis

4.0 Detailed Design

C

5.0

onstruction

6.0 Closing

1.1 Kick off Meeting

2.1 Develop Project Background

3.1 Risk Management

4.1 Defining System Design

5.1 Coding

1.2 Inquire Idea

2.2 Develop Project Aims and Objectives

3.2 Develop

Methodology and Techniques

5.2 Develop Prototype

1.3

Consultation

2.3 Develop Resource Plans

3.3 Arranging Proposal

4.3 Designing

Interface Website

1.4 Final Project Theme

and Description Submission

2.4 Interview and Observation

3.5 Proposal Submission

4.4 System

Design Defined

3.6 Literature Review

4.5 Database Defined

4.6 Interface

Website Defined

3.8 Preparing

Interim Presentation

3.9 Interim

Presentation

4.2 Designing

Database

3.7 Requirement

Gathering

*Figure 2.2.2.2. Work Breakdown Structure*

### Risk Management

Risk management plan is the process of identifying, assessing and taking steps to reduce the risk to acceptable level. Risk is uncertainty event that can have a negative or positive consequence. To ensure the project’s success, we need to determine the negative risk that may occur during development or maintenance of software project. It is important to be done because with risk management plan, it allows project manager with team member to take steps in dealing with potential risk by identifying, mitigating or avoiding the risk if they arise.

### Risk Identification

Identifying risk is the process of understanding what potential events that might hurt or enhance a particular project. It is important to list all the possible risk that may occur during the development of project so that we can be ready to response and reduce the negative effect of the risk. There are three information-gathering techniques that we use in identifying risk, they are:

1. Interviewing

Interviewing is a fact-finding technique that use to collect information in face to face, phone, email or instant-messaging documentation. Interviewing people who has a similar project experience is important so that we can view and checklist the possible problems that he or she had on a past project.

1. Brainstorming

Brainstorming is a technique that used by group to generate ideas or find a solution for specific problem. This technique can be done by amassing ideas spontaneously, doing research in internet, exchange opinion with project member about project development and et cetera. This approach is important to determine which risk is potentially to happen and know which risk response strategies to take such as risk avoidance, risk acceptance, risk transference or risk mitigation.

1. SWOT Analysis

SWOT Analysis is a technique of analyzing strengths, weakness, opportunities, and threat used to determine the internal and external factors that will impact the project future’s performance, whether it has a negative or positive effect.

### Risk Register

Risk register describes the result of risk identification process that contains a list of identified risk and other information that needed such as description, triggers, probability, impact, response strategy, root cause and risk owner. It is often displayed in a table or spreadsheet.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Risk | Description | Triggers | Probability | Impact | Response Strategy | Root Cause | Risk Owner |
| 1. | Server Down | The user and developer can’t access the website information because of down server | Many user access  the website | Medium | High | Data Backup | The capacity of server  ‘s bandwidth is too small. | User, Developer |
| 2. | Cyber Hacking | The website get hacked by someone and they change value in database | Having a bad intention to organizatio n or do it for self- satisfaction  . | Low | High | Do routine maintenance and create data backup | Weak security | User, Developer |
| 3. | Data corruption | Data has been crashed and can’t be found or viewed by user | Displaying the data that has been input to database | Medium | Medium | Validate the data input algorithm. | Wrong algorithm of data | User, Developer |
| 4. | Low internet connection | Weak internet connection | User want to view the updated | Low | High | Find other internet provider or place which | Unstable connection. | User |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | information in website |  |  | has a great connection |  |  |
| 5. | System Complexity | The user feel confuse in using the website’s features and accessing the information. | When user accesses the features and information of website. | Low | Low | Embrace the user to see the tutorial or giving socialization of using the website | Lack of knowledge about the function of the website as a source of natural disaster information | User |
| 6. | Bugs | Lots of bugs make an unstable access of website and data input problems | When user view the data of refugees and information  . | Medium | Medium | Examine the website and doing research with different developer that work in similar project. | Wrong logical programmi ng | User, Developer |

### Risk Analysis

*Table 2.3.2.1 Risk Register*

Qualitative risk analysis is the process of prioritizing risk for further analysis or action by assessing and combining their probability of occurrence and impact (Jorge Diego Fuentes Sanchez, 2013). Probability and impact matric is used to map the probability of risk occurrence and its impact if they arise. The benefit of this process is to enables the project manager and team member to focus on high priority of risk and reduce the impact and level of uncertainty.

### a. Probability and Impact Matrix Table

Probability and impact table describes the probability possibility in number from 1-3 that explained the level of risk occurrence gather with its impact that affects the cost, schedule, scope and quality of project. This is the probability and impact matrix based on risk register that have been planned.

y

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | R03 R06 | R01 |
| R05 |  | R02  R04 |

High



**P R O B A B I L I T Y**

Medium

Low

x

Low Medium High



**I M P A C T**

*Figure 2.3.3.1 Probability and Impact Matrix*

|  |  |  |
| --- | --- | --- |
| **Risk ID** | **Risk Name** | **Rank** |
| **R01** | Server down | 1 |
| **R03** | Data corruption | 2 |
| **R06** | Bugs | 3 |
| **R02** | Cyber hacking | 4 |
| **R04** | Low internet connection | 5 |
| **R05** | System complexity | 6 |

*Table 2.3.3.2 Rank of Risk*

### SWOT Analysis

SWOT analysis is a tool that examines the strength, weakness, opportunity and threat that may arise in developing the project. It identifies any opportunity from project strengths and threat from project’s weaknesses. The analysis also examines the level of project’s strengths offset threats as well as listing all the opportunities that may overcome the weaknesses.

### Strength

* + Capable to give complete information related to natural disaster such as the logistical aid that needed in each posts, location of post and the number of refugees in each post.
  + Increase communication and interaction of administrator, natural disaster recovery agency, and society.
  + Decrease the vagueness of information or hoax news about natural disaster in each area of post.
  + Increase the effectiveness of logistical aid distribution to each post which in need.
  + The responsive design of website can be accessed by user from their smartphone also laptop or PC.

### Weakness

* + Internet connection is needed to access the website, also the user need to spend extra amount of money to buy the package of internet.
  + At first trial, user may feel confuse about using the website.
  + The security and maintenance of website should be more strengthen.
  + Need to find administrator who understand about website and system information workflow.

### Opportunity

* + The number of similar project is still too few.
  + Many hoax news and wrong information about command post still spread out the public.
  + The accessible of information of natural disaster’s command post in each region is still less.
  + From this website, the distribution of logistical aid can be fixed and spread evenly.

### Threats

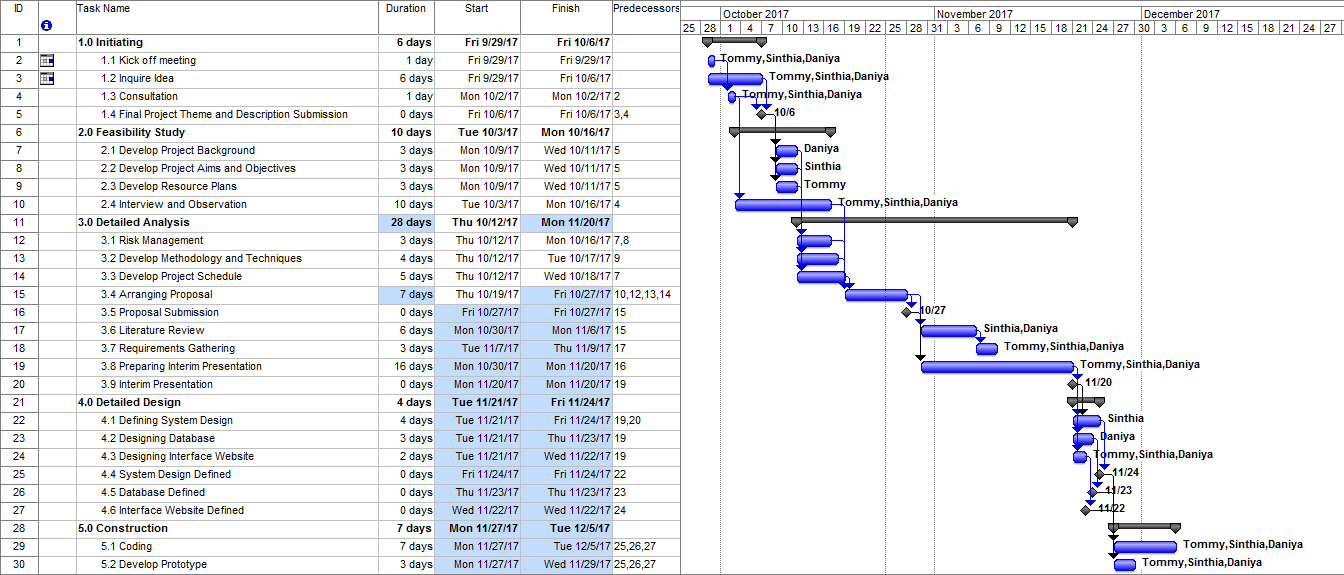
* + Cyber hacking that attack in unexpected time and condition.
  + Low bandwidth of hosting’s server makes the server of website is unconditionally down.
  + Lots of bug in website and data corruption can disturb the user to access the website for specific information or features.

### Gant Chart and Milestones

A Gantt chart is a standard format for displaying project schedule information by listing project activities and their corresponding start and finish dates in a calendar format (Kathy Schwalbe, 2010). By arranging a gantt chart, the project manager will be easier to inform other members about the details of starting date, end date, and who will be assigned in the tasks. It is also will be very useful to decide in which project should not be done late and which project can be start later. We use Microsoft Project 2007 to manage Gantt Chart. There are some elements in managing a gantt chart, they are :duration, start, finish, predecessor, and milestone. Duration is the range of time when a task is being done. Start is a start day of project. Finish is the time when the project should be ended and completed. Predecessor is a task which should be finished before the current task is started. Milestone is the completion of an activity- a particular in time(S. L Pfleeger,2009). Milestone is generally has 0 duration in gantt chart and symbolize by a diamond shape.

|  |  |  |
| --- | --- | --- |
| **Milestones** | **Tasks** | **Estimation Completion Date** |
| **Project Idea Approved** | Initiating | Monday 9th October 2017 |
| **Submit Project Plan** | Detailed Analysis | Friday 27th October 2017 |
| **Interim Presentation** | Detailed Analysis | Tuesday 21st November 2017 |
| **Database Defined** | Detailed Design | Friday 24th November 2017 |
| **System Design Defined** | Detailed Design | Monday 27th November 2017 |
| **Interface Website Defined** | Detailed Design | Thursday 23th November 2017 |
| **Submit Final Report** | Closing | Monday 18th December 2017 |

*Table 2.4.1.1 Milestone*





*Figure 2.4.1.2 Gant Chart / Schedule*

### References

Jorge Diego Fuentes Sanchez (2013).PMBOK Guide Fifth Edition. *A Guide to The Project Management Body of Knowledge*. (p. 328)

Kathy Schwalbe (2010). Information Technology Project Management Revised (p.

27,434-436).

Shari Lawrence PfLeeger, M.Atlee Joanne(2009).Software Engineering Theory and Practice – Fourth Edition.(p. 83, 119-121)

## CHAPTER 3: Requirement Analysis

### Introduction

In this chapter will be discussed about the features that available in natural disaster information system that followed by its functional and non-functional requirement. Functional and non-functional requirement are needed to define the requirement that user and admin want in operating natural disaster information system. Continued with analysis artifacts which defines the use case diagram of system. Use case diagram will be further explained in detail in form of high level use case, expanded use case and analysis cases diagram. Then continued to the design artifact that contains sequence diagram, database design, structural design and interface of natural disaster information system also be discussed in this chapter. The interface of system is not 100% presents the whole system in the future, it is due to the unexpected changes to fulfil the website requirement. The last is the implementation or testing of system to ensure the software that we made could handle the occurred natural disaster problem especially providing the information of available of post, refugees, volunteer, donator or aid that’s urgently needed in the post.

### Requirement Summary

In this system we developed for 5 users, they are registered users(member) or users that have been registered and had an account in this website, the admin of command post that in charge on particular command post (evacuation site), admin of disaster that is in charge in a disaster that occurred in particular province, admin of province that is in charge in a particular province, and super admin the highest level of admin.

A registered user can use function of login, logout, give donation in form of money or hold an event, or become a volunteer, he/she also can go to a particular command post by using function donate logistical aid, this function will give direction as we do in google maps. A command post admin will handle a command post and in charge to input the stocks of logistical aids and input the needs of which kind of volunteer in that command post and logistical aids that are needed in this command post. The admin of command post can also

approve the volunteer that users have submitted or the event that the users have submitted.

A disaster admin is in charge in a natural disaster that can use function to add admin of command post, thus the admin in command post does not register himself but need to be added by disaster admin. The admin of disaster can also update and remove the command post admin.

A province admin is in charge in particular province such as Bali, East Java, Aceh, DKI Jakarta and so forth, he is able to add, edit and delete a disaster in his/her province and also can manage the admin of its disaster, so the admin of disaster can not add himself but must be added by admin of province.

Super admin is the highest admin in admin hierarchy, he/she is able to add the admin of province and able to edit and remove them, she/he is also able to approve the testimonials that the user sent to this website about this application and approve the donation of money after receiving the donation receipt and matched it to the balance in bank account.

### Functional Requirements

Functional Requirement is used to know what operation and activities that a system must do and it should be able to define the features which offered by system to the user in detail.

* + 1. Register

This function is for unregistered users to register to be members of this web application. So that they can use more facilities to donate or to participate to be volunteer.

* + 1. Log in Users

This function is allowed users to go into the menu and use further facilities. By inputting the right username and password the users or admin will be allowed to go into the menus.

* + 1. Log in Admin

This function is for admin to log in to use their facilities by inputting their username and password. This function is only used by the admins not users.

* + 1. Log Out

This function is allowed the user and admin to log out. The users or admin will be brought to the home page.

* + 1. Edit Profile

This function is allowed the user to update their information or change their password or username.

* + 1. View List of Province

There will be menus to show list of articles, province, disaster or command post. Then, the users should choose ‘provinces’ then the system will show all the list of provinces.

* + 1. View List of Disaster

The users or admins should choose one of the province first then it will directly show the list of all disaster in that province.

* + 1. View List of Command Post

There will be menus to show list of articles, province, disaster or command post. Then, the users should choose ‘command post’ then the system will show all the list of command posts.

* + 1. Donate money

This function is for the users to donate money through this web application. They can transfer their money, and send the receipt. So that super admin can check it manually. Once their donation is confirmed, the donator’s name will be shown in the web application.

* + 1. Donate event

This function is for the users to hold an event in particular command post. The donator should fill the form and submit it to web application and it still needs approval from admin in command post.

* + 1. Donate Logistical Aids

This function is for the user who want to donate in form of goods, they can get the direction to go to that command post. So, it will be easier for the user.

* + 1. Become a volunteer

This function is for the users to participate to be a volunteer. They should fill the form about their skill and the part of command post that they want to join, whether they want to be medical staff, kitchen staff, administration staff or rescue staff.

* + 1. Update Details of Command Post

This function will be allowed the admin in command post to update the details of information in particular command post. At first, when an admin of command post is

added then the information about command post is still empty, so that the admin needs to update it.

* + 1. Add Stock in Command Post

The admin in command post will be allowed to add goods in stock in command post.

* + 1. Add Needs in Command Post

The admin in command post will be allowed to add goods in needs in command post.

* + 1. Update Stock in Command Post

The admin will be allowed to update the stock in command post by adding or subtracting the amount or the goods in stock

* + 1. Update Needs in Command Post

The needs in command post will be automatically subtracted when the stock is added but the updating manually of needs can be done.

* + 1. Delete Stock in Command Post

The admin will be allowed to delete any goods in stocks.

* + 1. Delete Needs in Command Post

The admin will be allowed to delete any goods in needs.

* + 1. Event Approval

This function is to be done for admin of command post to approve or decline the event that is already submitted by donator. Once the event is approved by the admin of command post, it will be directly shown up in command post details.

* + 1. Volunteer Approval

This function is for the admin to accept or decline volunteer in their command post. Once the volunteer is approved by the admin, then the names of volunteer will be on the list of volunteers in command post details.

* + 1. Add Admin Command Post

The admin of disaster has the right to add admin in command post.

* + 1. Edit Admin Command Post

The admin of disaster has the right to edit admin in command post.

* + 1. Delete Admin Command Post

The admin of disaster has the right to delete admin in command post.

* + 1. Add Natural Disaster

The admin of province will be allowed to add disaster when a disaster is occurred.

* + 1. Edit Natural Disaster

The admin of province will be allowed to edit/update disaster.

* + 1. Delete Natural Disaster

The admin of province will be allowed to delete disaster.

* + 1. View Admin of Disaster

This function is for the admin of province to see the list of admins in particular disaster, due to there will be more than one admin in each disaster.

* + 1. Add Admin of Disaster

The admin of province will be allowed to add admin of disaster and completed his/her details.

* + 1. Edit Admin of Disaster

The admin of province will be allowed to edit admin of disaster and his/her information.

* + 1. Delete Admin of Disaster

The admin of province will be allowed to edit admin of disaster and his/her information.

* + 1. Add Admin of Province

The Super admin will be allowed to add admin of province and completed his/her information.

* + 1. Edit Admin of Province

The Super admin will be allowed to edit admin of province and his/her information.

* + 1. Delete Admin of Province

The Super admin will be allowed to delete admin of province.

* + 1. Submit Testimonials

All users or admin are allowed to submit their testimonials about this web application by filled up their name, position, and content.

* + 1. Approve Testimonials

The super admin will approve some testimonials to be shown in the home page.

* + 1. Payment Approval

This function is for the super admin to check the payment which is done in this web application. Once the payment is confirmed, the donator will receive notification, and his/her name will be shown in donation report.

### Non-functional Requirements

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Priority** | **Description** |
| Functionality | High | Developer ensures the web of natural disaster information system has no bugs which can slow down the process inside the web application |
| Portability | High | Natural Disaster information system is expected to run in various platform of gadgets such as smartphone, laptop or personal computer. |
| Usability | High | The web site is expected to be easy to use in order to accommodate the user about the updated information of available post with its detail information, current status of natural disaster, and volunteer or donator recruitment. |
| Reliability | High | The access time of natural disaster information system is likely to be operated in 24 hours but with little rest for server to handle the unexpected occasion of down server. |

|  |  |  |
| --- | --- | --- |
| Integrity | High | The validity of data stored and display in the website will be highly assured for the user. It is because several approval from some admin is needed to release the post or comment related to natural disaster. |
| Efficiency | High | This web site will provide command post information in a province which chosen by user who wants to give aids and of course they can spend shorter time rather than searching the information in traditional way. |
| Backup | High | The natural disaster information system is expected to conduct backup at once in a month to prevent the data loss. |

* 1. **Iteration Plan**

*Table 3.4.1* Non-Functional Requirement

To ensure software development can be worked properly without any obstacle, it is recommended to use iteration plan. Iteration plan is the plan which uses repetition process in certain stage of methodology. The repetition process is beneficial to assure the requirement of project already in accordance with the goal. Moreover, it has a function to eliminate the risk impact that might attack during the process of project development. Therefore in this section, the iteration plan will be described based on the methodology we implement in the project, it is Waterfall with prototyping.

### Feasibility Study Analysis

Iteration plan will be conducted in this stage of feasibility of study if there are some insufficient required resources that have been planned to build natural disaster information system. The reason is that it could impact the project to cost more or slowing down the development of project. Another reason is when we face the inability to list all requirement and information related to natural disaster. The type of information are the workflow of logistical aid stock management, permission to build co-post, number of refugees and et cetera. The repetition process that we do are focusing in interviewing some volunteer or donator that has visited the natural disaster co-post and doing observation to people who work and has responsibility in recovering the natural disaster in Bali area to get more accurate and validated information about natural disaster. Project initiation and planning are also iterated in this stage to ensure the quality of software development.

### Detailed Analysis

Iteration plan will be conducted in this stage of detailed analysis if there are some possible additional risk that might attack during project development. The other step to be concern are project schedule and techniques which will also be evaluated if needed. Project schedule contains the duration of each activity needed to complete the project. It will be evaluated again if any member needs expanded duration or shorten the time for certain activity. Then for techniques is focused on how we collect some information for the future functional and non-functional requirement which will be offered to the user. The techniques will be evaluated if there is possibility of other techniques that will make the time of requirement gathering to be more efficient.

### Detailed Design

Iteration plan will be conducted in this stage of detailed design if the design specification is not fulfill the required goal of project. The design specification of system are including the functional and non-functional requirement, database design, high level use case, expanded use case, analysis diagram, sequence diagram, and design of web interface. The repetition process of design specification need to be implemented in order to produce the best of result and fulfill the user satisfaction.

### Construction

Iteration plan will be conducted in this stage of construction if there is any fault or unexpected result during the prototype testing. To overcome it, during the coding of prototype will be evaluated for several times until it can show the basic functional and performance of the future system.

### Testing

Iteration plan will be conducted in this stage of testing if the prototype still need evaluation especially for detailed analysis and detailed design. During the testing, validation and verification will be implemented to examine the system. Validation ensures all requirement are met and verification checks correctness of each function.

### System Delivery

Iteration plan will be conducted in this stage of system delivery if there is any additional requirement which needed to be implemented in system. After the deliverable and all documentation has met the project goal then the next stage of operation and maintenance can be continued.

### Operation & Maintenance

Iteration plan will be conducted in this stage of operation and maintenance if there is any unexpected bugs or data loss attack in the system. The iteration plan is needed to ensure the system can be worked properly without any possible bugs or data loss that could slow down the process of accessing the information in our system.

This is the list of use cases of our natural disaster information system or called as MyPosco.com and in this documentation of Final Project 304, we have done the iteration of 3 use cases. There are register user, login user and login all admin.

|  |  |  |  |
| --- | --- | --- | --- |
| **Iteration** | **Use Cases** | **Proposed Start Date** | **Proposed End Date** |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | Register User | November, 24th 2017 | December, 4th 2017 |
| 2. | Login User | November, 24th 2017 | December, 4th 2017 |
| 3. | Login All Admin | November, 24th 2017 | December, 4th 2017 |
| 4. | Log out | February, 5th  2017 | February, 6th  2017 |
| 5. | Edit Profile | February, 5th  2018 | February, 9th  2018 |
| 6. | View List of Province | February, 5th  2018 | February, 9th  2018 |
| 7. | View List of Natural Disaster | February, 5th  2018 | February, 9th  2018 |
| 8. | View List of Command Post | February, 5th  2018 | February, 9th  2018 |
| 9. | Donate Money | February, 12th 2018 | February, 15th 2018 |
| 10. | Donate Event | February, 12th 2018 | February, 15th 2018 |
| 11. | Donate Logistical Aids | February, 12th 2018 | February, 15th 2018 |
| 12. | Become a volunteer | February, 16th  2018 | February, 19th 2018 |
| 13. | Update Details of Command Post | February, 16th  2018 | February, 19th 2018 |

|  |  |  |  |
| --- | --- | --- | --- |
| 14. | Add stock in Command post | February, 16th  2018 | February, 19th 2018 |
| 15. | Add needs in command post | February, 16th  2018 | February, 19th 2018 |
| 16. | Update Stock in command post | February, 20th  2018 | February, 23th 2018 |
| 17. | Update needs in command post | February, 20th  2018 | February, 23th 2018 |
| 18. | Delete stock in command post | February, 23th  2018 | February, 27th 2018 |
| 19. | Delete needs in command post | February, 23th  2018 | February, 27th 2018 |
| 20. | Event approval | March, 1st 2018 | March, 5th  2018 |
| 21. | Volunteer approval | March, 1st 2018 | March, 5th  2018 |
| 22. | Add admin of command post | March, 6th 2018 | March, 10th  2018 |
| 23. | Edit admin of command post | March, 6th 2018 | March, 10th  2018 |
| 24. | Delete admin of command post | March, 6th 2018 | March, 10th  2018 |
| 25. | Add natural disaster | March, 12th  2018 | March, 15th  2018 |
| 26. | Edit natural disaster | March, 12th  2018 | March, 15th  2018 |

|  |  |  |  |
| --- | --- | --- | --- |
| 27. | Delete natural disaster | March, 12th  2018 | March, 15th  2018 |
| 28. | View admin of natural disaster | March, 16th  2018 | March, 17th  2018 |
| 29. | Add admin of natural disaster | March, 19th  2018 | March, 23th  2018 |
| 30. | Edit admin of natural disaster | March, 19th  2018 | March, 23th  2018 |
| 31. | Delete admin of natural disaster | March, 19th  2018 | March, 23th  2018 |
| 32. | Add admin of province | March, 26th  2018 | March, 30st  2018 |
| 33. | Edit admin of province | March, 26th  2018 | March, 30st  2018 |
| 33. | Delete admin of province | March, 26th  2018 | March, 30st  2018 |
| 35. | Submit testimonials | April, 2st 2018 | April, 5th  2018 |
| 36. | Approve testimonials | April, 2st 2018 | April, 5th  2018 |
| 37. | Payment Approval | April, 2st 2018 | April, 5th  2018 |

*Table 3.5.8 Iteration Plan*

### Iteration 1

* + 1. **Introduction**

The goal of doing iteration is to ensure the developed use cases can be working properly without any interference or debug that might slow down our information system of natural disaster. At this moment, we have managed to build a prototype of our system along with 3 functionality there are register user, login user and login all admin. It all work properly and fortunately can be done on time based on our Gantt chart plan.

### Purpose

The result of our iteration plan based on the waterfall with prototyping methodology is in accordance with what we have expected. From the feasibility study phase, we have done the project background, project aims and objectives, resource plan, interview and observation. Through the iteration process, we can obtain the complete information not only by holding a meeting with our mentor in the college but also from the disaster agency that really help us in determining the requirement of system. Then, in detailed analysis phase, we have finished the risk management, develop methodology and techniques, proposal, proposal submission, literature review, requirement gathering, and interim presentation. Through the iteration of this phase, we can analyse the risk of system openly from different point of view of our team member, and we are able to choose the appropriate methodology and technique to satisfy the user expectation and also admin. Continued with the detailed design phase, we have done system design, database design and interface of website. From this iteration, we have learned to make a user friendly interface and design specification of our project properly. In construction phase, we have done the iteration for the coding process and also prototype. In our prototype we have three use cases that are already worked such as register of user, login user and login admin. Through the iteration, we have learned to manage our time to code the project and decreases the debug that emerge in unexpected time. In testing phase, we need to ensure that our prototype can show the basic functionality and performance of the future system. So the iteration plan has been implemented to know whether the system already pass the criteria of validation and verification which means that, all the use cases that implemented has worked correctly and has met all the requirement from user and admin. In system delivery and operation & maintenance phase, we have not implemented the iteration plan because these

work is to be done in our final project 2 of BIT 305. Therefore the iteration that has been implemented is started from the feasibility of study until testing phase.

### Context

The implementation of iteration plan to our project is really helping us to determine the appropriate requirement both of design and specification of our information system. Began with feasibility study until the testing phase where we check the correctness of each functionality that we provide in our prototype, there are register for user, login for user and login for all admin. The login admin has separated into 4 category, there are login of super admin, login of province admin, login of natural disaster admin, and login of command post admin. The differences of admin is made based on the interview and observation that we conduct during the feasibility study phase with one of coordinator in disaster recovery agency, BPBD Bali. It is because, each of admin has their own functionality that support each other in dealing with the disaster recovery problem. Through the iteration plan, not only to decide the requirement, but also we can improve our skill in coding the project. Eventhough this is only the basic functionality of register and login, but through the iteration, we can specified the requirement of login and register form design based on user information that is needed and also admin identity that is needed because both of them has specific functionality that differ to one another.

### Schedule of Iteration Workflows

|  |  |  |  |
| --- | --- | --- | --- |
| **Workflow** | **Start Date** | **End Date** | **Duration (days)** |
| **Use Case 1: Register User** | November, 24th 2017 | December, 4th 2017 | 10 days |
| Analysis | November, 24th 2017 | December, 4th 2017 | 10 days |
| Design | November, 24th 2017 | December, 4th 2017 | 10 days |
| Implementation | November, 24th 2017 | December, 4th 2017 | 10 days |

|  |  |  |  |
| --- | --- | --- | --- |
| Testing | November, 24th 2017 | December, 4th 2017 | 10 days |
| **Use Case 2: Login User** | November, 24th 2017 | December, 4th 2017 | 10 days |
| Analysis | November, 24th 2017 | December, 4th 2017 | 10 days |
| Design | November, 24th 2017 | December, 4th 2017 | 10 days |
| Implementation | November, 24th 2017 | December, 4th 2017 | 10 days |
| Testing | November, 24th 2017 | December, 4th 2017 | 10 days |
| **Use Case 3: Login Admin** | November, 24th 2017 | December, 4th 2017 | 10 days |
| Analysis | November, 24th 2017 | December, 4th 2017 | 10 days |
| Design | November, 24th 2017 | December, 4th 2017 | 10 days |
| Implementation | November, 24th 2017 | December, 4th 2017 | 10 days |
| Testing | November, 24th 2017 | December, 4th 2017 | 10 days |

*Table 3.6.4.1 Iteration Workflow Schedule*

There is no slip date when develop all these use cases and we can provide it all in our prototype on time. This is because we have done the iteration in feasibility study that useful to gather the full information from the mentor in college also from the disaster agency’s delegation.

### Iteration Schedule Breakdown

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **Assigned To** |
| Use Case 1: Login All Admin | November 24, 2017 | December 4, 2017 | Tommy,Daniya,Sinthia |
| **1.1 Analysis** | November 24, 2017 | December 4, 2017 | Daniya,Sinthia |
| 1.1.1 Analysis of login form requirement for admin | November 24, 2017 | December 4, 2017 | Daniya,Sinthia |
| **1.2 Design** | November 24, 2017 | December 4, 2017 | Tommy |

|  |  |  |  |
| --- | --- | --- | --- |
| 1.2.1 Design of login form | November 24, 2017 | December 4, 2017 | Tommy |
| **1.3 Implementation** | November 24, 2017 | December 4, 2017 | Tommy |
| 1.3.1  Implementation of login form in website | November 24, 2017 | December 4, 2017 | Tommy |
| **1.4 Testing** | November 24, 2017 | December 4, 2017 | Sinthia |
| 1..4.1 Testing of login of all admin | November 24, 2017 | December 4, 2017 | Sinthia |

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **Assigned To** |
| Use Case 1: Login User | November 24, 2017 | December 4, 2017 | Tommy,Daniya,Sinthia |
| **1.1 Analysis** | November 24, 2017 | December 4, 2017 | Daniya,Sinthia |
| 1.1.1 Analysis of login form requirement for admin | November 24, 2017 | December 4, 2017 | Daniya,Sinthia |
| **1.2 Design** | November 24, 2017 | December 4, 2017 | Tommy |
| 1.2.1 Design of login form | November 24, 2017 | December 4, 2017 | Tommy |
| **1.3 Implementation** | November 24, 2017 | December 4, 2017 | Tommy |
| 1.3.1 Implementation of login form in website | November 24, 2017 | December 4, 2017 | Tommy |
| **1.4 Testing** | November 24, 2017 | December 4, 2017 | Daniya |
| 1..4.1 Testing of login of user | November 24, 2017 | December 4, 2017 | Daniya |

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **Assigned To** |
| Use Case 1: Register User | November 24, 2017 | December 4, 2017 | Tommy,Daniya,Sinthia |
| **1.1 Analysis** | November 24, 2017 | December 4, 2017 | Tommy, Sinthia |
| 1.1.1 Analysis of register requirement for user | November 24, 2017 | December 4, 2017 | Tommy,Sinthia |
| **1.2 Design** | November 24, 2017 | December 4, 2017 | Daniya |
| 1.2.1 Design of register form | November 24, 2017 | December 4, 2017 | Daniya |
| **1.3 Implementation** | November 24, 2017 | December 4, 2017 | Tommy |
| 1.3.1 Implementation of register form in website | November 24, 2017 | December 4, 2017 | Tommy |
| **1.4 Testing** | November 24, 2017 | December 4, 2017 | Sinthia |
| 1..4.1 Testing of register user | November 24, 2017 | December 4, 2017 | Sinthia |

*Table 3.6.5.1 Iteration Plan Task Breakdown by Workflow*

### Resource Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Resources | Usage | Quantity | Cost |
| Hardware | | | |  |
| 1 | Computer (MacOs Sierra version 10.12.6, Windows 8,  Windows 10,) | Development of web application and testing | 3 computers | Rp0 (Existing computers) |
| Software | | | |  |
| 2 | Sublime Text 3, XAMPP and Adobe Dreamweaver | Development of front end and back end of natural disaster information system | 3 platform | Rp0 ( Open source) |
| 3 | Browser (Chrome, Safari) | Conduct testing to ensure there is no bug in the website and design specification of user and admin has met. | 2 platform | Rp0 |
| Human Resources | | | |  |
| 3 | Team Member | Conduct testing of prototype system | 1 member | Rp0 (Include in team  project) |

*Table 3.6.6.1 Iteration Workflow Schedule*

### Evaluation Criteria

Based on the methodology of waterfall with prototyping we implement, we have several phase from feasibility study to operation and maintenance stage. In feasibility study phase, we have a criteria to have a clear understanding of specification that we should be performed in our system. Therefore we are able to build up the system and also documentation from the same understanding and miscommunication between team is able to be decreased. The result of the iteration is project background, project aims, objectives, resource plan, interview and observation can be done on time as the milestone we set. In detailed analysis phase, the result of deliverable is risk management, methodology, technique used to develop the system, literature review and requirement gathering. All deliverable has been achieved and finished on time by team member. In detailed design phase, the iteration is implemented to define the appropriate design for user-friendly interface of system and its specification fulfil the user and admin expectations. Fortunately, all design has been completed on the specified time frame based on our project schedule. In construction phase, we have set three use cases that we want to implement in our prototype. There are register and login for user, and login for all admin. All these use cases has been implemented as the criteria of our prototype system. Continued

with the testing phase, the criteria of iteration of this phase is when all the use cases implemented and design of website has been fulfilled what we have expected. In testing phase, we have done validation and verification of system and finally can achieve the criteria of testing phase where all use cases has been worked properly. For the system delivery phase and operation and maintenance, we will do it as the future work of BIT 305 or Final Project 2. The evaluation criteria for iteration of both phases, whether our system has fulfilled the expectation of user and admin especially from the disaster recovery agency and operation and maintenance of system can be operated easily by the user and secured the system from the unexpected risk that might attack.

### Iteration Review and Evaluation

Iteration of each phase except for system delivery and operation & maintenance has been met the specified time frame we set. Iteration plan has helped us in determining the requirement and specification of system. Started from the feasibility study to testing phase, the result has been in accordance with our expectation. Eventhough there is some bug that emerge during the testing phase iteration but fortunately we can overcome it and finished the protoype on time.

### Analysis Artifacts

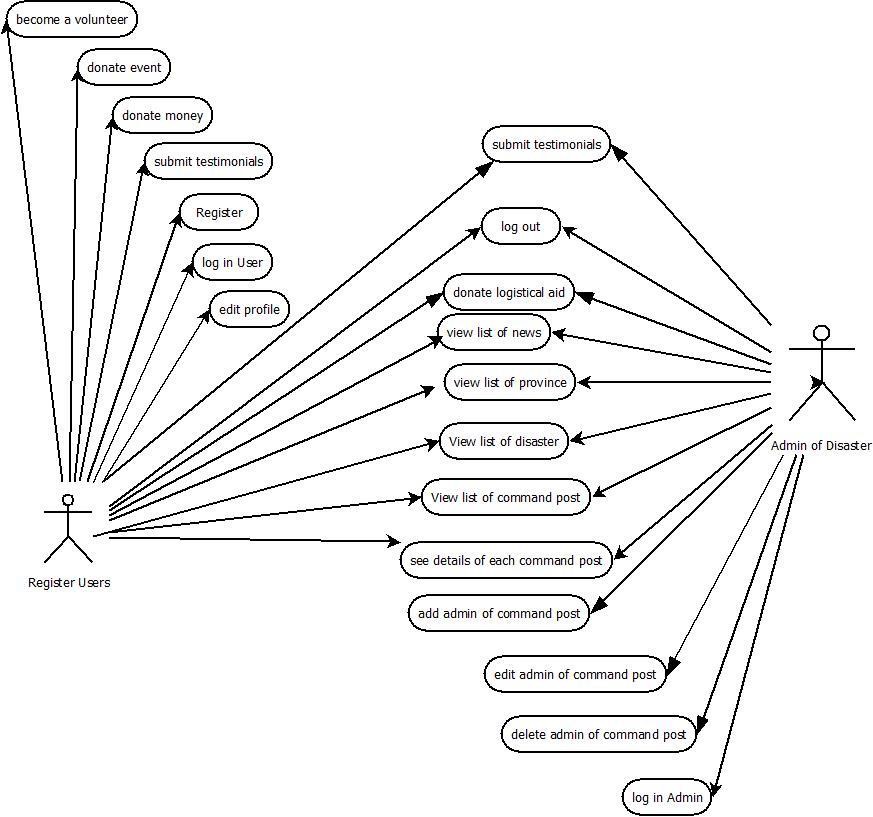
* + 1. **Use Case Diagram**



*Figure 3.7.1.1 Use Case Diagram of Registered User and Super Admin*



*Figure 3.7.1.2 Use Case Diagram of Registered User and Province Admin*



*Figure 3.7.1.3 Use Case Diagram of Registered User and Natural Disaster Admin*



*Figure 3.7.1.4 Use Case Diagram of Registered User and Command Post Admin*

### High Level Use Case

|  |  |
| --- | --- |
| Use Case | Register |
| Actor(s) | Users |
| Description | The candidate of members should fill up the forms in order to be registered users and use more facilities in this web application. The candidate of users should fill up the forms given, then they will get notification about their username and password that are already active. |

*Table 3.7.2.1 High Level Use Case of Register*

|  |  |
| --- | --- |
| Use Case | Log in User |
| Actor(s) | Registered Users |
| Description | The registered users should input the correct username and passwords to log in. If the log in successes, then the users can use their facilities or menus. |

*Table 3.7.2.2 High Level Use Case of Log in User*

|  |  |
| --- | --- |
| Use Case | Log in Admin |
| Actor(s) | Admin |
| Description | The admins should input the correct username and passwords to log in. If the log in successes, then admin can use their facilities or menus. |

*Table 3.7.2.3 High Level Use Case of Log In Admin*

|  |  |
| --- | --- |
| Use Case | Log out |
| Actor(s) | Registered Users and Admins |
| Description | The registered users or admins can log out if they have finished to use the menus and they will be moved to the home page after logging out. |

*Table 3.7.2.4 High Level Use Case of Log Out*

|  |  |
| --- | --- |
| Use Case | Edit Profile |
| Actor(s) | Registered Users |
| Description | The registered users can edit their information including change username and password. |

*Table 3.7.2.5 High Level Use Case of Edit Profile*

|  |  |
| --- | --- |
| Use Case | View List of Province |
| Actor(s) | Users and Admins |
| Description | The user or admin will be able to see list of provinces in Indonesia. By clicking the menu ‘provinces’ then the list of all provinces will be shown up. |

*Table 3.7.2.6 High Level Use Case of View List of Province*

|  |  |
| --- | --- |
| Use Case | View List of Natural Disaster |
| Actor(s) | Users and Admins |
| Description | The user or admin will be able to see list of disaster in a province. By clicking one of the province from the list of disasters, then the disaster in that province will appear. |

*Table 3.7.2.7 High Level Use Case of View List of Natural Disaster*

|  |  |
| --- | --- |
| Use Case | View List of Command Post |
| Actor(s) | Users and Admins |
| Description | The user or admin will be able to see list of command post by clicking menu “Command Post”, then the list of command post will appear. |

*Table 3.7.2.8 High Level Use Case of View List of Command Post*

|  |  |
| --- | --- |
| Use Case | Donate money |
| Actor(s) | Registered Users |
| Description | This is one of the facilities that given to registered users. They will be given the account bank number to transfer donation and forms to be filled up that they have donate money. Then the super admin will check it at give the report after that. |

*Table 3.7.2.9 High Level Use Case of Donate Money*

|  |  |
| --- | --- |
| Use Case | Donate event |
| Actor(s) | Registered Users |
| Description | This is one of the facilities that given to registered users. They can hold an event in particular command post by filling up the forms and it will be submitted to particular admin of command post. |

*Table 3.7.2.10 High Level Use Case of Donate Event*

|  |  |
| --- | --- |
| Use Case | Donate Logistical Aids |
| Actor(s) | Users and Admins |
| Description | This function is for the user who want to donate in form of goods, they can get the direction to go to that command post. So, it will be easier for the user. |

*Table 3.7.2.11 High Level Use Case of Donate Logistical Aid*

|  |  |
| --- | --- |
| Use Case | Become a volunteer |
| Actor(s) | Registered Users |
| Description | This is one of the facilities that given to registered users. They can be a volunteer by filling up the forms about their skill and in which part they want to take part in that command post. |

*Table 3.7.2.12 High Level Use Case of Become a Volunteer*

|  |  |
| --- | --- |
| Use Case | Update Details of Command Post |
| Actor(s) | Admin of Command Post |
| Description | Only admin of command post will be given this facility. The button of ‘edit’ will be place in the bottom part of details information of command post. After clicking it, the admin of command post will be given the forms to update information, and save it to database. |

*Table 3.7.2.13 High Level Use Case of Update Details of Command Post*

|  |  |
| --- | --- |
| Use Case | Add Stock in Command Post |
| Actor(s) | Admin of Command Post |
| Description | This facility is given to admin in Command Post. The admin in command post can add the stock of goods in that command post. It will be appeared in details of command post |

*Table 3.7.2.14 High Level Use Case of Add Stock in Command Post*

|  |  |
| --- | --- |
| Use Case | Add Needs in Command Post |
| Actor(s) | Admin of Command Post |
| Description | This facility is given to admin in Command Post. The admin in command post can add the needs of goods or volunteer or other things in that command post. It will be appeared in details of command post. |

*Table 3.7.2.15 High Level Use Case of Add Needs in Command Post*

|  |  |
| --- | --- |
| Use Case | Update Stock in Command Post |
| Actor(s) | Admin of Command Post |
| Description | This facility is given to admin in Command Post. The admin in command post can edit the stock of goods in that command post by adding or subtracting them. The needs of command post will be automatically subtracted when the stock in the same goods are added. |

*Table 3.7.2.16 High Level Use Case of Update Stock in Command Post*

|  |  |
| --- | --- |
| Use Case | Update Needs in Command Post |
| Actor(s) | Admin of Command Post |
| Description | This facility is given to admin in Command Post. Actually, the needs of command post will be automatically updated if the stock with the same goods is also updated, but the admin can still update it manually, if they want to reset or add or subtract. |

*Table 3.7.2.17 High Level Use Case of Update Needs in Command Post*

|  |  |
| --- | --- |
| Use Case | Delete Stock in Command Post |
| Actor(s) | Admin of Command Post |
| Description | This facility is given to admin in Command Post. The admin can delete any stock in his/her own command post. |

*Table 3.7.2.18 High Level Use Case of Delete Stock in Command Post*

|  |  |
| --- | --- |
| Use Case | Delete Needs in Command Post |
| Actor(s) | Admin of command post |
| Description | The admin will choose one of command post then system will display the information needs and delete it as they need. |

*Table 3.7.2.19 High Level Use Case of Delete Needs in Command Post*

|  |  |
| --- | --- |
| Use Case | Event Approval |
| Actor(s) | Admin of command post |
| Description | The admin of command post can approve an event that is submitted to be held in his/her command post. |

*Table 3.7.2.20 High Level Use Case of Event Approval*

|  |  |
| --- | --- |
| Use Case | Volunteer Approval |
| Actor(s) | Admin of command post |
| Description | The admin can accept or remove the volunteer that requested to take part in his/her command post. |

*Table 3.7.2.21 High Level Use Case of Volunteer Approval*

|  |  |
| --- | --- |
| Use Case | Add admin of command post |
| Actor(s) | Admin of natural disaster |
| Description | The admin fills the form of command post admin registration after the submission, then the system will save the data to database as a new member of admin command post |

*Table 3.7.2.22 High Level Use Case of Add Admin of Command Post*

|  |  |
| --- | --- |
| Use Case | Edit admin of command post |
| Actor(s) | Admin of natural disaster |
| Description | The admin chooses one of command post, then the system display the detail of post and admin of natural disaster edit the information of admin as they need. |

*Table 3.7.2.23 High Level Use Case of Edit Admin of Command Post*

|  |  |
| --- | --- |
| Use Case | Delete admin of command post |
| Actor(s) | Admin of natural disaster |
| Description | The admin will choose one of command post then system will display the detail of post and admin of natural disaster delete the admin of chosen co-post. |

*Table 3.7.2.24 High Level Use Case of Delete Admin of Command Post*

|  |  |
| --- | --- |
| Use Case | Add natural disaster |
| Actor(s) | Admin of province |
| Description | The admin of province adds natural disaster that occurred in particular province. After the submission, then the system will save the data to database as a new natural disaster in a particular province chosen. |

*Table 3.7.2.25 High Level Use Case of Add Natural Disaster*

|  |  |
| --- | --- |
| Use Case | Edit natural disaster |
| Actor(s) | Admin of province |
| Description | The admin of province chooses one of natural disaster and they are able to edit/ update name of disaster. |

*Table 3.7.2.26 High Level Use Case of Edit Natural Disaster*

|  |  |
| --- | --- |
| Use Case | Delete natural disaster |
| Actor(s) | Admin of province |
| Description | The admin of province deletes one of natural disaster in a province that has been fully recovered. |

*Table 3.7.2.27 High Level Use Case of Delete Natural Disaster*

|  |  |
| --- | --- |
| Use Case | View admin of natural disaster |
| Actor(s) | Admin of province |
| Description | The admin of province can view the list of admins of natural disaster for one natural disaster. |

*Table 3.7.2.28 High Level Use Case of View Admin of Natural Disaster*

|  |  |
| --- | --- |
| Use Case | Add admin of natural disaster |
| Actor(s) | Admin of province |
| Description | The admin of province adds admin of natural disaster, then system will save the data to database a new member of natural disaster admin. |

*Table 3.7.2.29 High Level Use Case of Add Admin of Natural Disaster*

|  |  |
| --- | --- |
| Use Case | Edit admin of natural disaster |
| Actor(s) | Admin of province |
| Description | The admin of province chooses one of natural disaster admin then after the system display the information about the natural disaster admin, they can edit the information as they need. |

*Table 3.7.2.30 High Level Use Case of Edit Admin of Natural Disaster*

|  |  |
| --- | --- |
| Use Case | Delete admin of natural disaster |
| Actor(s) | Admin of province |
| Description | The admin of province chooses one of natural disaster and delete particular admin in it. |

*Table 3.7.2.31 High Level of Use Case of Delete Admin of Natural Disaster*

|  |  |
| --- | --- |
| Use Case | Add admin of province |
| Actor(s) | Super admin |
| Description | Super admin fills the form of admin of province registration after the submission, then the system will save data to database as a new member of province admin. |

*Table 3.7.2.32 High Level Use Case of Add Admin of Province*

|  |  |
| --- | --- |
| Use Case | Edit admin of province |
| Actor(s) | Super admin |
| Description | Super admin chooses one of admin province, then system will display the information and super admin allowed to edit the information of chosen admin. |

*Table 3.7.2.33 High Level Use Case of Edit Admin of Province*

|  |  |
| --- | --- |
| Use Case | Delete admin of province |
| Actor(s) | Super admin |
| Description | Super admin chooses one of province admin and delete them. |

*Table 3.7.2.34 High Level Use Case of Delete Admin of Province*

|  |  |
| --- | --- |
| Use Case | Submit testimonials |
| Actor(s) | All admin and user |
| Description | All admin and user give their testimonial about natural disaster information system. |

*Table 3.7.2.35 High Level Use Case of Submit Testimonials*

|  |  |
| --- | --- |
| Use Case | Approve testimonials |
| Actor(s) | Super admin |
| Description | Super admin approves the particular testimonial about natural disaster information system. |

*Table 3.7.2.36 High Level Use Case of Approve Testimonials*

|  |  |
| --- | --- |
| Use Case | Payment Approval |
| Actor(s) | Super admin |
| Description | Super admin needs to check the payment whether it is successfully donated via bank account that is written in the web. Once the super admin confirmed the payment, then the name of donator will be show in the payment report. |

*Table 3.7.2.37 High Level Use Case of Payment Approval*

### Expanded Use Case

|  |  |
| --- | --- |
| **Use Case Name** | **Register** |
| **Goal in Content** | The users can register to be registered users in this web application |
| **Primary Actor Secondary Actor** | Users None |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins when users open register page. | 2. System displays the register page. |
| 3. The users complete the forms with the appropriate data | 4. The system should check each data. |
| 5. The users continue to fill up the forms until they are completed. |  |
| 6. The users click ‘Submit’ button. | 7. The data is saved to database. The system will save the database into appropriate table, give the identity (ID) number. |
|  | 8. The registered users will be directed to the user’s page. |
| **Alternatives course of Events** | |
| 4. If the data that is input is not matched with the requirements, there should be warning for the users to input the right data. | |

*Table 3.7.3.1 Expanded Use Case of Register*

|  |  |
| --- | --- |
| **Use Case Name** | **Log in User** |
| **Goal in Content** | The users can successfully log in to the system |
| **Primary Actor Secondary Actor** | Registered users None |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins when users open log in page. | 2. System displays log in page. |
| 3. The users fill up the form with their username and password | 4. The system will check whether the username or password are matched to database or not. |
| 5. The users will successfully log in to the system. |  |
| **Alternatives course of Events** | |
| 4. If the data is not matched with any username or password in database the system should give information to the users, asked them to check the username or password and retype them. | |

*Table 3.7.3.2 Expanded Use Case of Log In User*

|  |  |
| --- | --- |
| **Use Case Name** | **Log in Admin** |
| **Goal in Content** | The admin can successfully log in to the system |
| **Primary Actor Secondary Actor** | Admin None |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins when admin open log in page for admin. | 2. System displays log in page. |
| 3. The admins fill up the form with their username and password | 4. The system will check whether the username or password are matched to database or not. |
| 5. The admins will successfully log in to the system. |  |
| **Alternatives course of Events** | |

|  |
| --- |
|  |
| 5. If the data is not matched with any username or password in database the system should give information to the admin, asked them to check the username or password and retype them. |

*Table 3.7.3.3 Expanded Use Case of Log In Admin*

|  |  |
| --- | --- |
| **Use Case Name** | **Log out** |
| **Goal in Content** | The users and admins can successfully log out from the system. |
| **Primary Actor Secondary Actor** | Users and Admins None |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Users or admins click log out. | 2. System will redirect to home page. |
| **Alternatives course of Events** | |
|  | |

*Table 3.7.3.4 Expanded Use Case of Log Out*

|  |  |
| --- | --- |
| **Use Case Name** | **Edit Profile** |
| **Goal in Content** | The registered users can edit their profile. |
| **Primary Actor Secondary Actor** | Registered users None |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. The use case begins when the users click “Edit Profile” | 2. The system will show the forms to edit their profile. |
| 3. The users will edit the profile | 4. The system checks the data in the forms. |
| 5. The users click “Submit” button. | 6. The data changes are replaced the former data in database. |
| **Alternatives course of Events** | |
| 4. If the users did not input the suitable data, the system should warn it and ask to retype. | |

*Table 3.7.3.5 Expanded Use Case of Edit Profile*

|  |  |
| --- | --- |
| **Use Case Name** | **View List of Provinces** |
| **Goal in Content** | The registered users and admins can see the list of provinces |
| **Primary Actor Secondary Actor** | Admin and Users None |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins when users or admins click ‘Provinces’ in navigation bar. | 2. System will show the list of provinces. |
| 3. The users or admins choose one of the provinces in Indonesia. |  |
| **Alternatives course of Events** | |
|  | |

*Table 3.7.3.6 Expanded Use Case of View List of Provinces*

|  |  |
| --- | --- |
| **Use Case Name** | **View List of Natural Disaster** |
| **Goal in Content** | The registered users and admins can see the list of disaster |
| **Primary Actor Secondary Actor** | Admin and Users None |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begin when users or admins click ‘Provinces’ in navigation bar. | 2. System will show the list of provinces. |
| 3. The users or admins choose one of the provinces in Indonesia. | 4. The system will show the list of natural disaster in it. |
| 5. The users or admins can choose one of natural disaster. | 6. The system will show the list of command post in it. |
| **Alternatives course of Events** | |
|  | |

*Table 3.7.3.7 Expanded Use Case of View List of Natural Disaster*

|  |  |
| --- | --- |
| **Use Case Name** | **View List of Command Post** |
| **Goal in Content** | The registered users and admins can see the list of command post in particular disaster. |
| **Primary Actor Secondary Actor** | Admin and Users None |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begin when users or admins click ‘Provinces’ in navigation bar. | 2. System will show the list of provinces. |
| 3. The users or admins choose one of the provinces in Indonesia. | 4. The system will show the list of natural disaster in it. |
| 5. The users or admins can choose one of natural disaster. | 6. The system will show the list of command post in it. |
| **Alternatives course of Events** | |
|  | |

*Table 3.7.3.8 Expanded Use Case of View List of Command Post*

|  |  |
| --- | --- |
| **Use Case Name** | **Donate Money** |
| **Goal in Content** | The registered user can donate money through this web application. |
| **Primary Actor Secondary Actor** | Registered user Super Admin |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. The use case begins when donator click “Donation”. | 2. The system will show dropdown menu that gives choices to choose fund, event, goods or become a volunteer. |
| 3. Users choose “Fund” | 4. The system will inform the account bank number and the forms to be filled about the donation. |
| 5. Users filled up the form | 6. The system should check one by one the form, if the data that is input meet the requirements of system or not. |

|  |  |
| --- | --- |
| 7. Users click “Submit” button. | 8. The system will save it to database, and needs to be approved by the super admin and give information to wait for 2 days. |
| 9. Super admin checks donation that has been submit and approved it. | 10. The user’s name, donation amount and donation date will be appeared in the report. |
| **Alternatives course of Events** | |
| 6. The system should warn the donator if they fill the form incorrectly, and ask them to retype it.  9. Super admin can remove the donation submitted that it is not matched with the information in the bank’s account. | |

*Table 3.7.3.9 Expanded Use Case of Donate Money*

|  |  |
| --- | --- |
| **Use Case Name** | **Donate Event** |
| **Goal in Content** | The registered user can donate or hold an event in particular command post. |
| **Primary Actor Secondary Actor** | Registered users  Admin of command post |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. The use case begins when donator click “Donation”. | 2. The system will show dropdown menu that gives choices to choose fund, event, goods or become a volunteer. |
| 3. Users choose “Event” | 4. The system will give information or steps how to donate or hold an event in particular command post. |
| 5. Users go to find the command post that they wanted to donate and click for “View Details”. | 6. The system will show the details of command post and three buttons in the top of it. They are:” Get Direction”,” Donate Event”, and “Become a Volunteer”. |
| 7. Users click “Donate Event” | 8. The system will show the forms to be filled up. |

|  |  |
| --- | --- |
| 9. The users filled up all the forms and click “Submit”. | 10. The system will save it to database that should be approved or declined by the command post’s admin. |
| 11. Admin of command post will check the details of the event submitted and approved it. | 12. The event will be appeared in command post, after the admin approved it. |
| **Alternatives course of Events** | |
| 11. The admin can decline the event if there are two events in the same day or the event is not needed by the command post. | |

*Table 3.7.3.10 Expanded Use Case of Donate Event*

|  |  |
| --- | --- |
| **Use Case Name** | **Donate Logistical Aids** |
| **Goal in Content** | The registered user can get the direction to the command post and give the logistical aid by themselves. |
| **Primary Actor Secondary Actor** | Registered users None |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. The use case begins when donator click “Donation”. | 2. The system will show dropdown menu that gives choices to choose fund, event, goods or become a volunteer. |
| 3. Users choose “Logistical Aids” | 4. The system will give information or steps how to donate or hold an event in particular command post. |
| 5. Users go to find the command post that they wanted to donate and click for “View Details”. | 6. The system will show the details of command post and three buttons in the top of it. They are:” Get Direction”,” Donate Event”, and “Become a Volunteer”. |
| 7. Users click “Get Directions” | 8. The system will show the direction to the command post. |

|  |  |
| --- | --- |
| 9. The users clicks the command post position that is shown up in the system. | 10. The system will guide us to go to command post as we saw in google maps. |
| **Alternatives course of Events** | |
|  | |

*Table 3.7.3.11 Expanded Use Case of Donate Logistical Aids*

|  |  |
| --- | --- |
| **Use Case Name** | **Become a volunteer** |
| **Goal in Content** | The registered user can participate to register to be a volunteer in particular command post. |
| **Primary Actor Secondary Actor** | Registered users  Admin of command post |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. The use case begins when donator click “Donation”. | 2. The system will show dropdown menu that gives choices to choose fund, event, goods or become a volunteer. |
| 3. Users choose “Become a Volunteer” | 4. The system will give information or steps how to be a volunteer in particular command post. |
| 5. Users go to find the command post that they wanted to donate and click for “View Details”. | 6. The system will show the details of command post and three buttons in the top of it. They are:” Get Direction”,” Donate Event”, and “Become a Volunteer”. |
| 7. Users click “Become a Volunteer” | 8. The system will give the form for candidates of volunteer. |
| 9. The users fill up all the forms. | 10. The system will check the data one by one whether it meets the system requirement or not. |
| 11. The users click the “Submit” button. | 12. The system saves it to database, and give it to admin of command post to be approved. |

|  |  |
| --- | --- |
| 13. The admin of command post approves the volunteer. | 14. The name of the volunteer will be appeared in the command post by the system. |
| **Alternatives course of Events** | |
| 10. The system will give the warning to the users if they did not fill one of the field or they fill it wrong.  13. The admin can decline the volunteer if he/she does not meet the requirement. | |

*Table 3.7.3.12 Expanded Use Case of Become a Volunteer*

|  |  |
| --- | --- |
| **Use Case Name** | **Update Details of Command Post** |
| **Goal in Content** | Admin in command post can update information about command post |
| **Primary Actor Secondary Actor** | Admin in Command Post None |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. The use case begins when admin in click ‘Update Details’ | 2. The system will give the form for updating the details. |
| 3. Admin of command post change the data in the form. | 4. The system will check one by one the form. |
| 5. Admin continues to complete all the form and at the end click “Submit” | 6. The system will save them to database. |
|  | 7. The system will be directed to details of command post with the updated data. |
| **Alternatives course of Events** | |
| 4. The system should warn the admin of command post if the data which is input does not meet the requirements. | |

*Table 3.7.3.13 Expanded Use Case of Update Details of Command Post*

|  |  |
| --- | --- |
| **Use Case Name** | **Add Stock in Command Post** |
| **Goal in Content** | Admins of command post can add stock in their command post |

|  |  |
| --- | --- |
| **Primary Actor Secondary Actor** | Admins of Command Post None |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. The use case begins when admins click “Stock” | 2. The system will show the list of stock in that command post and the form to add stock. |
| 3. The admins fill the form to add the stock completely. | 4. The system will check one by one the data that is input with the form requirements. |
| 5. The admins click “add” | 6. The data will be saved to database and will be added to the list, and published to command post |
| **Alternatives course of Events** | |
| 3. The system should warn the admins if they input inappropriate data, and ask to retype them. | |

*Table 3.7.3.14 Expanded Use Case of Add Stock in Command Post*

|  |  |
| --- | --- |
| **Use Case Name** | **Add Needs in Command Post** |
| **Goal in Content** | Admins of command post can add needs in their command post |
| **Primary Actor Secondary Actor** | Admins of Command Post None |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. The use case begins when admins click “Needs” | 2. The system will show the list of needs in that command post and the form to add needs. |
| 3. The admins fill the form to add the needs completely. | 4. The system will check one by one the data that is input with the form requirements. |
| 5. The admins click “add” | 6. The data will be saved to database and will be added to the list, and published to command post |

|  |
| --- |
|  |
| **Alternatives course of Events** |
| 4. The system should warn the admins if they input inappropriate data, and ask to retype them. |

*Table 3.7.3.15 Expanded Use Case of Add Needs in Command Post*

|  |  |
| --- | --- |
| **Use Case Name** | **Update Stock in Command Post** |
| **Goal in Content** | Admins of command post can update stock in their command post |
| **Primary Actor Secondary Actor** | Admins of Command Post None |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. The use case begins when admins click “Stock” | 2. The system will show the list of stock in that command post and the form to add needs. |
| 3. The admins click the icon to update the data of stock in one of the stocks. | 4. The system will move the page for updating page. |
| 5. The admins update the information by filling up the form | 6. The data is checked one by one by the system. |
| 7. The admins then click ‘update’ | 8. The data will be saved to database and the data of the needs will automatically change the amount in the needs. |
| **Alternatives course of Events** | |
| 6. The system should warn the admins if they input inappropriate data, and ask to retype them. | |

*Table 3.7.3.16 Expanded Use Case of Update Stock in Command Post*

|  |  |
| --- | --- |
| **Use Case Name** | **Delete Stock in Command Post** |
| **Goal in Content** | Admins of command post can delete stock in command post. |
| **Primary Actor Secondary Actor** | Admins of Command Post None |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. The use case begins when admins click “Stock” | 2. The system will show the list of needs in that command post and the form to add stock. |
| 3. The admins click the icon to delete the stock in one of the record. | 4. The system will remove the record on the list and on database. |
| **Alternatives course of Events** | |
|  | |

*Table 3.7.3.17 Expanded Use Case of Delete Stock in Command Post*

|  |  |
| --- | --- |
| **Use Case Name** | **Update Needs in Command Post** |
| **Goal in Content** | Admins of command post can update needs in their command post |
| **Primary Actor Secondary Actor** | Admins of Command Post None |
| **Actor Action** | **System Response** |
| 1. The use case begins when admins click “Needs” | 2. The system will show the list of needs in that command post and the form to add needs. |
| 3. The admins click the icon to update the data of need in one of the needs | 4. The system will move the page for updating page. |
| 5. The admins update the information by filling up the form | 6. The data is checked one by one by the system. |
| 7. The admins then click ‘update’ | 8. The data will be saved to database and changed in the command post. |
| **Alternatives course of Events** | |
| 6. The system should warn the admins if they input inappropriate data, and ask to retype them. | |

Table 3.7.3.18 *Expanded Use Case of Update Needs in Command Post*

|  |  |
| --- | --- |
| **Use Case Name** | **Delete Needs in Command Post** |
| **Goal in Context** | Admin of command post can delete one of needs in a post |
| **Primary Actor Secondary Actor** | Admin of command post (None) |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Admin of command post delete the needs in a post | 2. System displays message of confirmation about the deletion of needs information. |
| 3. Admin of command post approves the deletion of needs information | 4. System saves the changes and displays the new modification of needs changes. |
| **Alternatives course of Events** | |
|  | |

*Table 3.7.3.19 Expanded Use Case of Delete Needs in Command Post*

|  |  |
| --- | --- |
| **Use Case Name** | **Event Approval** |
| **Goal in Context** | Admin of command post is able to approve the event which has been submitted by users. |
| **Primary Actor Secondary Actor** | Admin of command post (None) |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins when the admin of command post chooses “List of Events” in the menu. | 2. The system will show the admin the list of the event. |
| 3. The admin approves the event. | 4. The event will be appeared in command post details, so the users are able to see that. |
| **Alternatives course of Events** | |
| **3.** The admin can decline the event, if it’s not suitable for the refuges or command post. | |

*Table 3.7.3.20 Expanded Use Case of Event Approval*

|  |  |
| --- | --- |
| **Use Case Name** | **Volunteer Approval** |
| **Goal in Context** | Admin of command post is able to approve the volunteer which has request to participate in the command post. |
| **Primary Actor Secondary Actor** | Admin of command post (None) |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins when the admin of command post chooses “List of Volunteer” in the menu. | 2. The system will show the admin the list of the volunteer. |
| 3. The admin approves the volunteer. | 4. The volunteer will be appeared in command post details, so the users are able to see that. |
| **Alternatives course of Events** | |
| 4. The admin can decline the request from the volunteer if he/she does not meet the requirement or the volunteer has been full. | |

*Table 3.7.3.21 Expanded Use Case of Volunteer Approval*

|  |  |
| --- | --- |
| **Use Case Name** | **Add admin of command post** |
| **Goal in Context** | Admin of natural disaster successfully add new admin of command post |
| **Primary Actor Secondary Actor** | Admin of natural disaster (None) |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Admin of natural disaster fill the form of command post admin registration and submit it. | 2. System saves the data as new member of command post and display them in website. |
| **Alternatives course of Events** | |
| - | |

*Table 3.7.3.22 Expanded Use Case of Add Admin of Command Post*

|  |  |
| --- | --- |
| **Use Case Name** | **Edit admin of command post** |
| **Goal in Context** | Admin of natural disaster successfully edit the information of command post admin |
| **Primary Actor Secondary Actor** | Admin of natural disaster (None) |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Admin of natural disaster choose one of command post | 2. System displays the details of command post |
| 3. Admin edit the information of command post admin and submit it. | 4. System displays message of confirmation about information modification of co-post admin |
| 5. Admin of natural disaster approves the confirmation | 6. System saves the modification of co-post admin information and display the new information modification of co-post admin in website. |
| **Alternatives course of Events** | |
| 5. If admin of natural disaster choose to not approve the confirmation then the system will cancel the modification of co-post admin information | |

*Table 3.7.3.23 Expanded Use Case of Edit Admin of Command Post*

|  |  |
| --- | --- |
| **Use Case Name** | **Add natural disaster** |
| **Goal in Context** | Admin of province successfully adds natural disaster that occurred in a province |
| **Primary Actor Secondary Actor** | Admin of province (None) |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Admin of province add new natural disaster option in website and submit it. | 2. System saves the data added to database and display the new natural disaster in website |
| **Alternatives course of Events** | |
| - | |

*Table 3.7.3.24 Expanded Use Case of Add Natural Disaster*

|  |  |
| --- | --- |
| **Use Case Name** | **Edit natural disaster** |
| **Goal in Context** | Admin of province successfully edit the natural disaster that has been added before |
| **Primary Actor Secondary Actor** | Admin of province (None) |
|  | |
| **Actor Action** | **System Response** |
| 1. Admin of province choose one of natural disaster | 2. System displays the field of natural disaster |
| 3. Admin edit the natural disaster’s name | 4. System displays message of confirmation about natural disaster’s modification |
| 5. Admin approves the confirmation | 6. System saves the modification confirmation and display the new changes made of natural disaster. |
| **Alternatives course of Events** | |
| 5. If admin of province chooses to not approve the confirmation then the system will cancel the modification of natural disaster. | |

*Table 3.7.3.25 Expanded Use Case of Edit Natural Disaster*

|  |  |
| --- | --- |
| **Use Case Name** | **Delete natural disaster** |
| **Goal in Context** | Admin of province successfully delete natural disaster that has been fully recovered |
| **Primary Actor Secondary Actor** | Admin of province (None) |
| **Actor Action** | **System Response** |
| 1. Admin of province choose one of natural disaster and delete it. | 2. System displays message of confirmation about deletion of natural disaster |
| 3. Admin approves the confirmation | 4. System save the deletion confirmation and delete the natural disaster with its co-post. |
|  | 5. System display the new changes made in the list of natural disaster in website. |
| **Alternatives course of Events** | |
| 3. If admin of province chooses to not approve the confirmation then the system will cancel the deletion of natural disaster. | |

*Table 3.7.3.26 Expanded Use Case of Delete Natural Disaster*

|  |  |
| --- | --- |
| **Use Case Name** | **Add admin of natural disaster** |
| **Goal in Context** | Admin of province successfully add new admin of command post |
| **Primary Actor Secondary Actor** | Admin of province (None) |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins when the admin of province chooses to click “List of Disaster admin” | 2. The system will show the list of admins of disaster and the forms to add admin of disaster. |
| 3. Admin of province fill the form of natural disaster admin registration and submit it. | 4. System displays message of confirmation about natural disaster admin addition |
| 5. Admin approves the confirmation | 6. System saves the data as new member of natural disaster admin and display them in the website. |
| **Alternatives course of Events** | |
| 3. If admin of province chooses to not approve the confirmation then the system will cancel to add new member of natural disaster admin. | |

*Table 3.7.3.27 Expanded Use Case of Add Admin of Natural Disaster*

|  |  |
| --- | --- |
| **Use Case Name** | **Delete admin of command post** |
| **Goal in Context** | Admin of natural disaster successfully delete co-post admin |
| **Primary Actor Secondary Actor** | Admin of natural disaster (None) |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Admin of natural disaster choose one of command post | 2. System displays the details of command post |
| 3. Admin delete the co-post admin | 4. System displays message of confirmation about deletion of co-post admin |

|  |  |
| --- | --- |
| 5. Admin of natural disaster approves the confirmation | 6. System saves the deletion confirmation and display the new changes made of co-post admin in website |
| **Alternatives course of Events** | |
| 5. If admin of natural disaster chooses to not approve the confirmation then the system will cancel the deletion of natural disaster admin. | |

*Table 3.7.3.28 Expanded Use Case of Delete Admin of Command Post*

|  |  |
| --- | --- |
| **Use Case Name** | **View Admin of Natural Disaster** |
| **Goal in Context** | Admin of province could see the list of admins of disaster. |
| **Primary Actor Secondary Actor** | Admin of province (None) |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins when the admin of province chooses to click “List of Natural Disaster” | 2. The system will show the list of disaster and the forms to add disaster. |
| 3. The admin clicks “View Admin” in one of the disaster. | 4. System shows the list of admin sin that disaster. |
| **Alternatives course of Events**  **-** | |

*Table 3.7.3.29 Expanded Use Case of View Admin of Natural Disaster*

|  |  |
| --- | --- |
| **Use Case Name** | **Edit admin of natural disaster** |
| **Goal in Context** | Admin of province successfully edit the information of natural disaster admin |
| **Primary Actor Secondary Actor** | Admin of province (None) |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins when admin of province choose one of natural disaster | 2. System displays the details of natural disaster |

|  |  |
| --- | --- |
| 3. Admin edit the information of natural disaster admin and submit it. | 4. System displays message of confirmation about information modification of natural disaster admin |
| 5. Admin approves the confirmation | 6. System saves the modification of natural disaster admin information and display the new information modification in website. |
| **Alternatives course of Events** | |
| 5. If admin of natural disaster choose to not approve the confirmation then the system will cancel the modification of co-post admin information | |

*Table 3.7.3.30 Expanded Use Case of Edit Admin of Natural Disaster*

|  |  |
| --- | --- |
| **Use Case Name** | **Delete admin of natural disaster** |
| **Goal in Context** | Admin of province successfully delete natural disaster admin |
| **Primary Actor Secondary Actor** | Admin of province (None) |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Admin of province choose one of natural disaster | 2. System displays the details of natural disaster chosen |
| 3. Admin delete the natural disaster admin | 4. System displays message of confirmation about deletion of natural disaster admin |
| 5. Admin approves the confirmation | 6. System saves the deletion confirmation and display the new changes made of natural disaster admin in website. |
| **Alternatives course of Events** | |
| 5. If admin of province choose to not approve the confirmation then the system will cancel the deletion of natural disaster admin. | |

*Table 3.7.3.31 Expanded Use Case of Delete Admin of Natural Disaster*

|  |  |
| --- | --- |
| **Use Case Name** | **Add admin of province** |
| **Goal in Context** | Super admin successfully add new admin of province |

|  |  |
| --- | --- |
| **Primary Actor Secondary Actor** | Super admin (None) |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Super admin fill the form of province admin registration and submit it. | 2. System saves the data as new member of province admin and display them in the website. |
| **Alternatives course of Events** | |
|  | |

*Table 3.7.3.32 Expanded Use Case of Add Admin of Province*

|  |  |
| --- | --- |
| **Use Case Name** | **Edit admin of province** |
| **Goal in Context** | Super admin successfully edit the information of province admin |
| **Primary Actor Secondary Actor** | Super admin (None) |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Super admin choose one of province | 2. System displays the details of province admin |
| 3. Super admin edit the information of province admin and submit it. | 4. System displays message of confirmation about information modification of province admin |
| 5. Super admin approves the confirmation | 6. System saves the modification of province admin information and display the new information modification in website. |
| **Alternatives course of Events** | |
| 5. If super admin choose to not approve the confirmation then the system will cancel the modification of province admin information | |

*Table 3.7.3.33 Expanded Use Case of Edit Admin of Province*

|  |  |
| --- | --- |
| **Use Case Name** | **Delete admin of province** |
| **Goal in Context** | Super admin successfully delete natural disaster admin |

|  |  |
| --- | --- |
| **Primary Actor Secondary Actor** | Super admin (None) |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Super admin choose one of province | 2. System displays the details of province admin in particular province chosen. |
| 3. Super admin delete the province admin | 4. System displays message of confirmation about deletion of province admin |
| 5. Super admin approves the confirmation | 6. System saves the deletion confirmation and display the new changes made of province admin in website. |
| **Alternatives course of Events** | |
| 5. If super admin chooses to not approve the confirmation then the system will cancel the deletion of province admin. | |

*Table 3.7.3.34 Expanded Use Case of Delete Admin of Province*

|  |  |
| --- | --- |
| **Use Case Name** | **Submit testimonial** |
| **Goal in Context** | All admin and user successfully submit their testimonial in website |
| **Primary Actor Secondary Actor** | All admin and user (None) |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Admin or user submit the testimonial in the field of testimonial in website | 2. System save the testimonial to database and only super admin who could view and approve it to be released in website. |
| **Alternatives course of Events** | |
| - | |

*Table 3.7.3.35 Expanded Use Case of Submit Testimonial*

|  |  |
| --- | --- |
| **Use Case Name** | **Approve testimonial** |
| **Goal in Context** | Super admin successfully approves the testimonial and release them in the website |

|  |  |
| --- | --- |
| **Primary Actor Secondary Actor** | Super admin (None) |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins when super admin chooses testimonial option | 2. System display the list of testimonials |
| 3. Super admin approves the testimonial | 4. System displays the testimonial to website |
| **Alternatives course of Events** | |
| 4. If super admin chooses to not approve the testimonials then the system will cancel to release the testimonial to website. | |

*Table 3.7.3.36 Expanded Use Case of Approve Testimonial*

|  |  |
| --- | --- |
| **Use Case Name** | **Payment Approval** |
| **Goal in Context** | Super admin successfully approves the payment approval from the users |
|  |  |
| **Primary Actor Secondary Actor** | Super admin (None) |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins when super admin chooses list of payment. | 2. System display the list of payment. |
| 3. Super admin checks the payment in their bank account and if it’s matched with the information submitted, the admin will approve it. | 4. System display the donation in donation report. |
| **Alternatives course of Events** | |
| 3. If super admin does not find that the information submitted does not match with information in bank account, super admin declines it. | |

*Table 3.7.3.37 Expanded Use Case of Payment Approval*

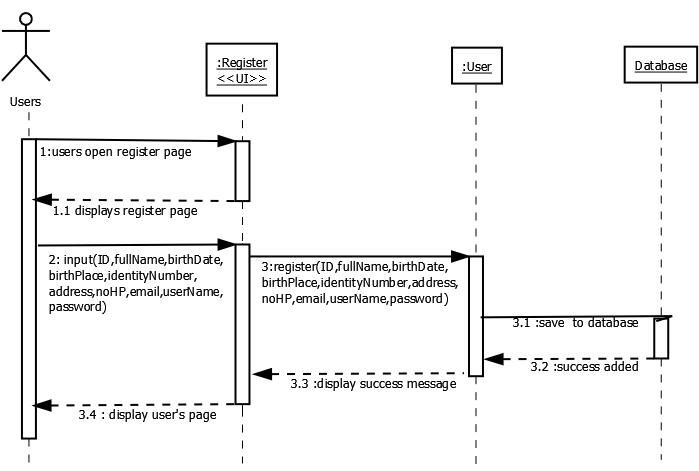
### Analysis Class Diagram



*Figure 3.7.4.1 Analysis Class Diagram*

### Design Artifacts

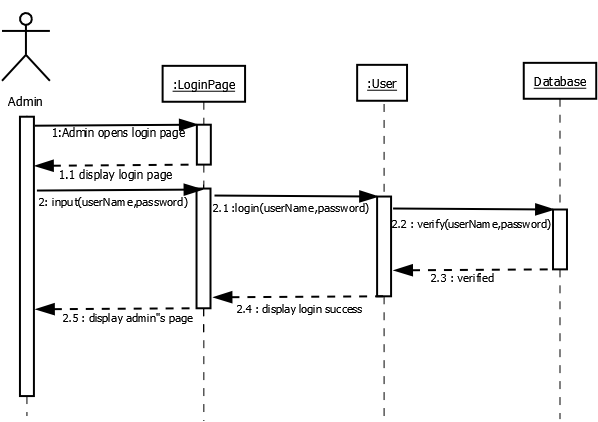
* + 1. **Sequence Diagram**



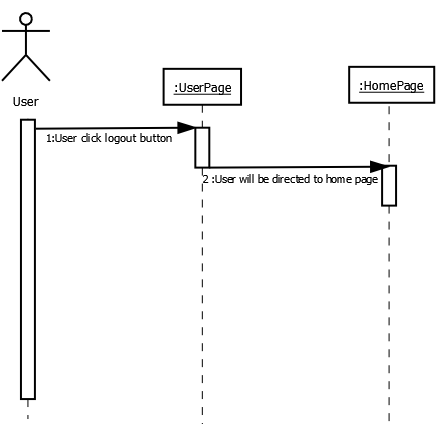
*Figure 3.8.1.1 Register Sequence Diagram*



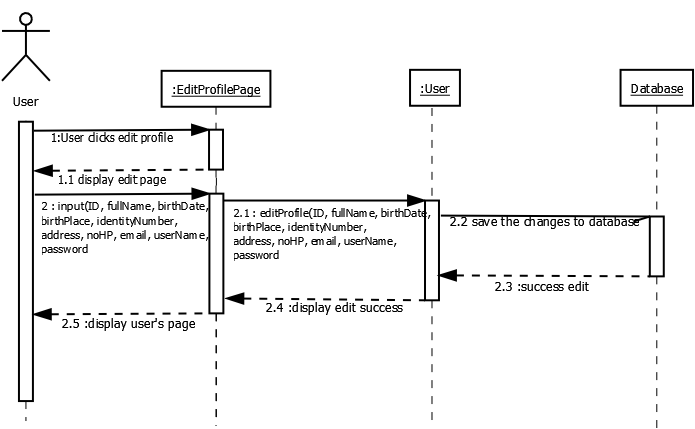
*Figure 3.8.1.2 Login User Sequence Diagram*



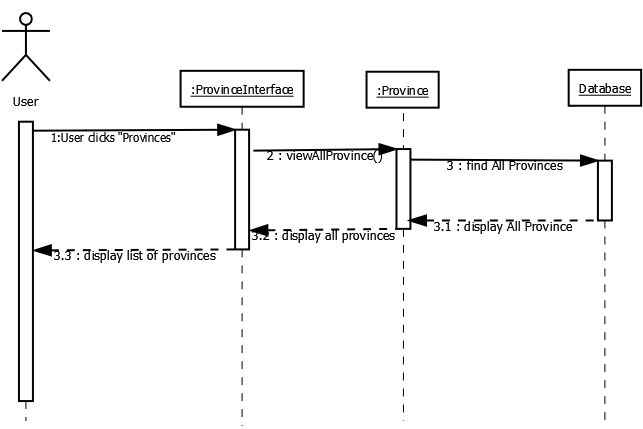
*Figure 3.8.1.3 Login Admin Sequence Diagram*



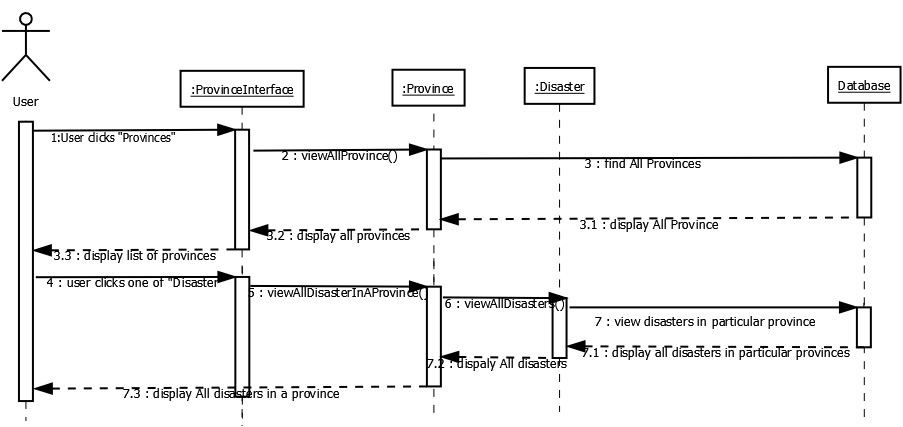
*Figure 3.8.1.4 Log Out Sequence Diagram*



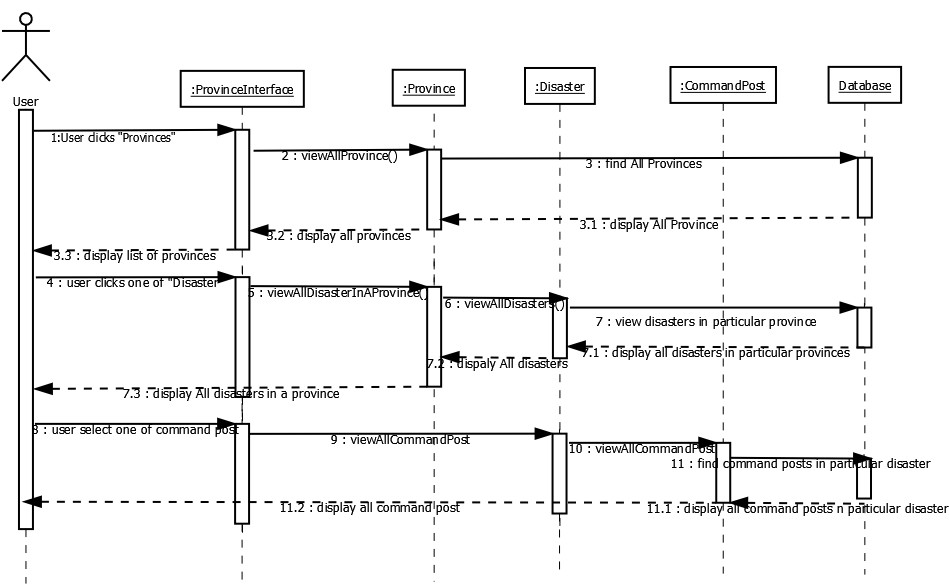
*Figure 3.8.1.5 Edit Profile Sequence Diagram*



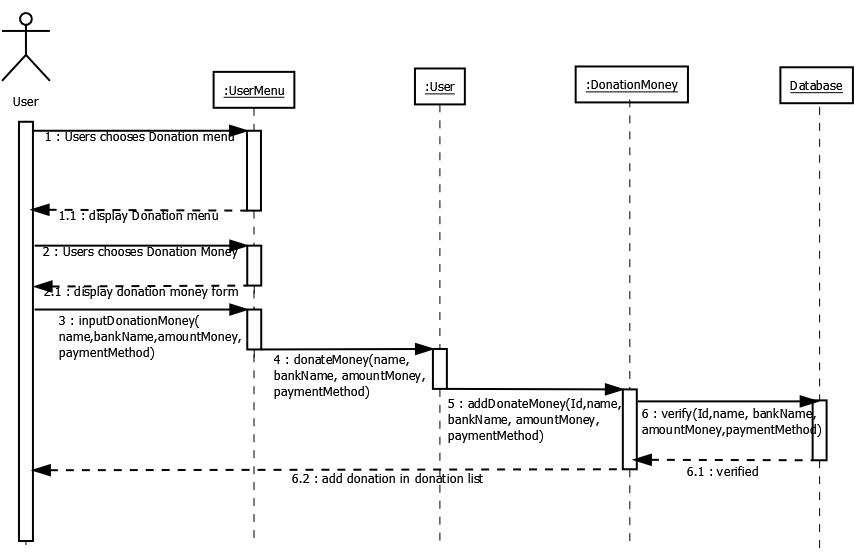
*Figure 3.8.1.6 View List of Province Sequence Diagram*



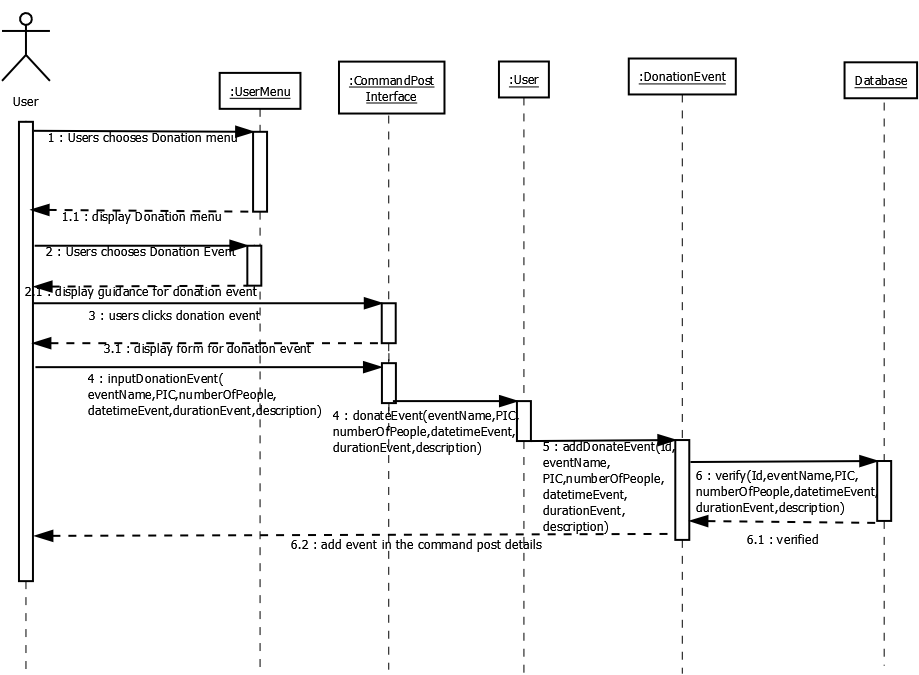
*Figure 3.8.1.7 View List of Natural Disaster Sequence Diagram*



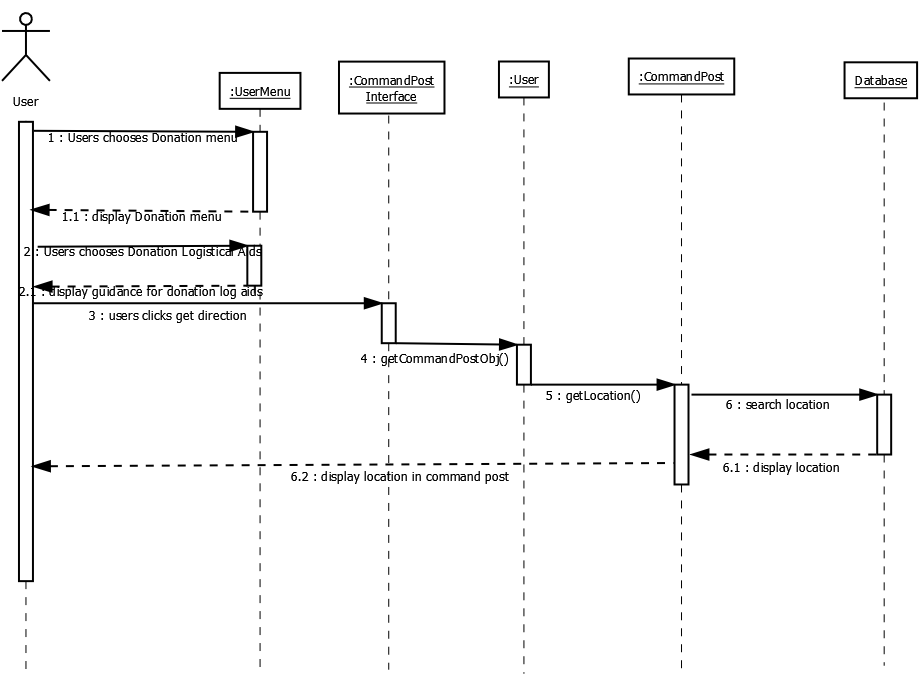
*Figure 3.8.1.8 View List of Command Post Sequence Diagram*



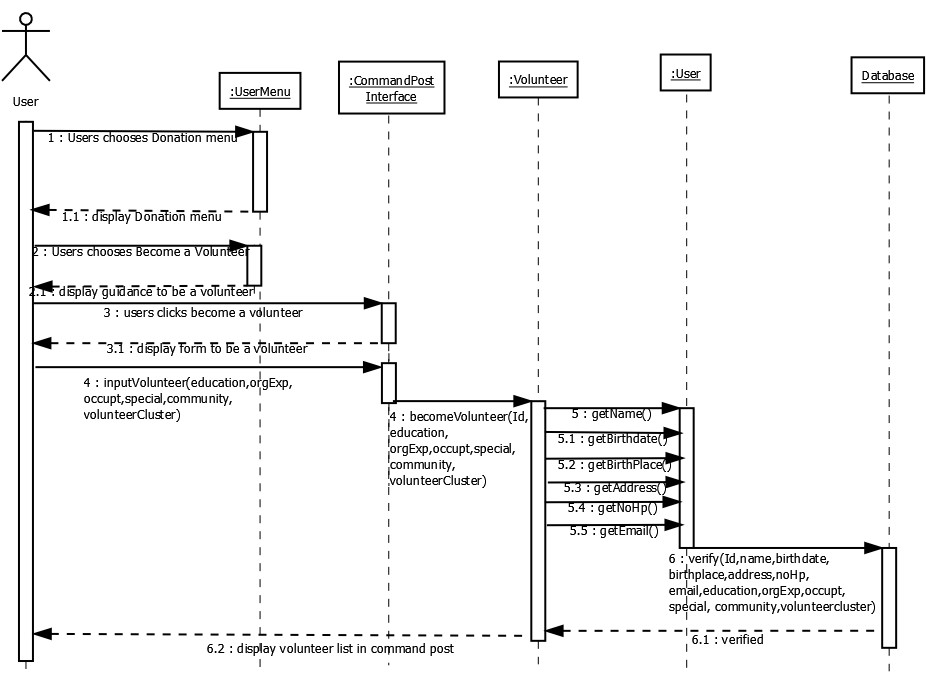
*Figure 3.8.1.9 Donate Money Sequence Diagram*



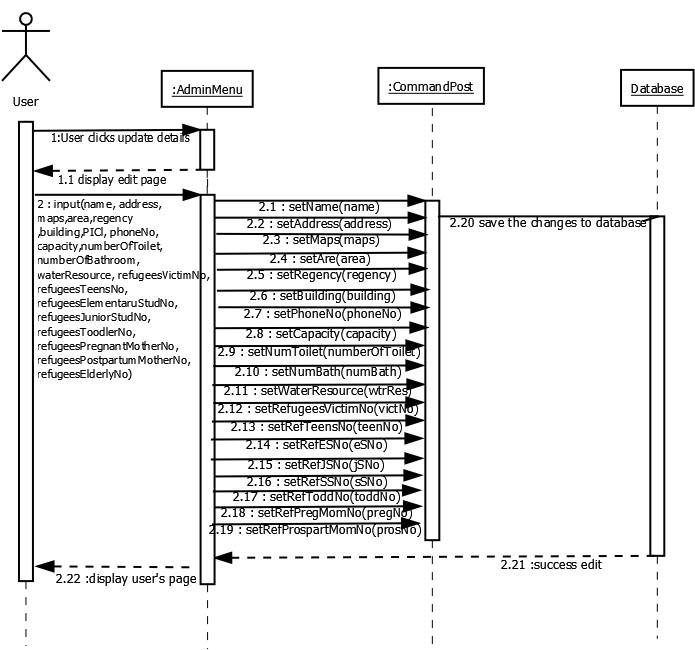
*Figure 3.8.1.10 Donate Event Sequence Diagram*



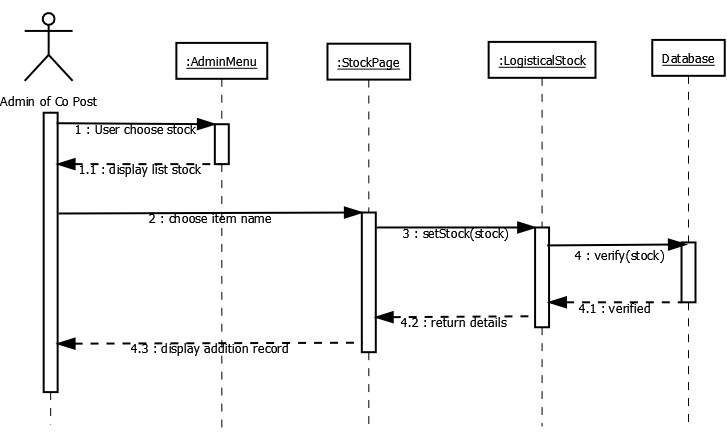
*Figure 3.8.1.11 Donate Logistical Aid Sequence Diagram*



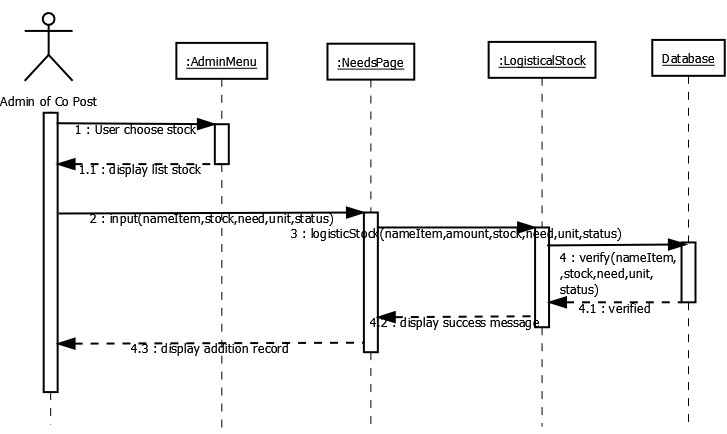
*Figure 3.8.1.12 Become a Volunteer Sequence Diagram*



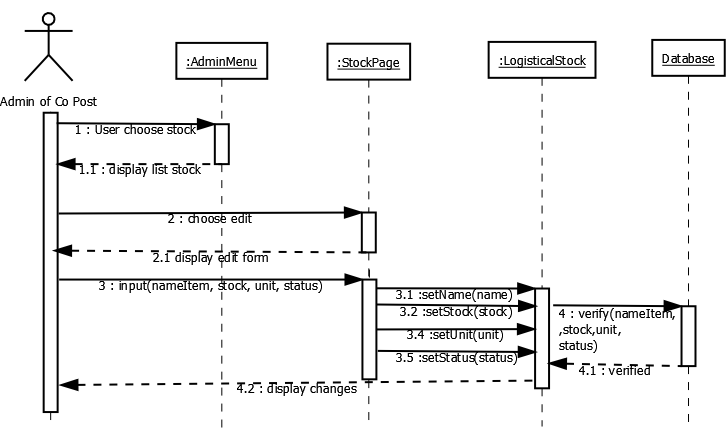
*Figure 3.8.1.13 Update Details of Command Post Sequence Diagram*



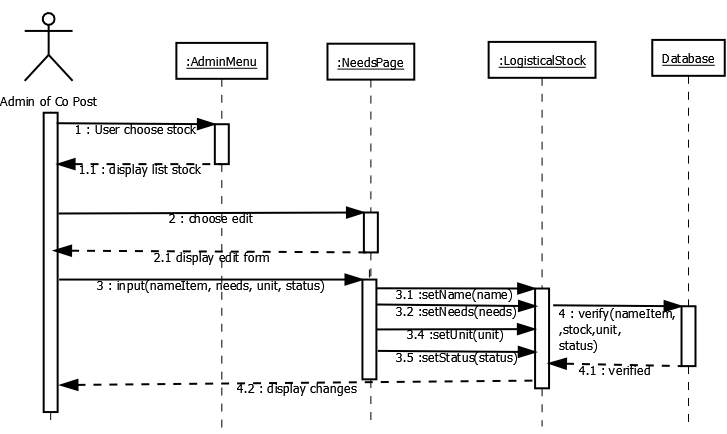
*3.8.1.14 Add Stock in Command Post Sequence Diagram*



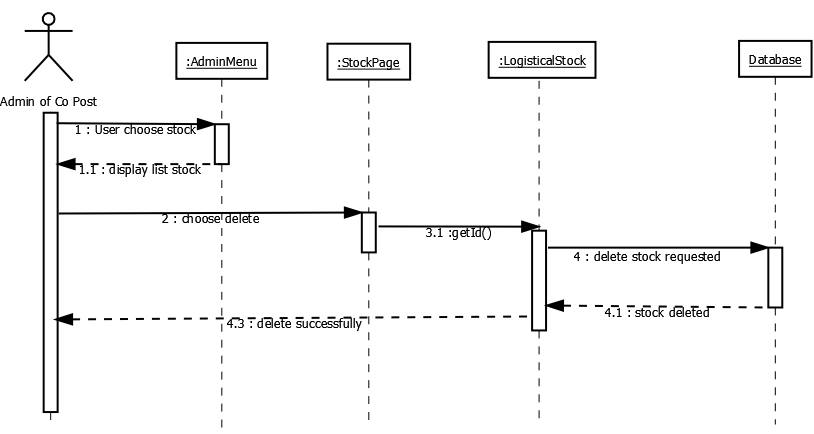
*Figure 3.8.1.15 Add Needs in Command Post Sequence Diagram*



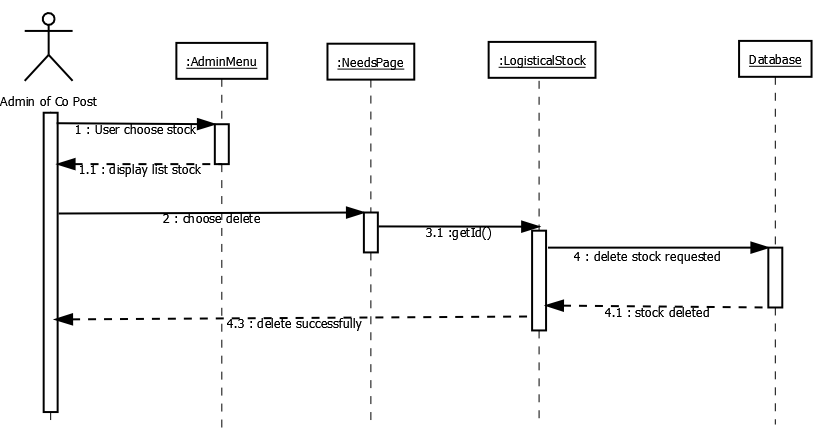
*Figure 3.8.1.16 Update Stock in Command Post Sequence Diagram*



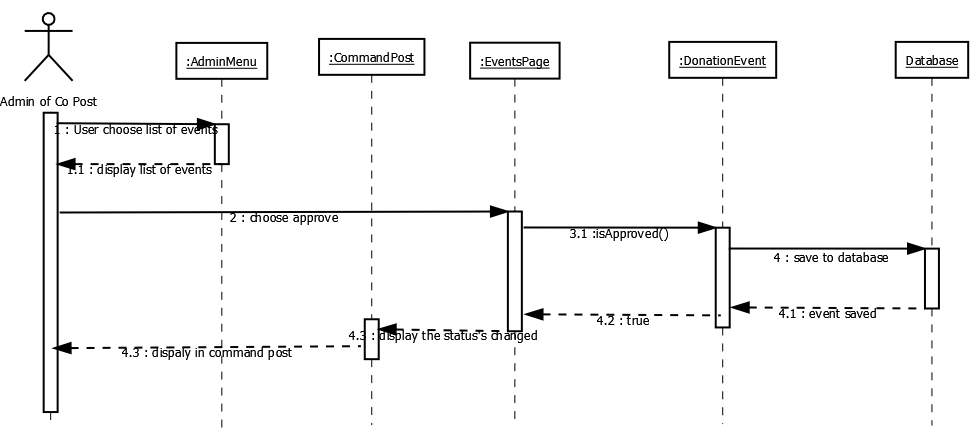
*Figure 3.8.1.17 Update Needs in Command Post Sequence Diagram*



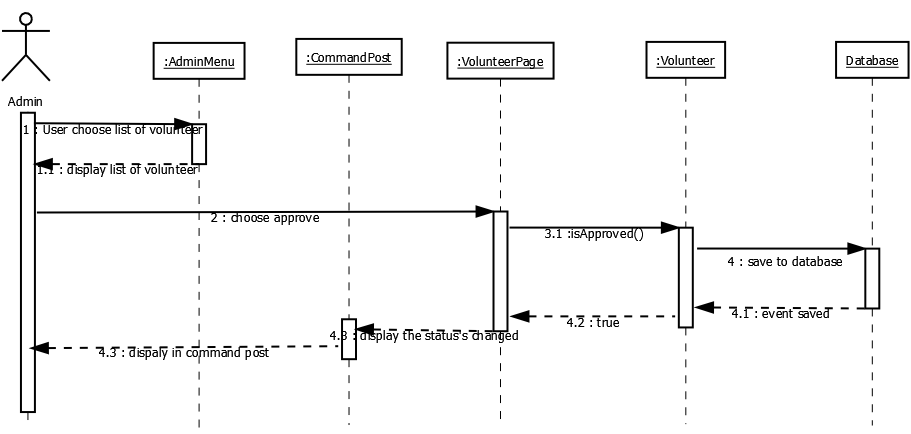
*Figure 3.8.1.18 Delete Stock in Command Post Sequence Diagram*



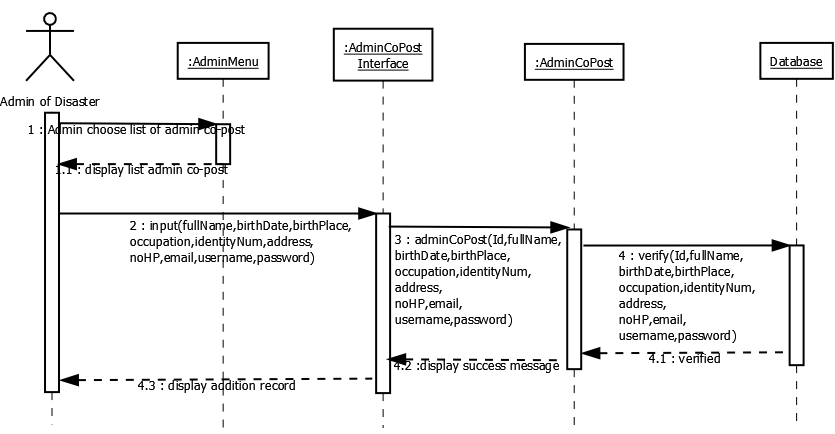
*Figure 3.8.1.19 Delete Needs in Command Post Sequence Diagram*



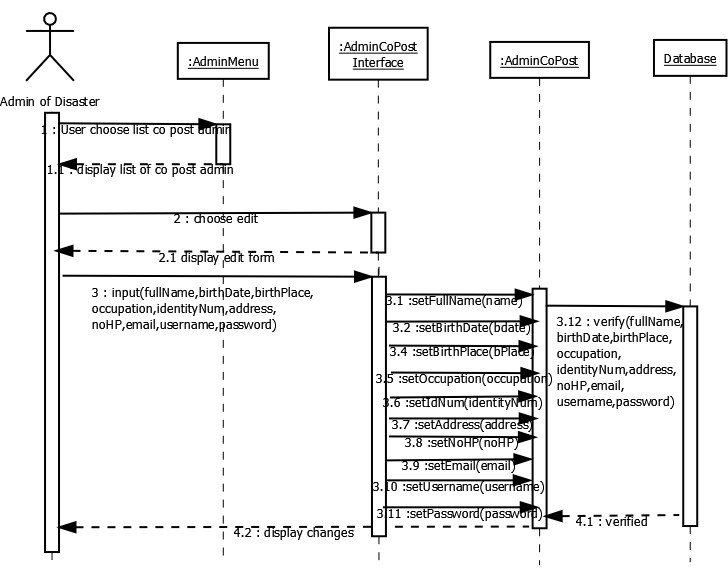
*Figure 3.8.1.20 Event Approval Sequence Diagram*



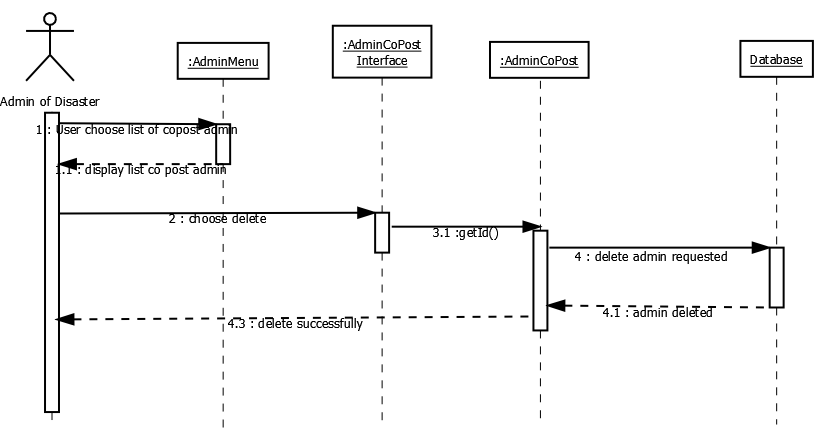
*Figure 3.8.1.21 Volunteer Approval Sequence Diagram*



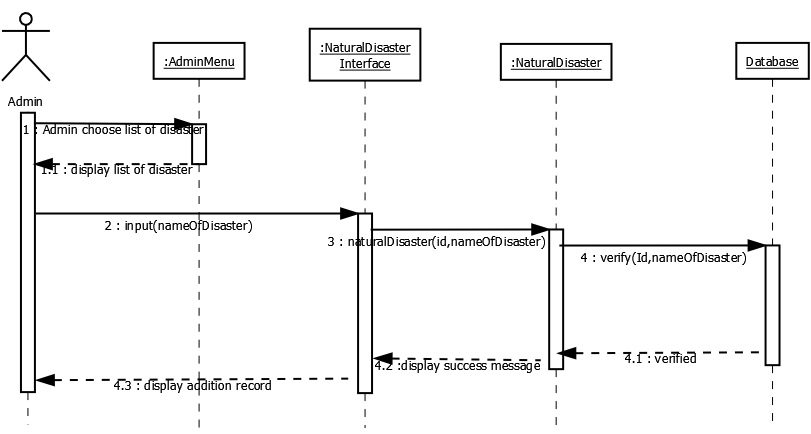
*Figure 3.8.1.22 Add Admin Command Post Sequence Diagram*



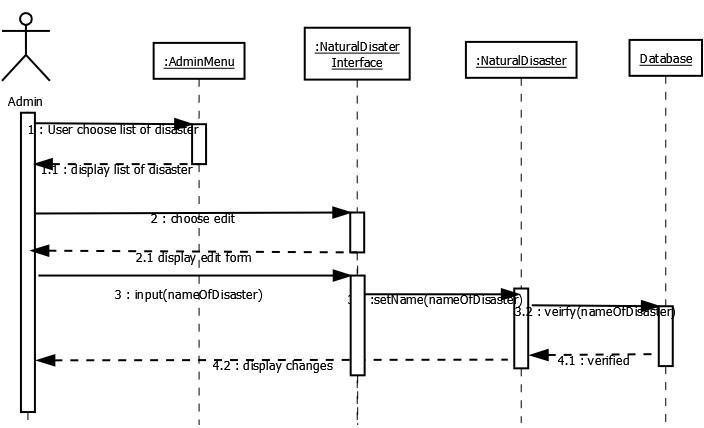
*Figure 3.8.1.23 Edit Admin Command Post Sequence Diagram*



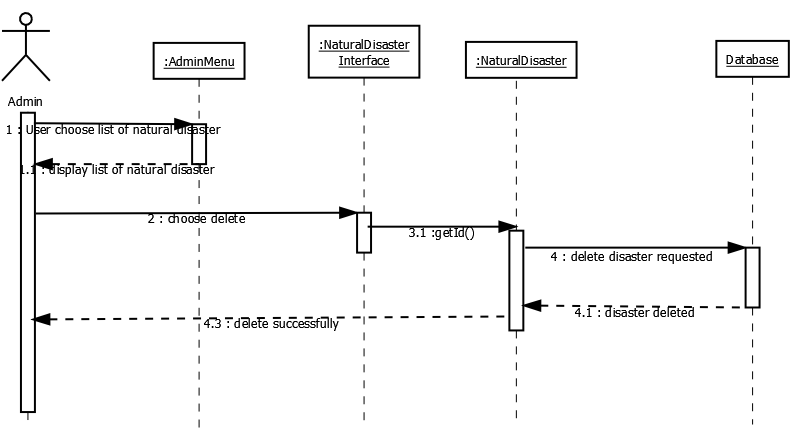
*Figure 3.8.1.24 Delete Admin Command Post Sequence Diagram*



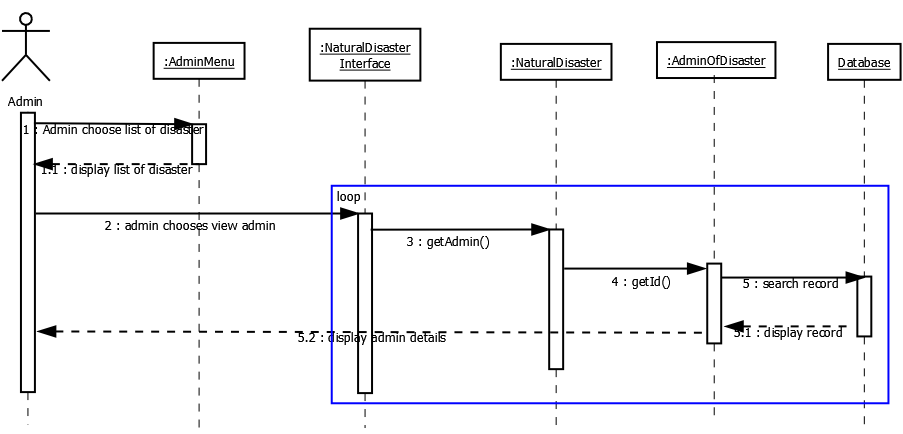
*Figure 3.8.1.25 Add Natural Disaster Sequence Diagram*



*Figure 3.8.1.26 Edit Natural Disaster Sequence Diagram*



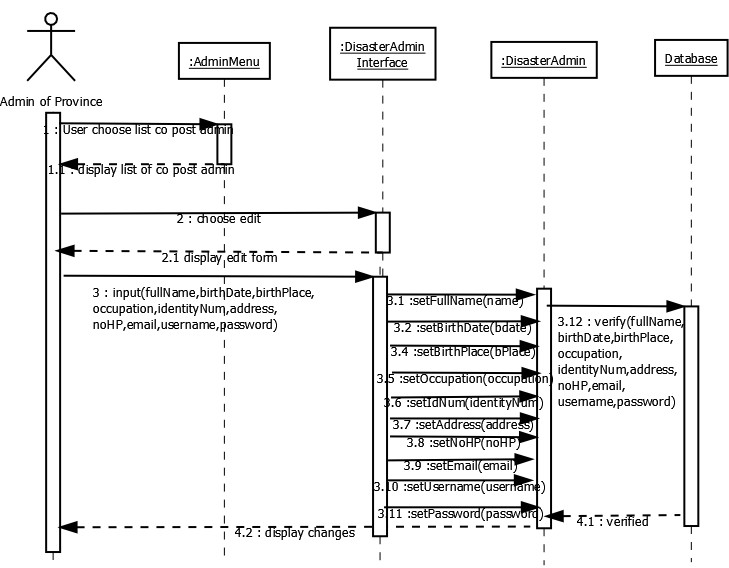
*Figure 3.8.1.27 Delete Natural Disaster Sequence Diagram*



*Figure 3.8.1.28 View Admin of Natural Disaster Sequence Diagram*



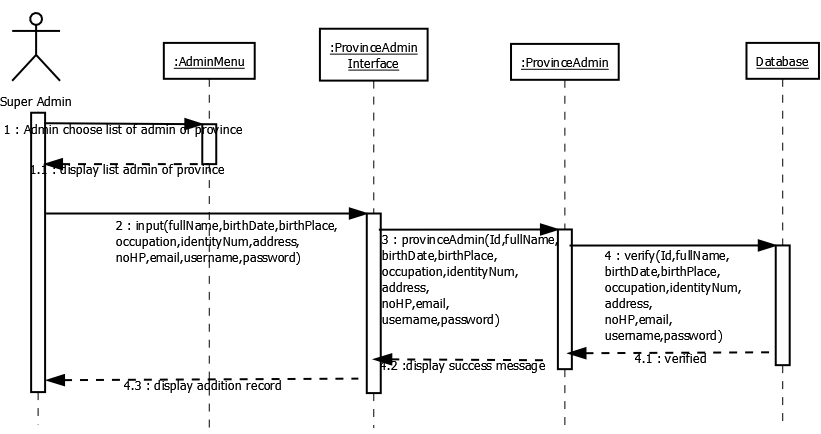
*Figure 3.8.1.29 Add Admin of Natural Disaster Sequence Diagram*



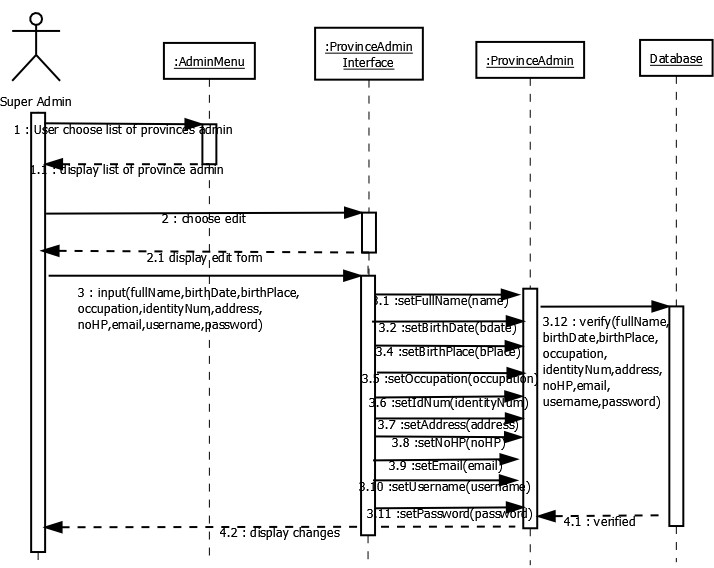
*Figure 3.8.1.30 Edit Admin of Natural Disaster Sequence Diagram*



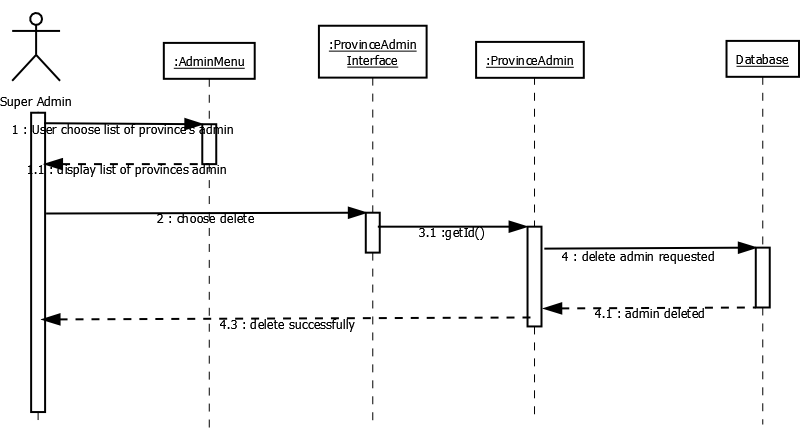
*Figure 3.8.1.31 Delete Admin of Natural Disaster Sequence Diagram*



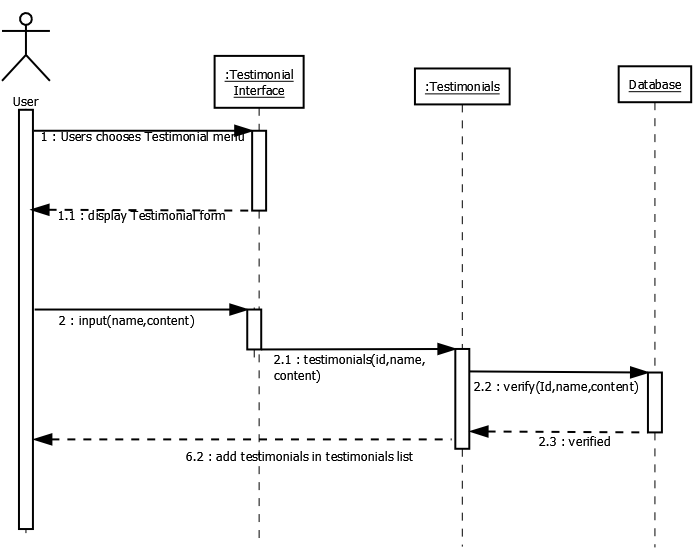
*Figure 3.8.1.32 Add Admin of Province Sequence Diagram*



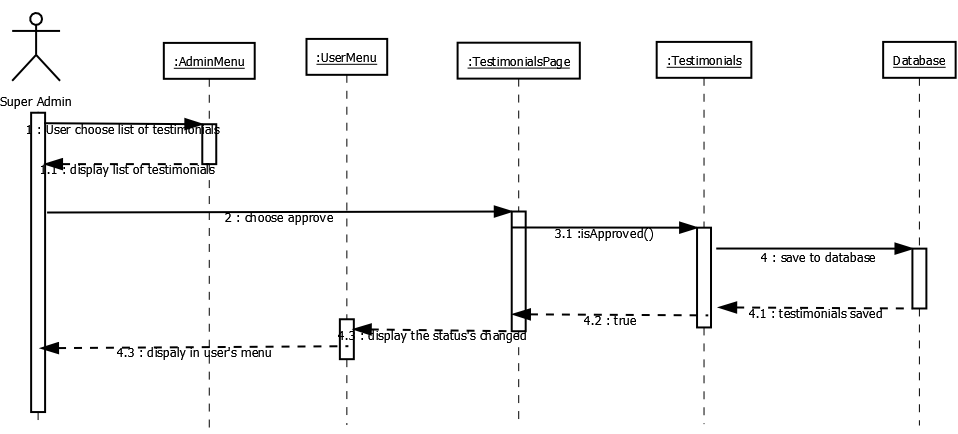
*Figure 3.8.1.33 Edit Admin of Province Sequence Diagram*



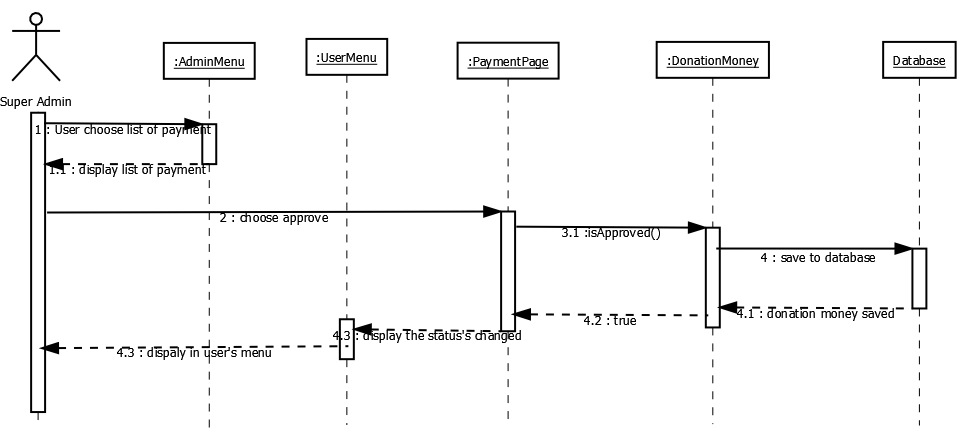
*Figure 3.8.1.34 Delete Admin of Province Sequence Diagram*



*Figure 3.8.1.35 Submit Testimonials Sequence Diagram*



*Figure 3.8.1.36 Approve Testimonials Sequence Diagram*



*Figure 3.8.1.37 Payment Approval Sequence Diagram*

### Database Design

**Super Admin**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Column Name** | **Data Type** | **Size** | **Allow Null** | **Default** |
| PK | SuperAdmin\_Id | Varchar | 5 | No |  |
|  | Name | Varchar | 30 | No |  |
|  | Birthdate | Date | 6 | No |  |
|  | Birthplace | Text | 20 | No |  |
|  | Address | Varchar | 100 | No |  |
|  | IdentityNumber | Varchar | 20 | No |  |
|  | Nohp | Varchar | 15 | No |  |
|  | Email | Varchar | 50 | No |  |
|  | Username | Varchar | 30 | No |  |
|  | Password | Varchar | 20 | No |  |
| FK | adminProvince\_Id | Varchar | 5 | No |  |
| FK | Testimonial\_Id | Varchar | 5 | No |  |
| FK | donationMoney\_Id | Varchar | 5 | No |  |
| FK | Province\_Id | Varchar | 5 | No |  |
| FK | Naturaldisaster\_Id | Varchar | 5 | No |  |
| FK | commandPost\_Id | Varchar | 5 | No |  |

*Table 3.8.2.1 Database Design Super Admin*

### User

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Column Name** | **Data Type** | **Size** | **Allow Null** | **Default** |
| PK | User\_Id | Varchar | 5 | No |  |
|  | Fullname | Varchar | 30 | No |  |
|  | Birthdate | Date | 6 | No |  |
|  | Birthplace | Text | 20 | No |  |
|  | Address | Varchar | 100 | No |  |
|  | IdentityNumber | Varchar | 20 | No |  |
|  | Nohp | Varchar | 15 | No |  |
|  | Email | Varchar | 50 | No |  |
|  | Username | Varchar | 30 | No |  |
|  | Password | Varchar | 20 | No |  |
| FK | commandPost\_Id | Varchar | 5 | No |  |
| FK | Province\_Id | Varchar | 5 | No |  |
| FK | Naturaldisaster\_Id | Varchar | 5 | No |  |
| FK | donationEvent\_Id | Varchar | 5 | No |  |
| FK | donationMoney\_Id | Varchar | 5 | No |  |
| FK | Testimonial\_Id | Varchar | 5 | No |  |

*Table 3.8.2.2 Database Design User*

### Volunteer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Column Name** | **Data Type** | **Size** | **Allow Null** | **Default** |
| PK | Volunteer\_Id | Varchar | 5 | No |  |
|  | Education | Text | 30 | No |  |
|  | organisationExperince | Text | 30 | No |  |
|  | Occupation | Text | 30 | No |  |
|  | Specialist | Text | 30 | No |  |
|  | Community | Text | 30 | No |  |
|  | VolunteerCluster | Text | 30 | No |  |

*Table 3.8.2.3 Database Design Volunteer*

### Donation Event

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Column Name** | **Data Type** | **Size** | **Allow Null** | **Default** |
| PK | donationEvent\_Id | Varchar | 5 | No |  |
|  | eventName | Text | 30 | No |  |
|  | PIC | Text | 30 | No |  |
|  | numberofPeople | Text | 30 | No |  |
|  | dateTimeEvent | Text | 30 | No |  |
|  | durationEvent | Text | 30 | No |  |
|  | Description | Text | 5000 | No |  |

*Table 3.8.2.4 Database Design Donation Event*

### Donation Money

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Column Name** | **Data Type** | **Size** | **Allow Null** | **Default** |
| PK | donationMoney\_Id | Varchar | 5 | No |  |
|  | Name | Varchar | 30 | No |  |
|  | bankName | Text | 20 | No |  |
|  | amountMoney | Integer | 30 | No |  |
|  | PaymentMethod | Text | 30 | No |  |

*Table 3.8.2.5 Database Design Donation Money*

### Testimonial

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Column Name** | **Data Type** | **Size** | **Allow Null** | **Default** |
| PK | Testimonial\_Id | Varchar | 5 | No |  |
|  | Name | Varchar | 30 | No |  |
|  | Comment | Text | 5000 | No |  |

*Table 3.8.2.6 Database Design Testimonial*

### Admin of Province

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Column Name** | **Data Type** | **Size** | **Allow Null** | **Default** |
| PK | AdminProvince\_Id | Varchar | 5 | No |  |
|  | Fullname | Varchar | 30 | No |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Birthdate | Date | 6 | No |  |
|  | Birthplace | Text | 20 | No |  |
|  | Occupation | Text | 20 | No |  |
|  | Address | Varchar | 100 | No |  |
|  | IdentityNumber | Varchar | 20 | No |  |
|  | Nohp | Varchar | 15 | No |  |
|  | Email | Varchar | 50 | No |  |
|  | Username | Varchar | 30 | No |  |
|  | Password | Varchar | 20 | No |  |
| FK | NaturalDisaster\_Id | Varchar | 5 | No |  |
| FK | Province\_Id | Varchar | 5 | No |  |
| FK | CommandPost\_Id | Varchar | 5 | No |  |
| FK | AdminNaturalDisaster\_Id | Varchar | 5 | No |  |
| FK | Testimonial\_Id | Varchar | 5 | No |  |

*Table 3.8.2.7 Database Design Admin of Province*

### Province

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Column Name** | **Data Type** | **Size** | **Allow Null** | **Default** |
| PK | Province\_Id | Varchar | 5 | No |  |
| FK | AdminProvince\_Id | Varchar | 5 | No |  |
|  | nameProvince | Text | 20 | No |  |
| FK | Naturaldisaster\_Id | Varchar | 5 | No |  |
| FK | AdminNaturalDisaster\_Id | Varchar | 5 | No |  |

*Table 3.8.2.8 Database Design Province*

### Admin of Natural Disaster

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Column Name** | **Data Type** | **Size** | **Allow Null** | **Default** |
| PK | AdminNaturalDisaster\_Id | Varchar | 5 | No |  |
|  | Fullname | Varchar | 30 | No |  |
|  | Birthdate | Date | 6 | No |  |
|  | Birthplace | Text | 20 | No |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Occupation | Text | 20 | No |  |
|  | Address | Varchar | 100 | No |  |
|  | IdentityNumber | Varchar | 20 | No |  |
|  | Nohp | Varchar | 15 | No |  |
|  | Email | Varchar | 50 | No |  |
|  | Username | Varchar | 30 | No |  |
|  | Password | Varchar | 20 | No |  |
| FK | NaturalDisaster\_Id | Varchar | 5 | No |  |
| FK | CommandPost\_Id | Varchar | 5 | No |  |
| FK | AdminCommandPost\_Id | Varchar | 5 | No |  |
| FK | Testimonial\_Id | Varchar | 5 | No |  |

*Table 3.8.2.9 Database Design Admin of Natural Disaster*

### Natural Disaster

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Column Name** | **Data Type** | **Size** | **Allow Null** | **Default** |
| PK | NaturalDisaster\_Id | Varchar | 5 | No |  |
|  | nameNaturalDisaster | Text | 50 | No |  |
| FK | AdminNaturalDisaster\_Id | Varchar | 5 | No |  |
| FK | commandPost\_Id | Varchar | 5 | No |  |

*Table 3.8.2.10 Database Design Natural Disaster*

### Admin of Command Post

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Column Name** | **Data Type** | **Size** | **Allow Null** | **Default** |
| PK | AdminCommandPost\_Id | Varchar | 5 | No |  |
|  | Fullname | Varchar | 30 | No |  |
|  | Birthdate | Date | 6 | No |  |
|  | Birthplace | Text | 20 | No |  |
|  | Occupation | Text | 20 | No |  |
|  | Address | Varchar | 100 | No |  |
|  | IdentityNumber | Varchar | 20 | No |  |
|  | Nohp | Varchar | 15 | No |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Email | Varchar | 50 | No |  |
|  | Username | Varchar | 30 | No |  |
|  | Password | Varchar | 20 | No |  |
| FK | NaturalDisaster\_Id | Varchar | 5 | No |  |
| FK | Province\_Id | Varchar | 5 | No |  |
| FK | CommandPost\_Id | Varchar | 5 | No |  |
| FK | Testimonial\_Id | Varchar | 5 | No |  |
| FK | LogisticalStock\_Id | Varchar | 5 | No |  |
| FK | donationEvent\_Id | Varchar | 5 | No |  |
| FK | Volunteer\_Id | Varchar | 5 | No |  |

*Table 3.8.2.11 Database Design Admin of Command Post*

### Command Post

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Column Name** | **Data Type** | **Size** | **Allow Null** | **Default** |
| PK | CommandPost\_Id | Varchar | 5 | No |  |
|  | AdminCoPost\_Id | Varchar | 30 | No |  |
|  | CommandPostName | Varchar | 50 | No |  |
|  | Address | Text | 30 | No |  |
|  | Maps | Text | 50 | No |  |
|  | Area | Text | 30 | No |  |
|  | Regency | Text | 30 | No |  |
|  | Building | Text | 30 | No |  |
|  | PIC | Text | 30 | No |  |
|  | PhoneNo | Varchar | 30 | No |  |
|  | Capacity | Integer | 20 | No |  |
|  | numberOfToilet | Integer | 10 | No |  |
|  | waterResource | Integer | 10 | No |  |
|  | RefugeesVictimNo | Integer | 10 | No |  |
|  | RefugeesTeensNo | Integer | 10 | No |  |
|  | RefugeesElementaryStudNo | Integer | 10 | No |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | RefugeesJuniorStudNo | Integer | 10 | No |  |
|  | RefugeesSeniorStudNo | Integer | 10 | No |  |
|  | RefugeesToddlerNo | Integer | 10 | No |  |
|  | RefugeesPregnantMotherNo | Integer | 10 | No |  |
|  | RefugeesPostpartumMotherNo | Integer | 10 | No |  |
|  | RefugeesElderlyNo | Integer | 10 | No |  |
| FK | LogisticalStock\_Id | Varchar | 5 | No |  |
| FK | Volunteer\_Id | Varchar | 5 | No |  |
| FK | donationEvent\_Id | Varchar | 5 | No |  |

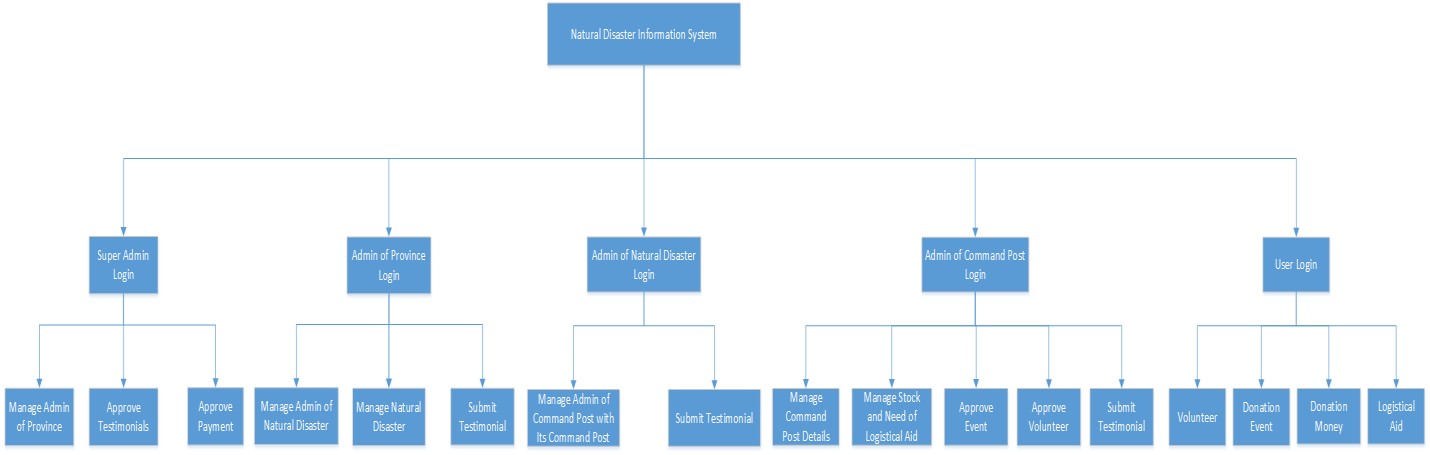
*Table 3.8.2.12 Database Design Command Post*

### Logistical Stock

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Column Name** | **Data Type** | **Size** | **Allow Null** | **Default** |
| PK | LogisticalStock\_Id | Varchar | 5 | No |  |
|  | nameItem | Text | 30 | No |  |
|  | Amount | Text | 30 | No |  |
|  | Unit | Text | 30 | No |  |
|  | Status | Text | 30 | No |  |

*Table 3.8.2.13 Database Design Logistical Stock*

### Structural Design



*Figure 3.8.3.1 Structural Design*

### User Interface

User Interface or we usually call it as UI, is the most critical part for developing the system. UI will be the space where interactions between human and machine occur. User friendly interface equals to great user experience. In designing the user interface of the system, there are many aspects that we have to determine, start from combination of color, layout, navigation, etc.

### Designing Language

We use orange to be primary color of user interface for the user. Orange is a symbol of excitement and creativity. The orange color transfers prime conditions and positive energy to the surroundings, so that one can quickly recover from the disappointment and the wounded heart. Some say that people who love orange are those who are rich in new ideas, as well as those who are sincere. Orange color can also be interpreted as a social activity. With this color, we want to the users can move their heart to help others. We use blue as primary color of user interface for the admin. From a psychological point of view, blue means being trustworthy and responsible. Color has a positive effect on body and mind. The blue color is also impressed as a color that gives significance, something important and trust without giving the impression of grim and evil. On the basis of that, many companies use blue, either on the logo or uniforms of their employees, in the hope of giving a significant impression and trust in their service users. We also want to the administrator that use our system feel confident with their service.

### Color Used

In designing use interface of the system, we combine some color to make the best interface which comfortable for user eyes. We use color variations to give colors to text, icons, backgrounds, menus, navigation and all elements in the user interface system.

### Color for UI of user



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Color** | | | **Hex Color Code** | **Usage** |
|  | | | #fc5830 | Header, footer, link text, button, menu, navigation, divider, text, icon |
|  |  |  | #ffffff | Background, Link text, header |
|  | | |
|  | | | #1a1a1a | Text, paragraph |
|  | | | #848494 | Text, paragraph |
|  | | | #d9d9d9 | Background divider, icon |

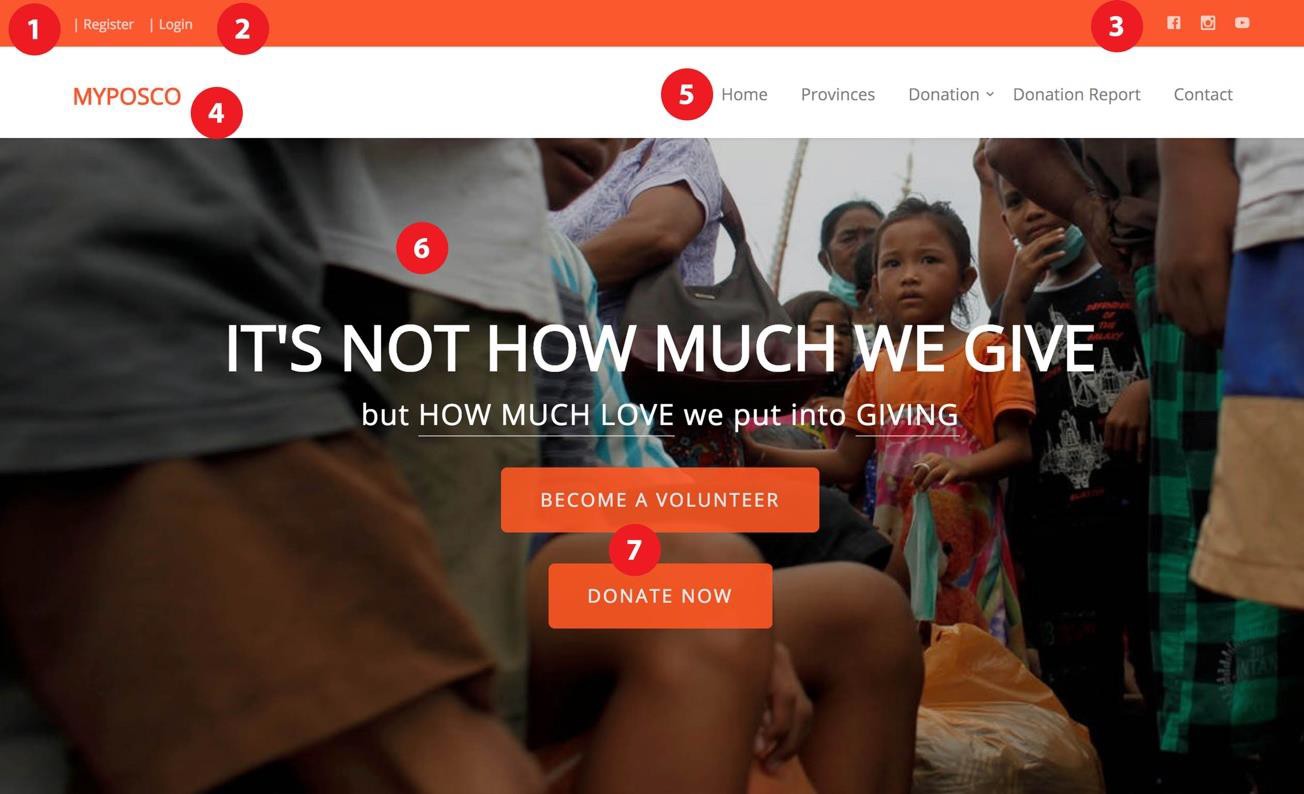
*Figure 3.8.4.2.1 Color for UI of User*

### Color for UI of Admin

|  |  |  |
| --- | --- | --- |
| **Color** | **Hex Color Code** | **Usage** |
|  | #408eba | Header |
|  | #222d32 | Navigation and Menu |
|  | #18a55d | Button Accept |
|  | #db4c3f | Button Trash |
|  | #f19b2c | Button Add and Edit |

*Figure 3.8.4.2.2 Color for UI of Admin*

### Website User Interface for User



Explanation:

* + - * 1. Register

*Figure 3.8.4.3.1 Web UI for User: Homepage Header*

This is link to go to the register form for user.

* + - * 1. Login

This is link to go to the login form for user.

* + - * 1. Social Media

We provide social media link for this website using social media icon.

* + - * 1. Logo / Name of website

This is space for logo or name of website.

* + - * 1. Menu

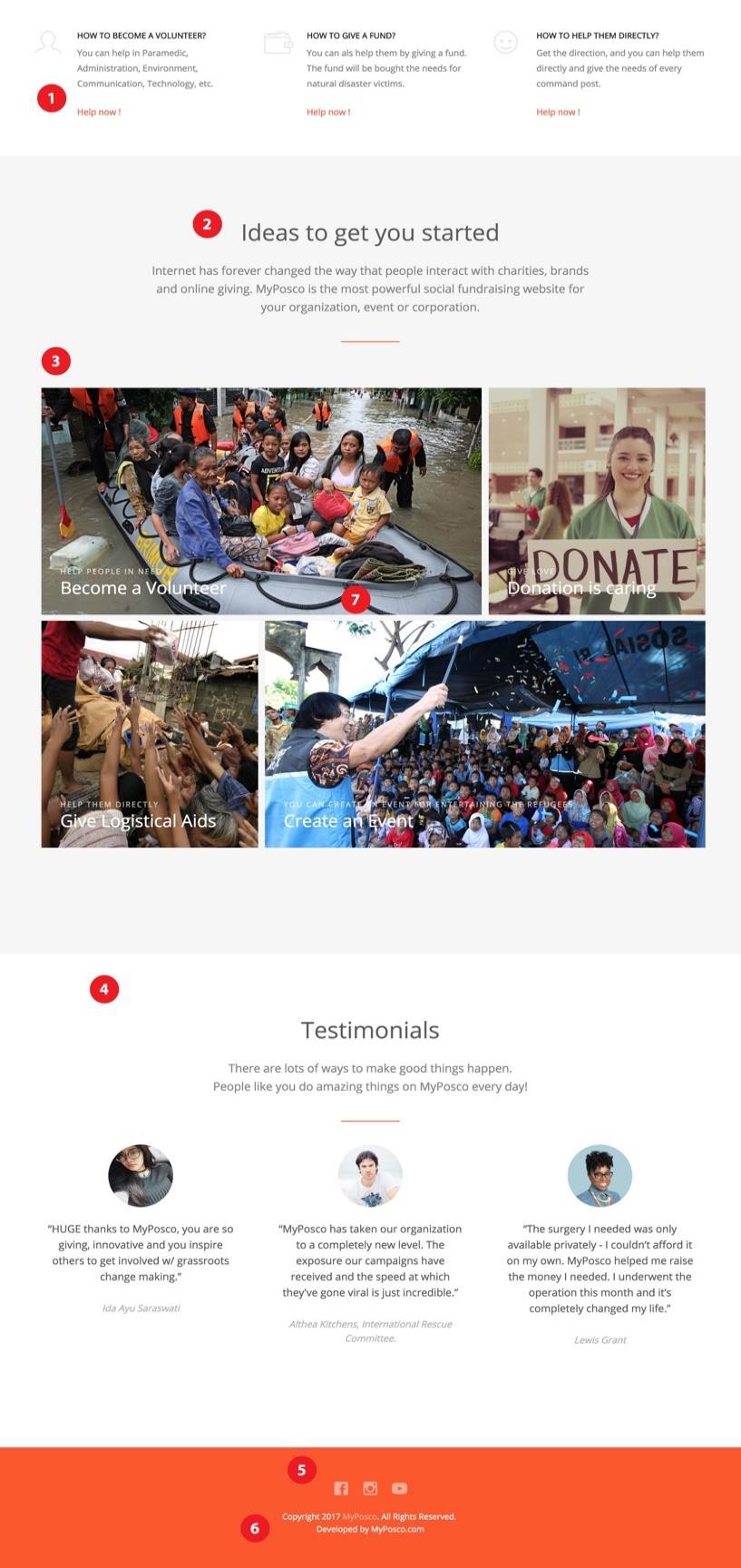
This is the menu navigation.

* + - * 1. Main Banner

This website also provides header banner

* + - * 1. Become volunteer and donate button

This is shortcut button to go to become volunteer page and donate page.



*Figure 3.8.4.3.2 Web UI for User: Homepage Body*

Explanation:

1. Highlight shortcut

There are shortcuts to become volunteer page, donate page and logistical aids page.

1. Section Title

This contains the title of content.

1. Highlight Banner

There are 4 highlight banner to be shortcut to become volunteer page, donate page, logistical aids page and create event page.

1. Testimonial

This will show the testimonial from user and sub-admin (province admin, natural disaster admin and command post admin) based on approval from super admin

1. Social Media

We provide social media link for this website using social media icon.

1. Copyright

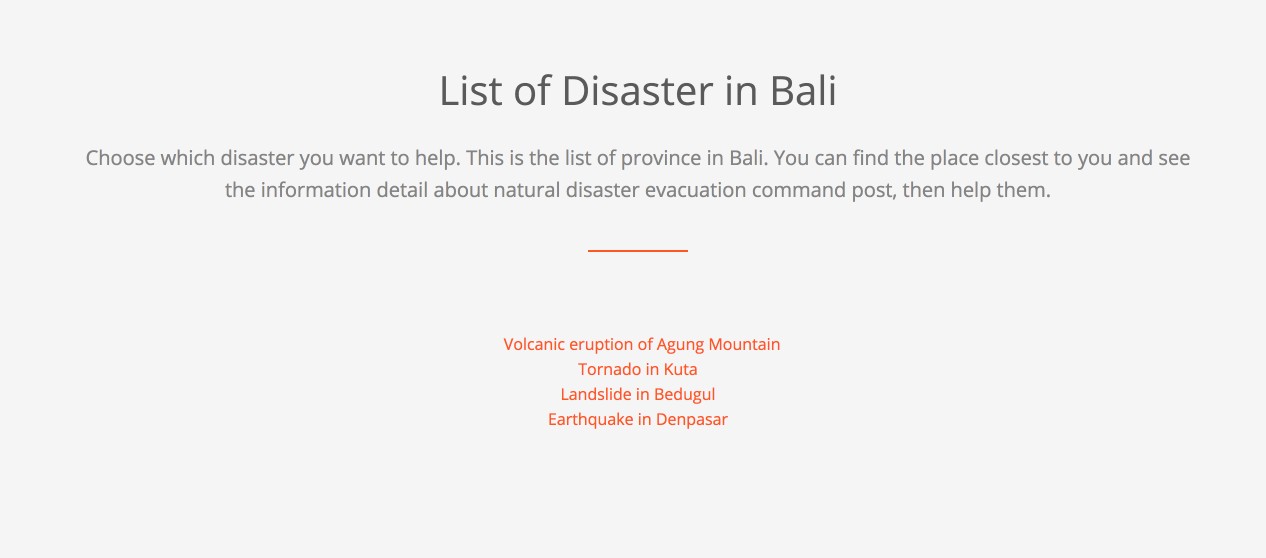
It will display the owner of this website.



*Figure 3.8.4.3.3 Web UI for User: Province Page*

Explanation:

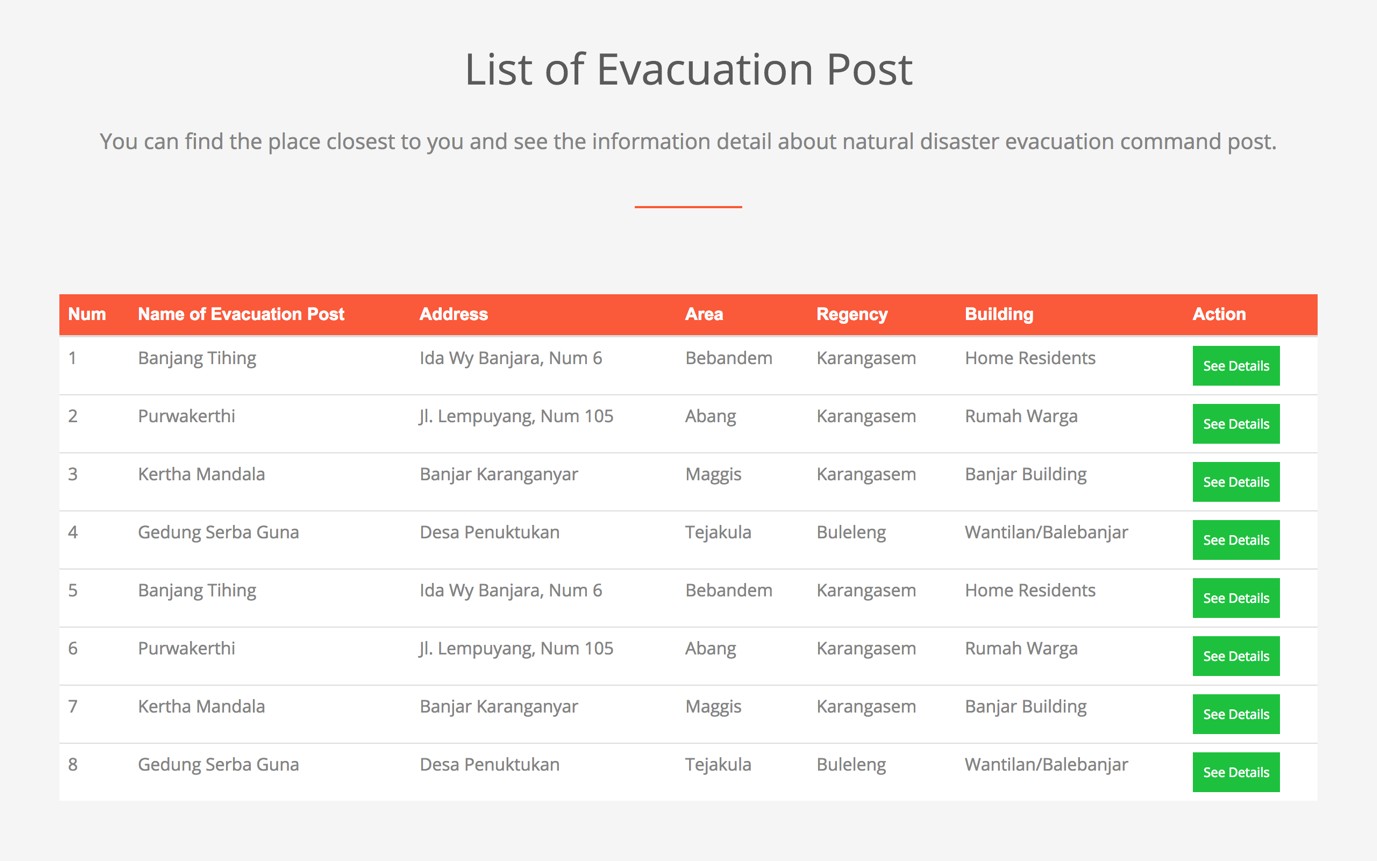
This page will provide list of provinces in Indonesia. User can choose the province that they want to help.



*Figure 3.8.4.3.4 Web UI for User: Natural Disaster Page*

Explanation:

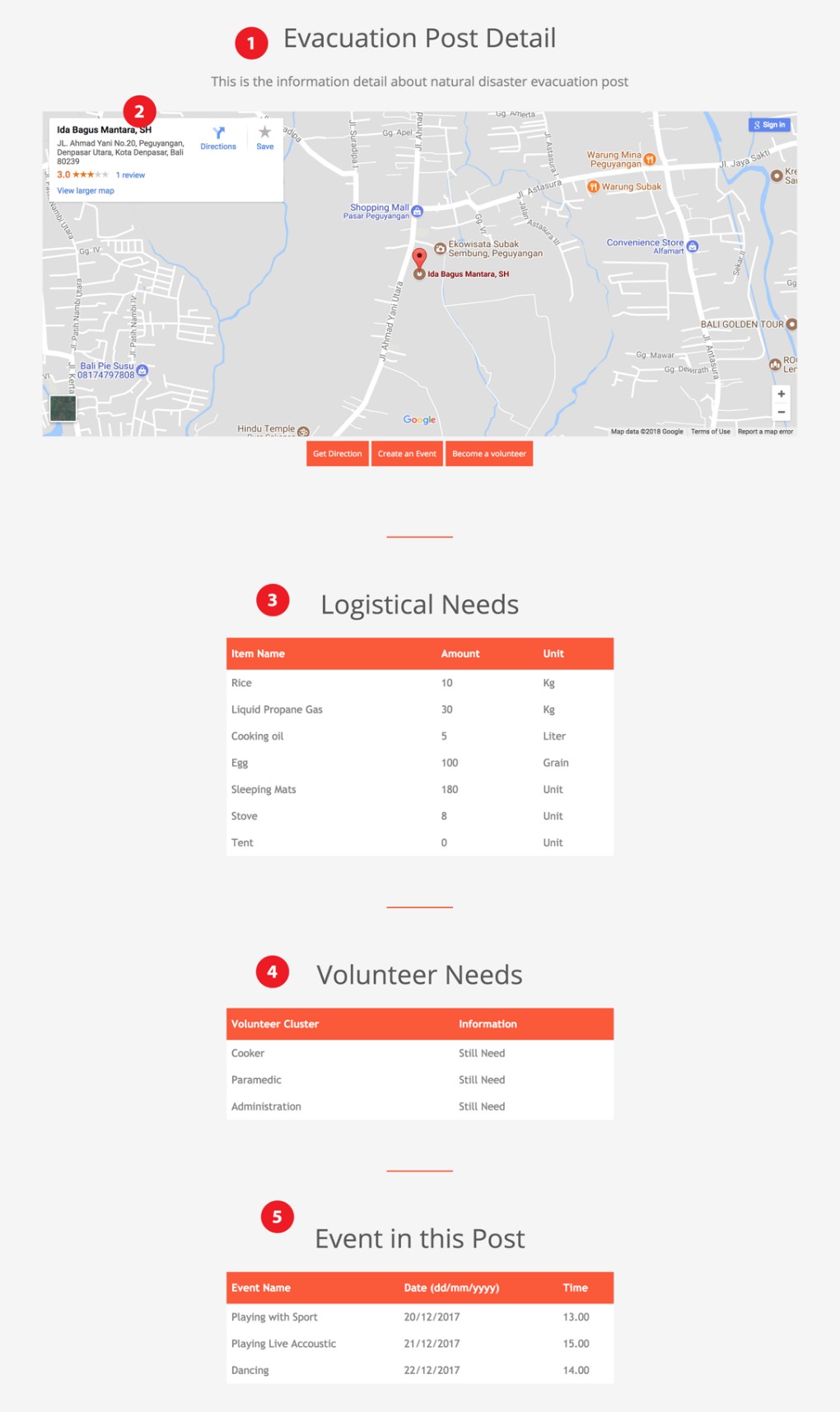
When user choose the province in province page, this page will be showed. This page will provide list of natural disaster in a province. User can choose the disaster that they want to help.



Explanation:

*Figure 3.8.4.3.5 Web UI for User: Command Post List Page*

When user choose the disaster in disaster page, this page will be showed. This page will provide list of command post in a natural disaster page. User can choose the command post that they want to help.



Explanation:

*Figure 3.8.4.3.6 Web UI for User: A Command Post Page*

1. Evaluation Post Detail

This will provide the information about command post.

1. Maps

This map will show the location of the command post. There are also get direction button to get the direction to the command post, create an event button to create the event in the command post, become a volunteer button to register the user become a volunteer in that command post.

1. Logistical Needs

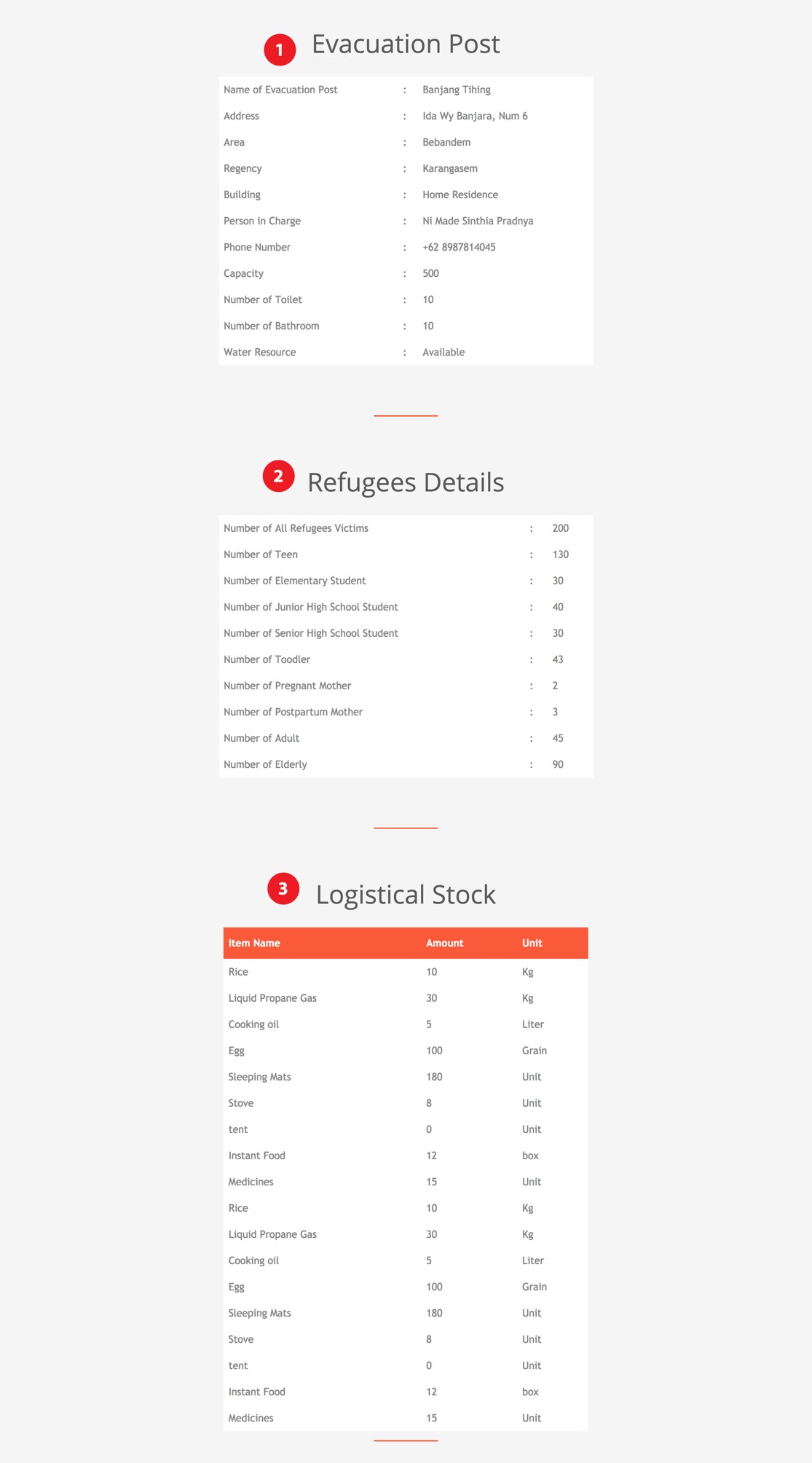
This will provide the list of logistical needs in the command post.

1. Volunteer Needs

This will provide the list of volunteer needs in the command post.

1. Event in this Post

This will provide the list of event in the command post.



*Figure 3.8.4.3.7 Web UI for User: A Command Post Page*

Explanation:

1. Evacuation Post

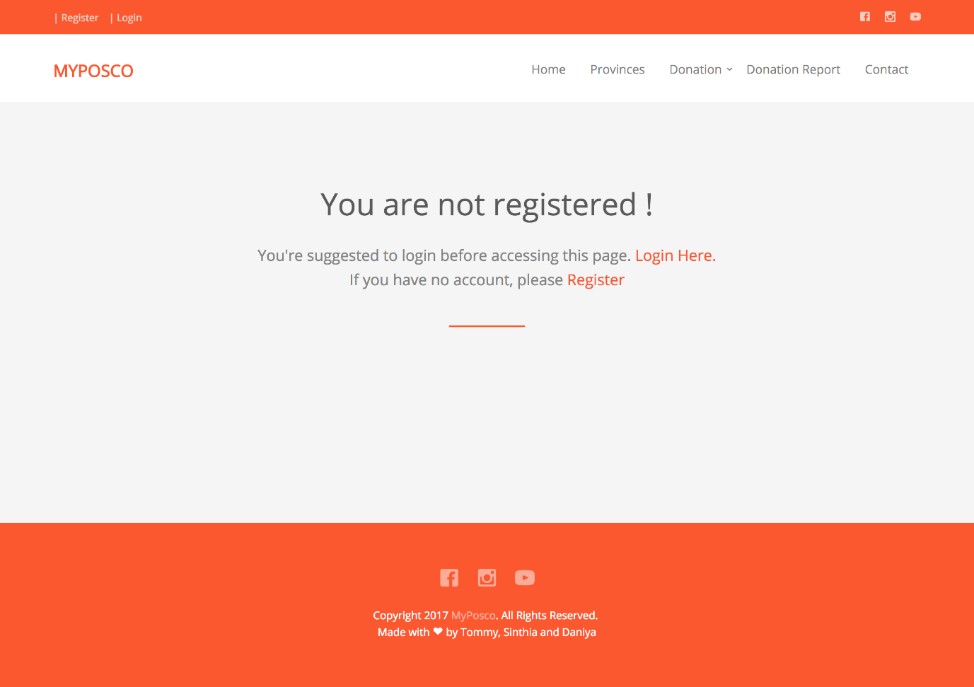
This will show the detail of evacuation post.

1. Refugees Detail

This will show the detail of refugees.

1. Logistical Stock

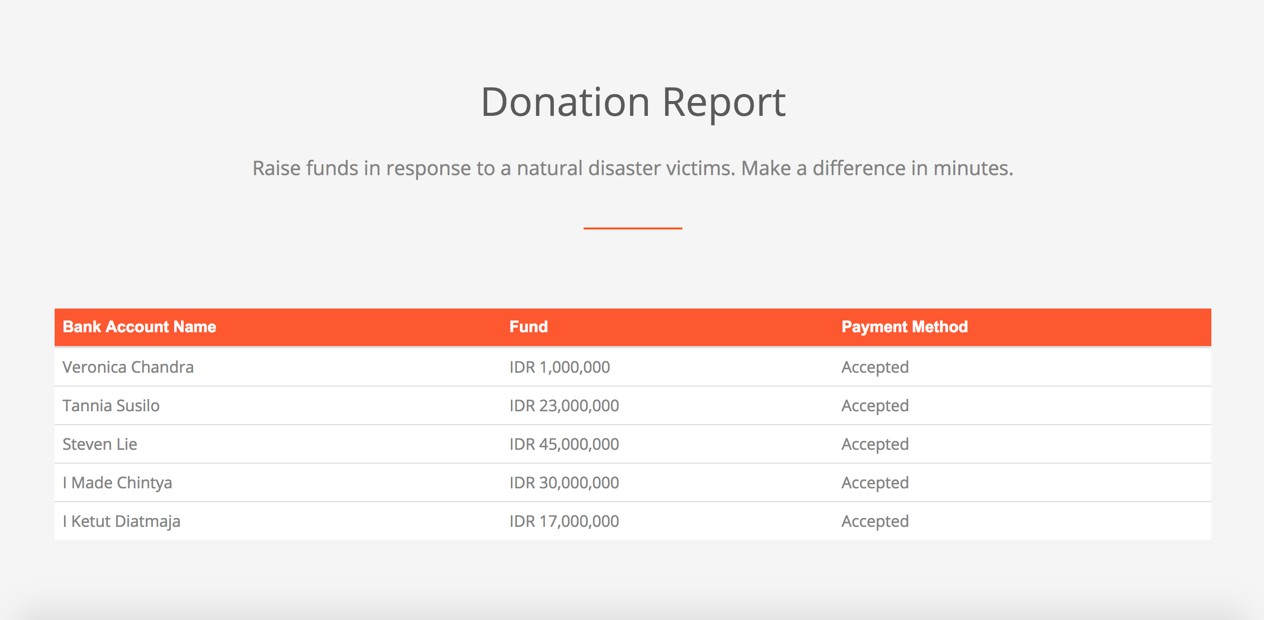
This will show the detail of logistical stock which available.



*Figure 3.8.4.3.8 Web UI for User: Unregistered User*

Explanation:

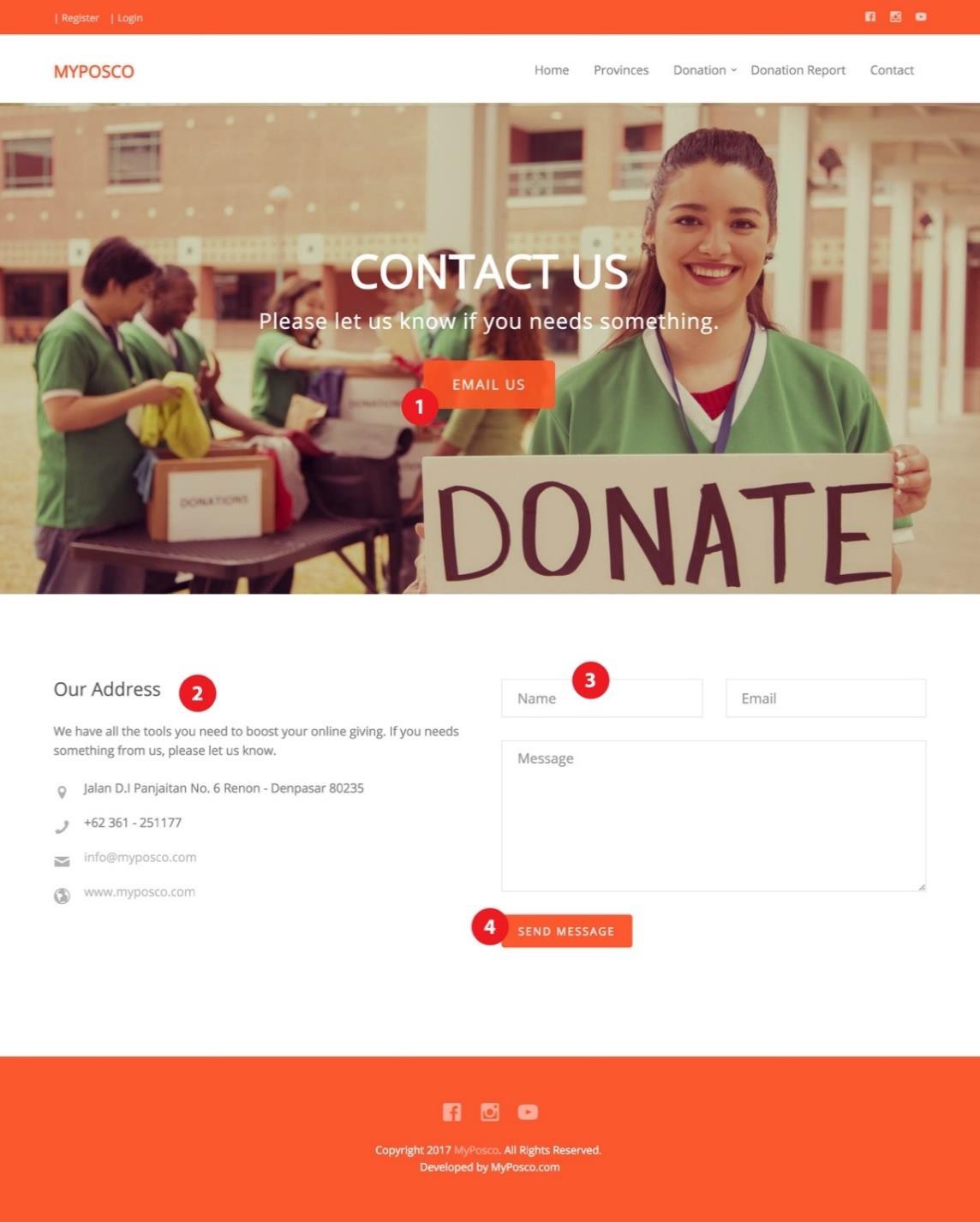
When unregistered user choose to give fund or event from donation menu, this page will be showed.



*Figure 3.8.4.3.9 Web UI for User: Donation Report*

Explanation:

When unregistered, registered user and admins choose the donation report menu, this page will be showed.



Explanation:

*Figure 3.8.4.3.10 Web UI for User: Contact Us Page*

* 1. Email us button

This button will direct the user to email section in the bottom of this website.

* 1. Our Address

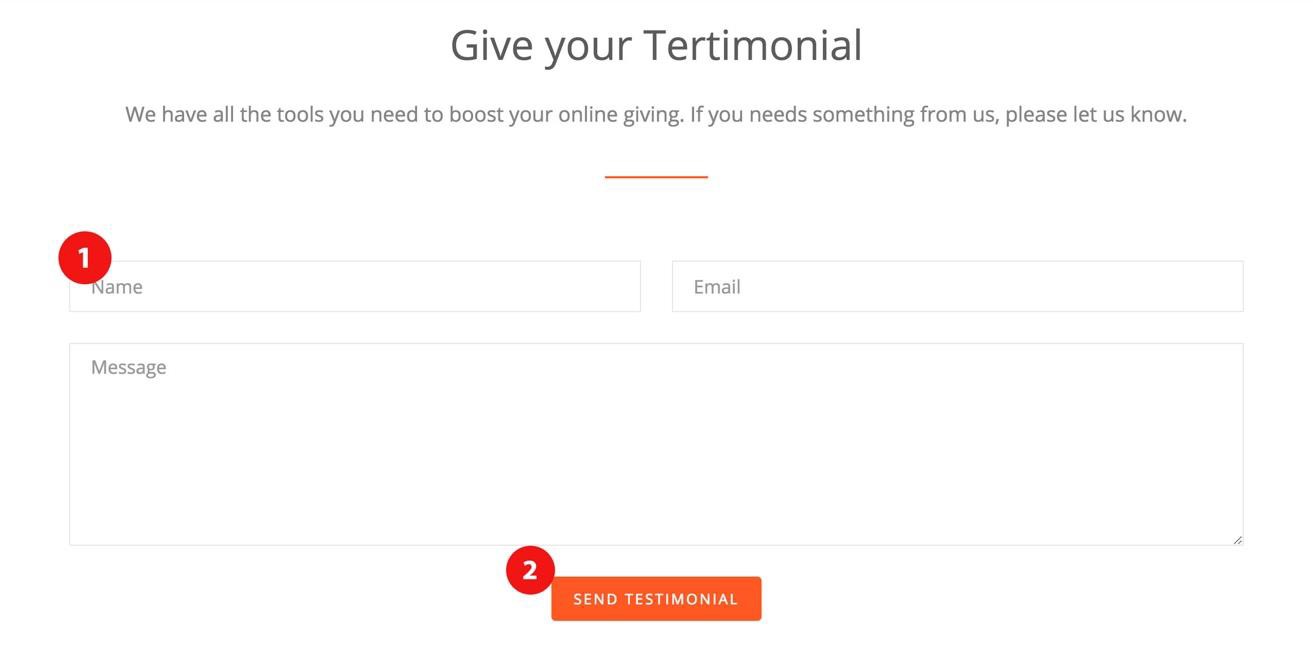
This is the detail information of us such as address, phone number, email and our website link.

* 1. Contact Us Form

All user and admins can use this facility to send us message for further information related to our website.

* 1. Send Message

Through this button, the user or admins can send the message to us.



*Figure 3.8.4.3.11 Web UI for User: Give Your Testimonial Page*

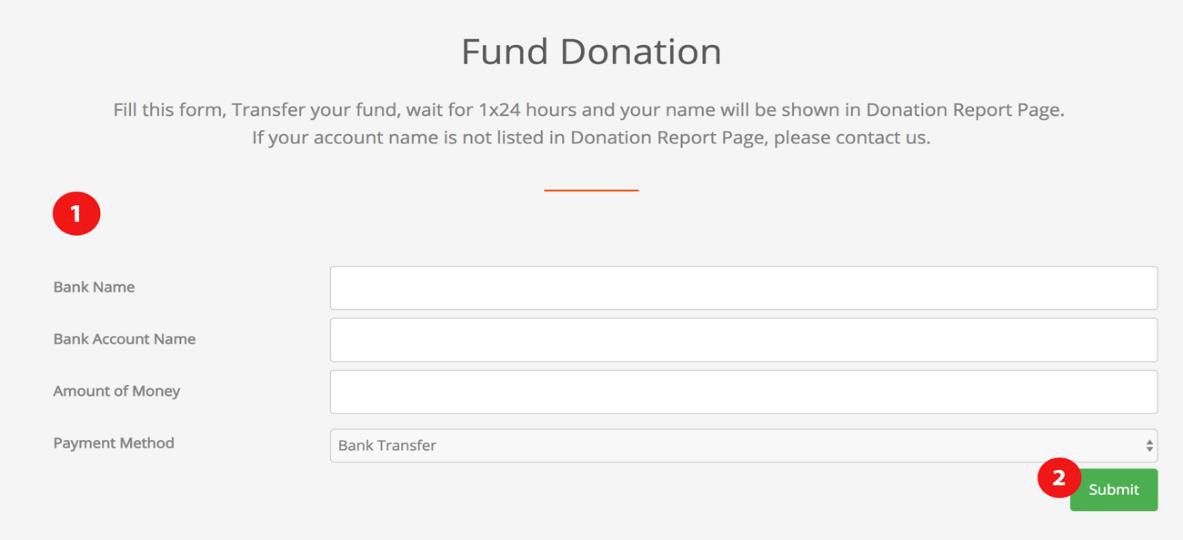
Explanation:

1. Testimonial form

Unregistered, registered user and admins can send their testimonial by filling up this form.

1. Send testimonial

Through this button, the message can be sent to us which handled by super admin.



Explanation:

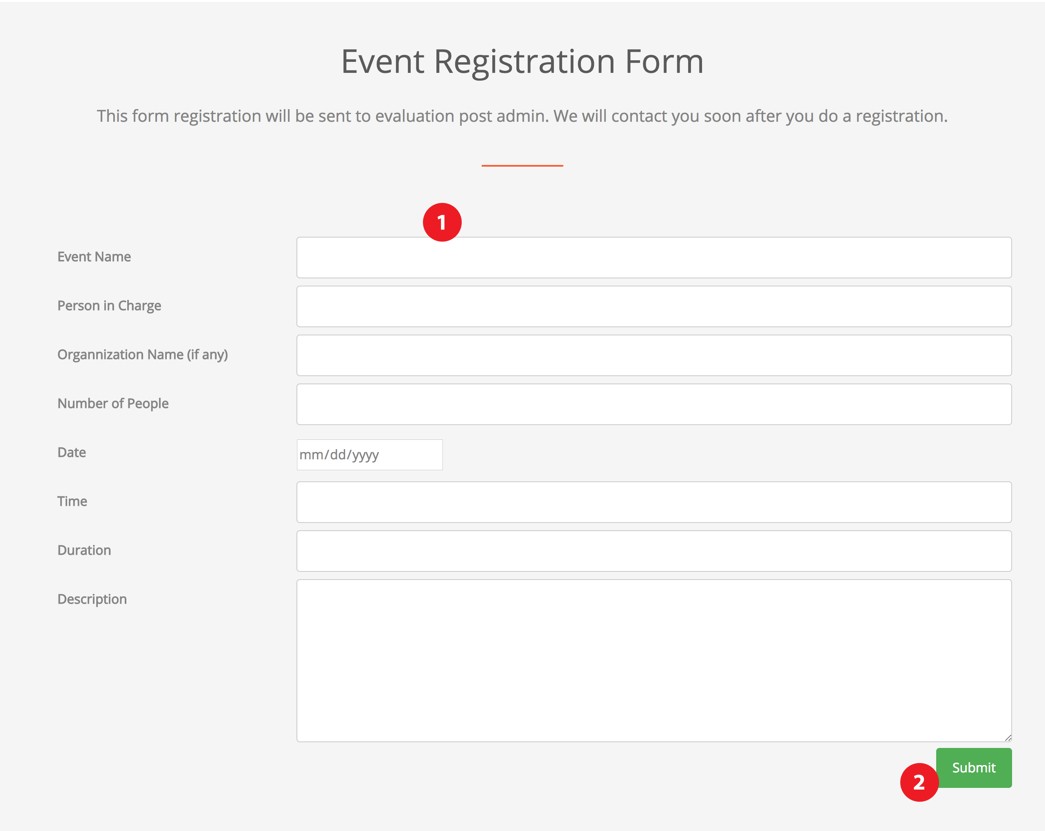
*Figure 3.8.4.3.12 Web UI for User: Donate Fund*

1. Donation fund form

The registered user can apply their fund by filling up this form.

1. Submit button

Through this button, the form of donation fund is sent to us which the approval will be handled by super admin.



*Figure 3.8.4.3.13 Web UI for User: Event Registration Form*

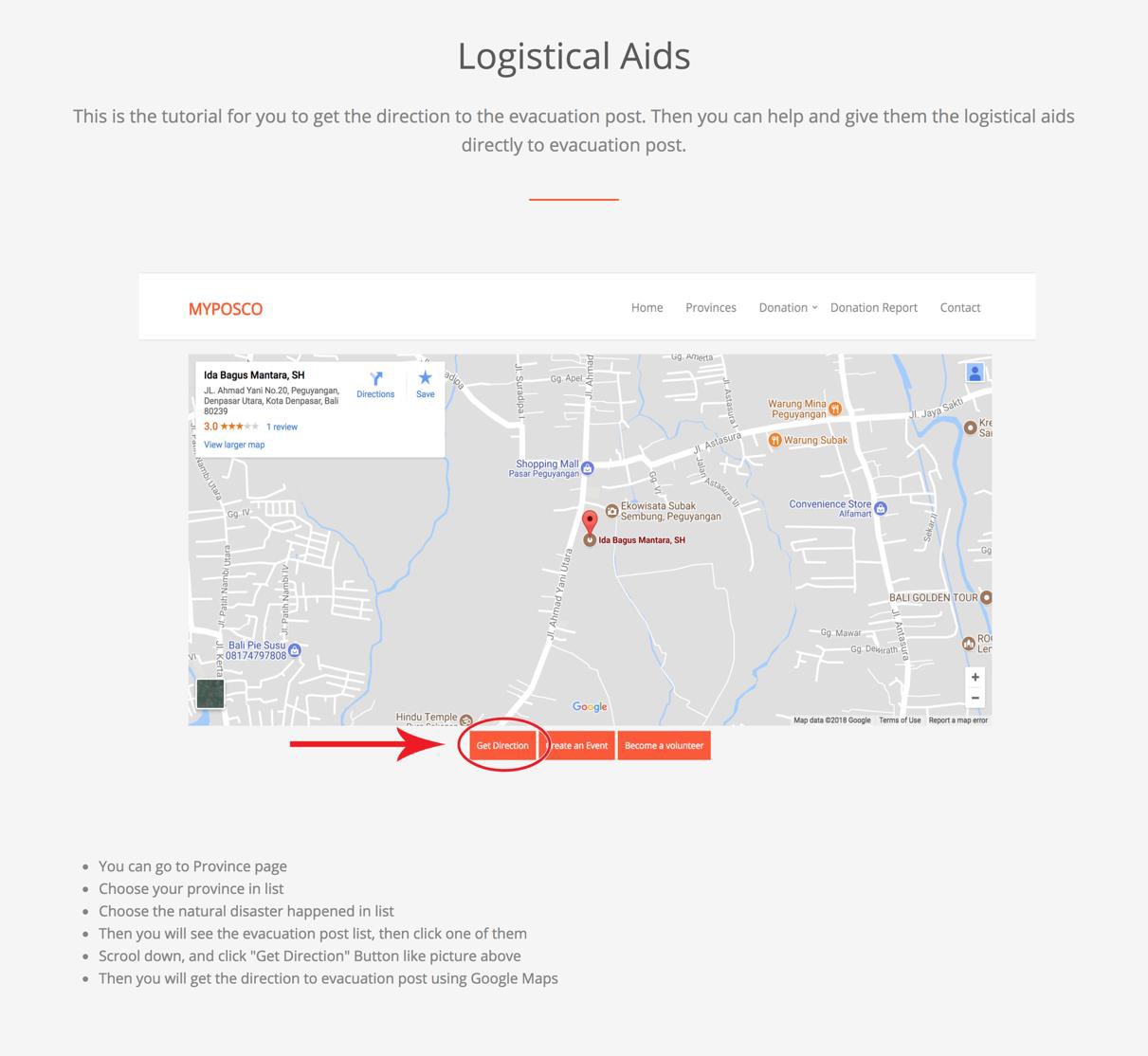
Explanation:

1. Event Registration Form

The registered user can apply their event in command post by filling up this form.

1. Submit button

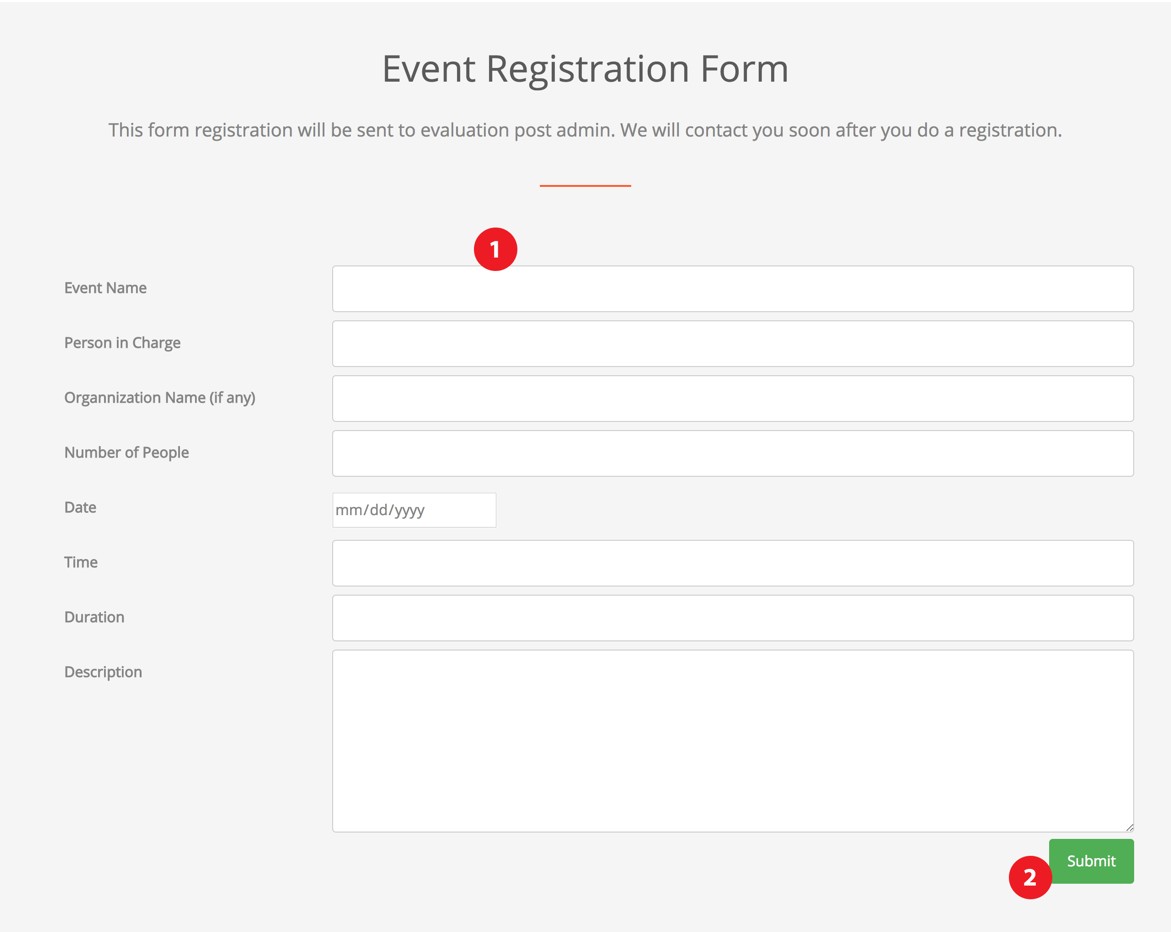
Through this button, the user can send their event registration form which will be confirmed further by the admin of command post.



*Figure 3.8.4.3.14 Web UI for User: Logistical Aids*

Explanation:

When user choose the donation of logistical aids, create an event or become a volunteer button, this page will be showed and tell user about how to donate in the website.



*Figure 3.8.4.3.15 Web UI for User: Volunteer Registration Form*

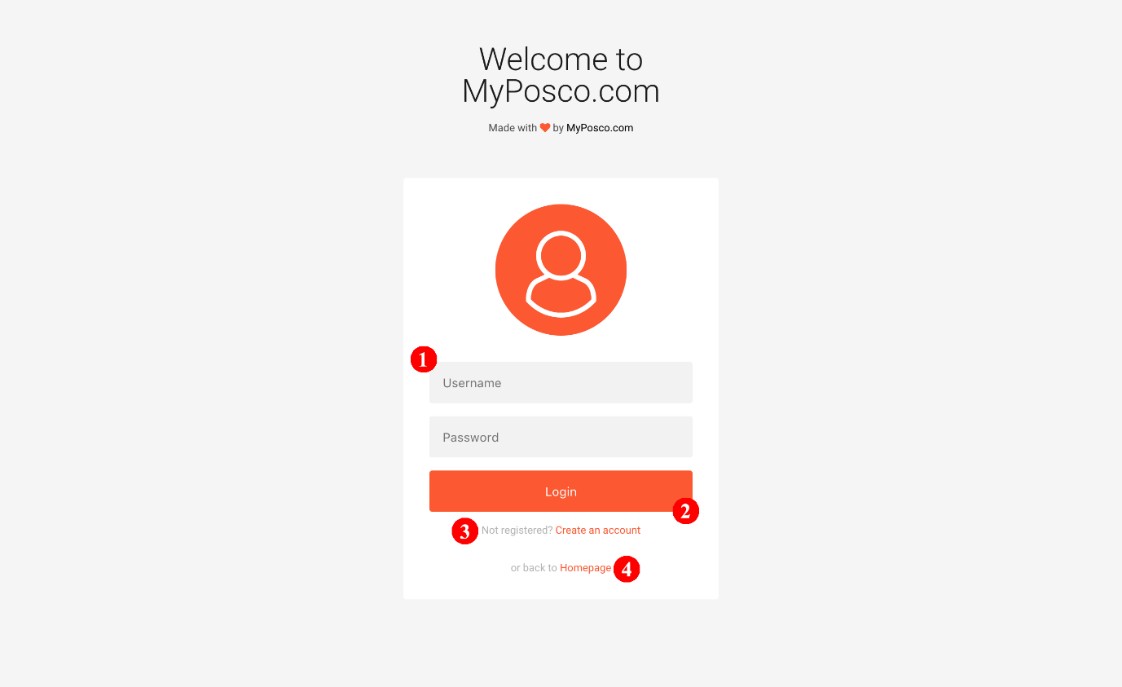
Explanation:

1. Volunteer Registration Form

Registered user can apply themselves as volunteer in command post by filling up this form.

1. Submit

Through this button, the form is sent and the approval will be done by admin command post.



Explanation:

*Figure 3.8.4.3.16 Web UI for User: Login Page*

1. Login Form

The registered user can do login by filling up this form.

1. Login button

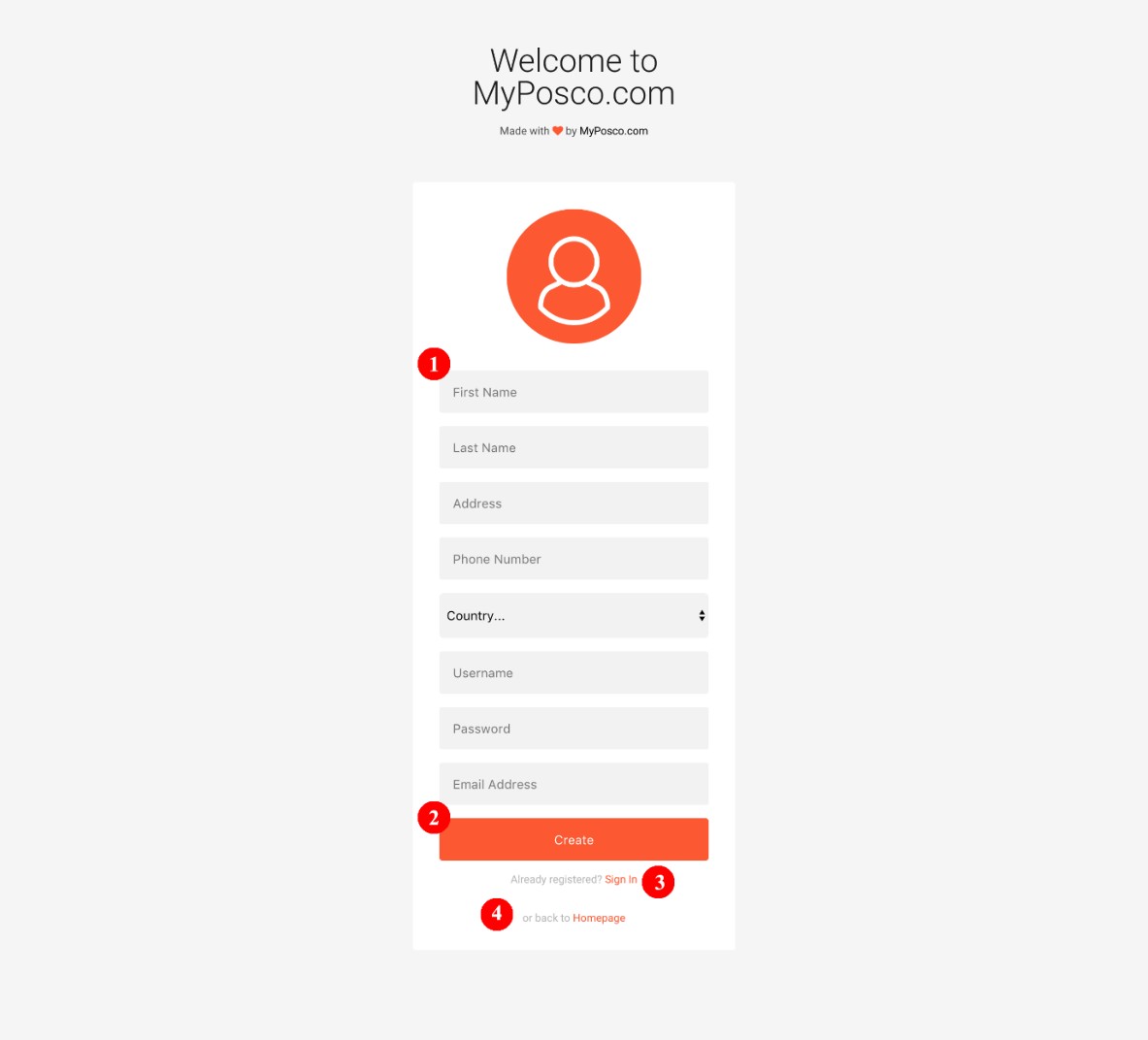
Through this button, the user can enter the homepage of MyPosco.com.

1. Register button

Through this button, the unregistered user can filling up the registration form first.

1. Homepage button

If the user want to go back to home page of MyPosco.com.



Explanation:

*Figure 3.8.4.3.17 Web UI for User: Registration Form Page*

1. Registration Form

The unregistered user can apply as a member of our website by filling up this form.

1. Create button

To create a new member of our MyPosco.com site.

1. Sign In Button

For the registered user to go to sign in or login form.

1. Homepage button

Through this button, the user can go back to homepage of MyPosco.com site.

### Website User Interface for Admin



Explanation:

* + - * 1. Title

*Figure 3.8.4.4.1* Web UI for Admin: Admin Login

The title of page will tell the user that this is the admin login page.

* + - * 1. Icon

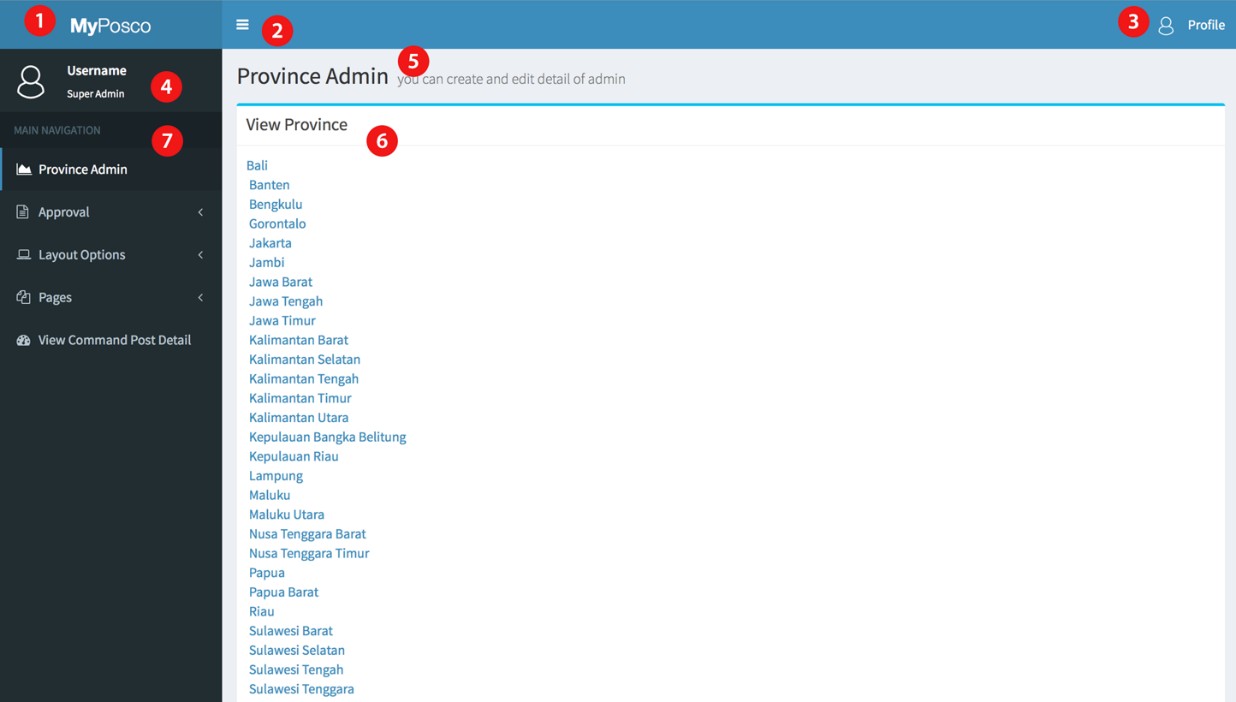
This icon will represent about admin account.

* + - * 1. Form

This is a form filled field for username and password. If the password is wrong, the form will be show the error.

* + - * 1. Login Button

When administrator have done filling up the form require to login, the admin is asked to click this button. After that, the website will show an administrator panel based on admin rule, such as super admin, province admin, disaster admin, or command post admin.



*Figure 3.8.4.4.2* Web UI for Admin: Super Admin Panel(Dashboard-View Province)

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

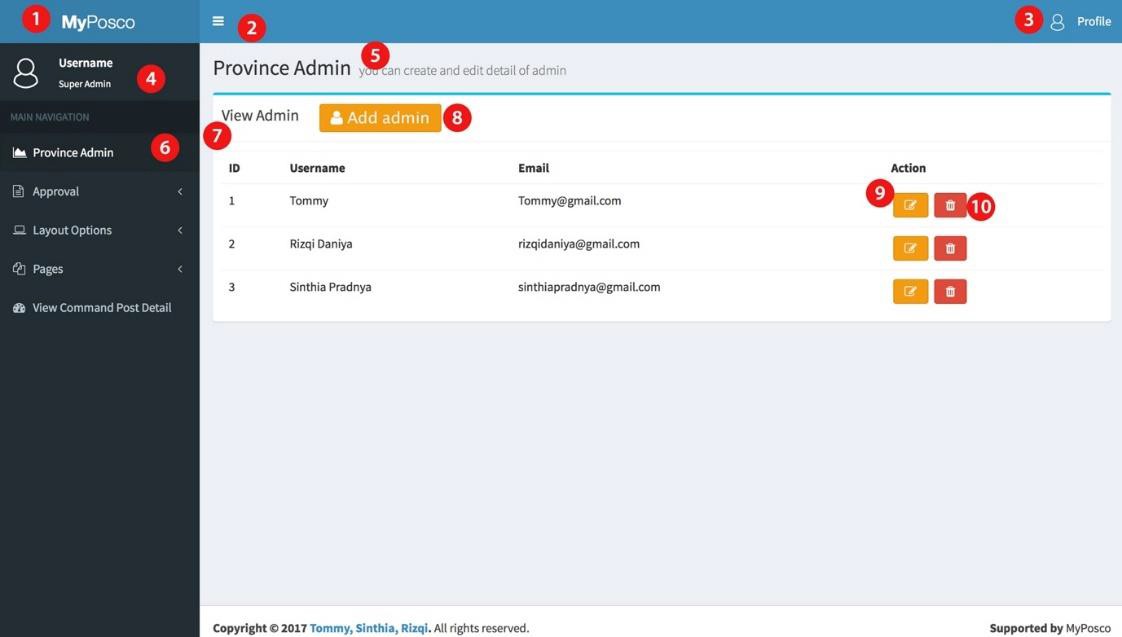
After the navigation is clicked, the title page will show base on navigation

1. Sub-title page

This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Sidebar navigation

This is navigation which can be clicked and show what page which you want to be managed



*Figure 3.8.4.4.3* Web UI for Admin: Super Admin Panel (View Admin Province)

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Add admin

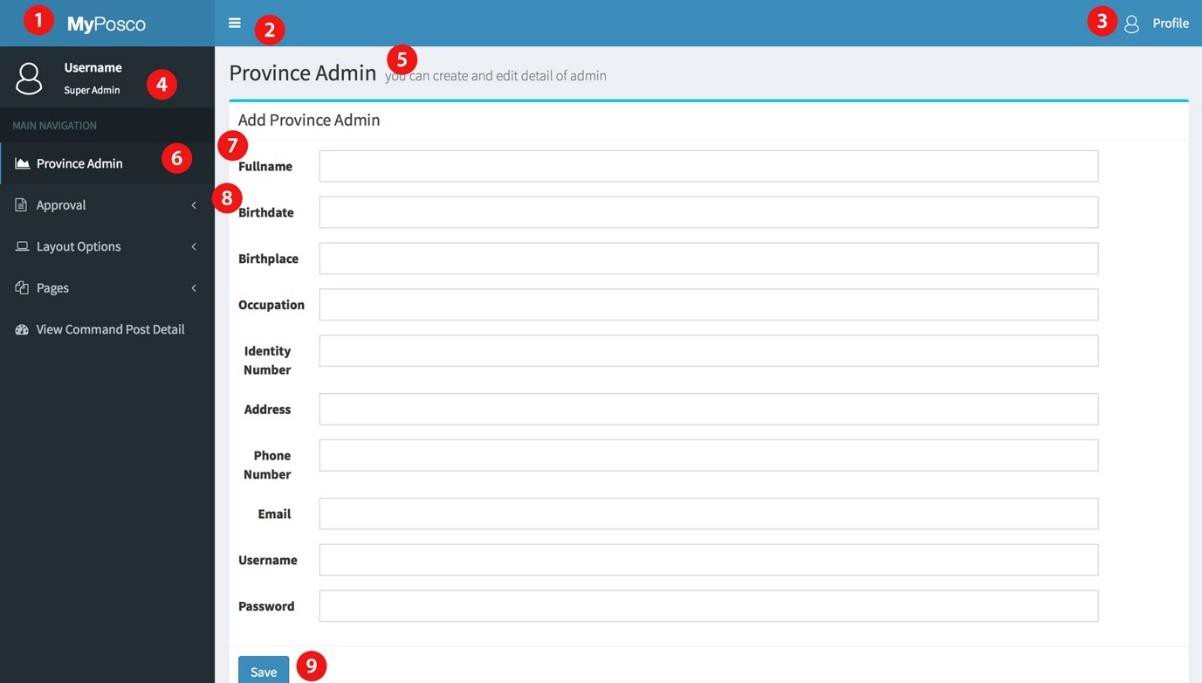
After click add admin button, admin will direct to add admin page.

1. Edit admin

After click edit admin button, admin will direct to edit admin page

1. Delete admin

After click edit admin button, admin will be deleted.



*Figure 3.8.4.4.4* Web UI for Admin: Super Admin Panel(Add Admin Province)

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Add admin form

There are some field that required to be filled up before adding the admin.

1. Save admin

Click save and admin will be added to view admin.



*Figure 3.8.4.4.5* Web UI for Admin: Super Admin Panel(Edit Admin Province)

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

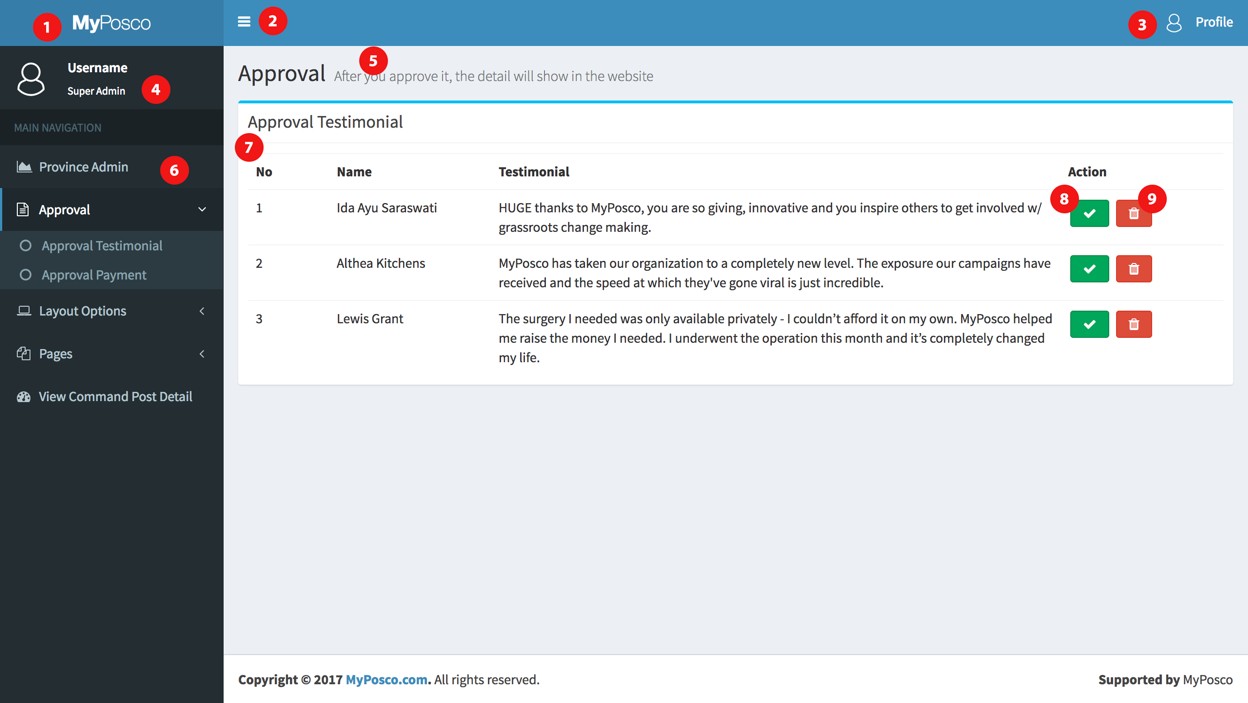
This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Edit admin form

admin can make changes for each admin and edit province admin form will show the last save.

1. Save admin

Click save and admin will be edit the admin.



*Figure 3.8.4.4.6 Web UI for Admin: Super Admin Panel (Approval Testimonial)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

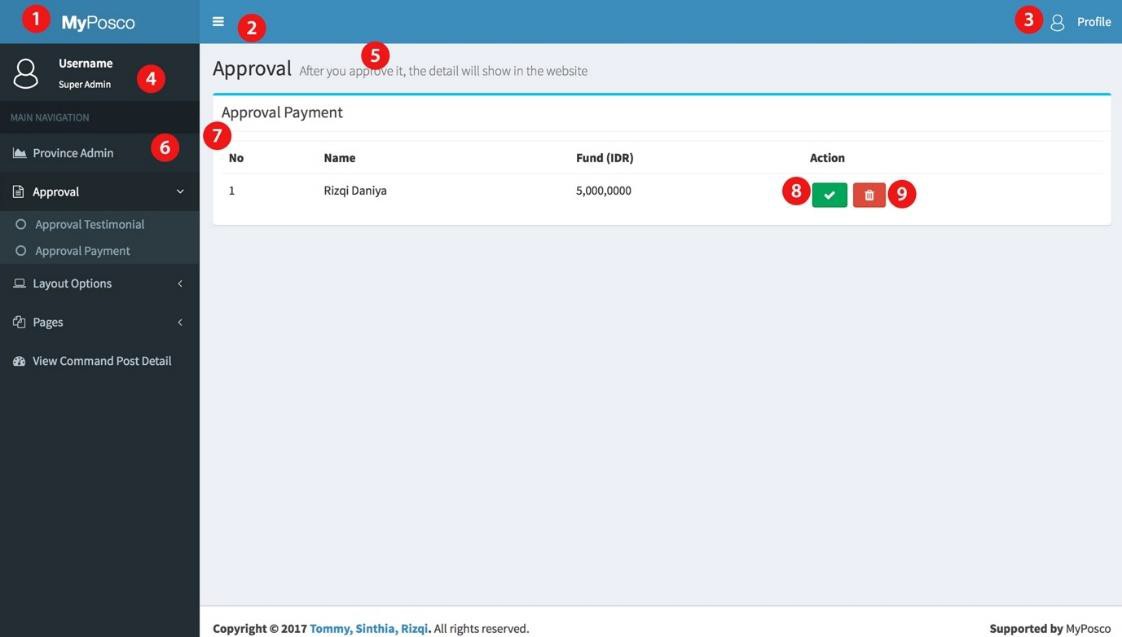
This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Accept testimonial

By click this button, admin can accept the testimonial or not, if admin accept this testimonial, it will be showed in the webpage.

1. Ignore testimonial

If admin ignore the testimonial, it will be deleted from the testimonial list.



*Figure 3.8.4.4.7 Web UI for Admin: Super Admin Panel (Approval Payment)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

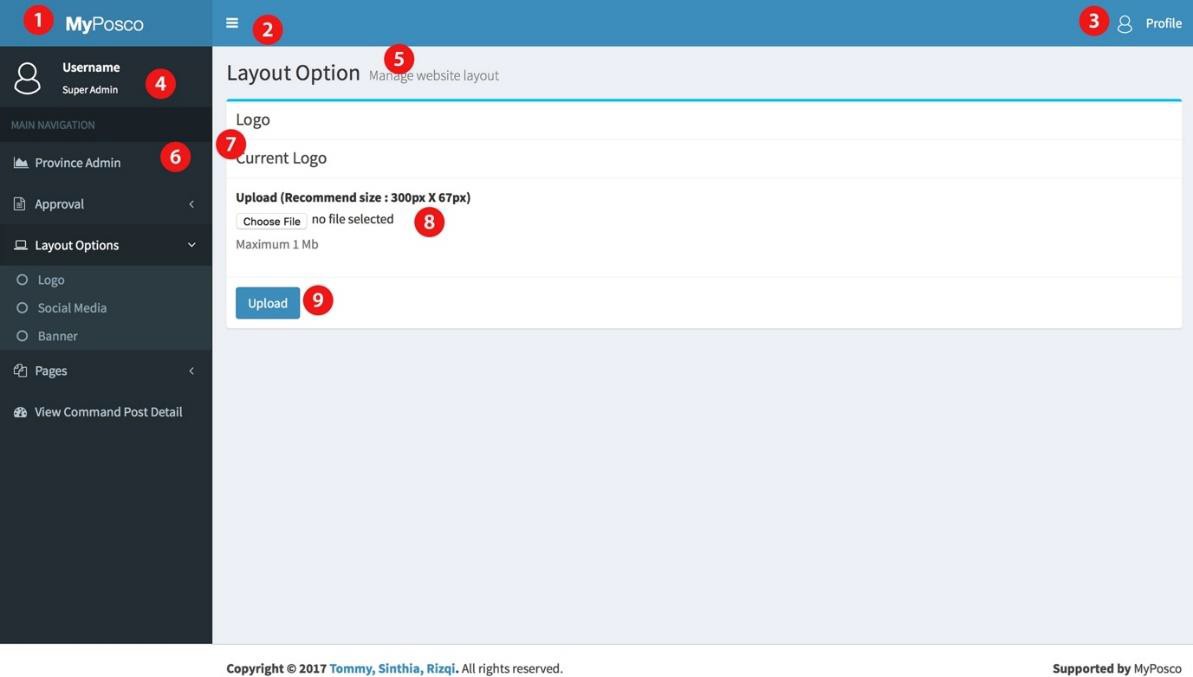
This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Accept payment

By click this button, admin can accept the payment or not, if admin accept this payment, it will be showed in the webpage.

1. Ignore payment

If admin ignore the payment, it will be deleted from the payment list.



*Figure 3.8.4.4.8 Web UI for Admin: Super Admin Panel (Layout Option-Logo)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

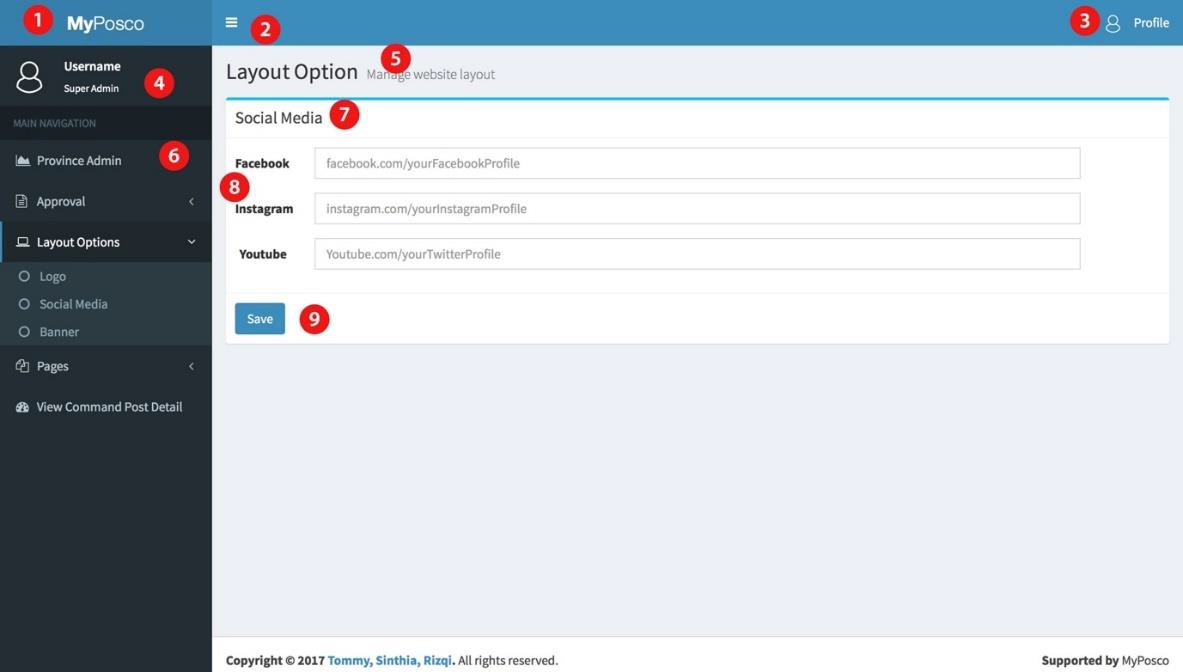
This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Upload photo

For the logo, there is a button to upload the logo.

1. Upload

This button will process to upload the logo into website database.



*Figure 3.8.4.4.9 Web UI for Admin: Super Admin Panel (Layout Option- Social Media)*

Explanation:

* 1. Logo

We can put domain name, name of application or put a logo here.

* 1. Menu show/hide

We provide the button to hide or show sidebar navigation.

* 1. Profile button

The admin can look their profile and we provide log out button here.

* 1. Detail admin

This is detail of admin in form of username and the rule of admin

* 1. Title page

After the navigation is clicked, the title page will show base on navigation

* 1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

* 1. Sub-title page

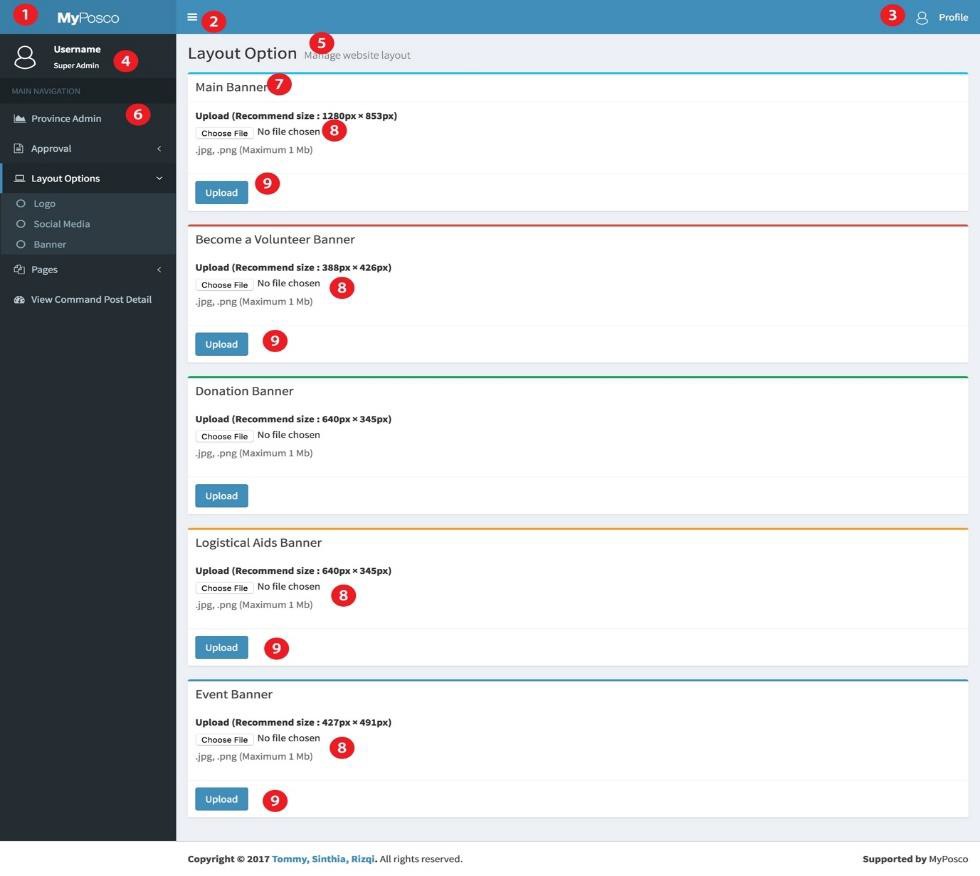
This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

* 1. Social Media Form

There are some field to input link the social media.

* 1. Save

This button will process the link and upload it into database.



*Figure 3.8.4.4.10 Web UI for Admin: Super Admin Panel (Layout Option-Banner)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

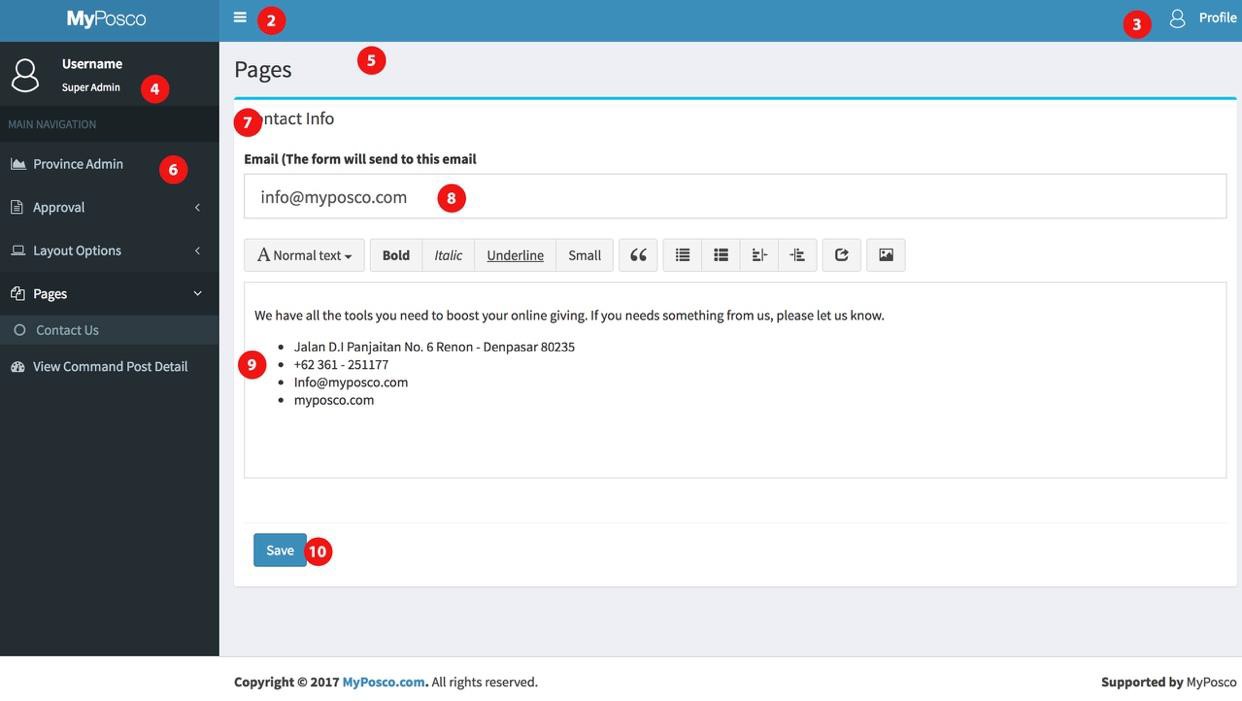
This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Upload photo

For the logo, there is a button to upload the banner. There are main banner, become a volunteer banner, donation banner, logistical banner and create an event banner.

1. Upload

This button will process to upload the banner into website database.



*Figure 3.8.4.4.11 Web UI for Admin: Super Admin Panel(Pages-Contact Us)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Email

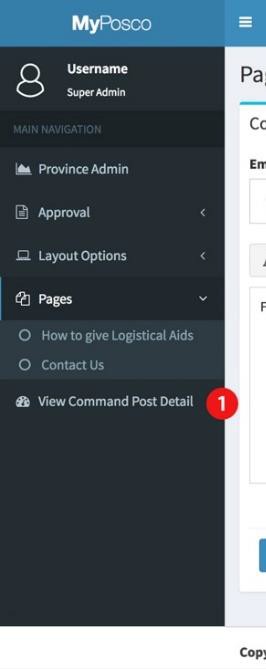
Is required to fill this field, the email will show into contact us page and user can send the email to admin via contact form in contact us page.

1. Content

This is field to manage the content of contact us page.

1. Save

This button will process the form and save it into database.



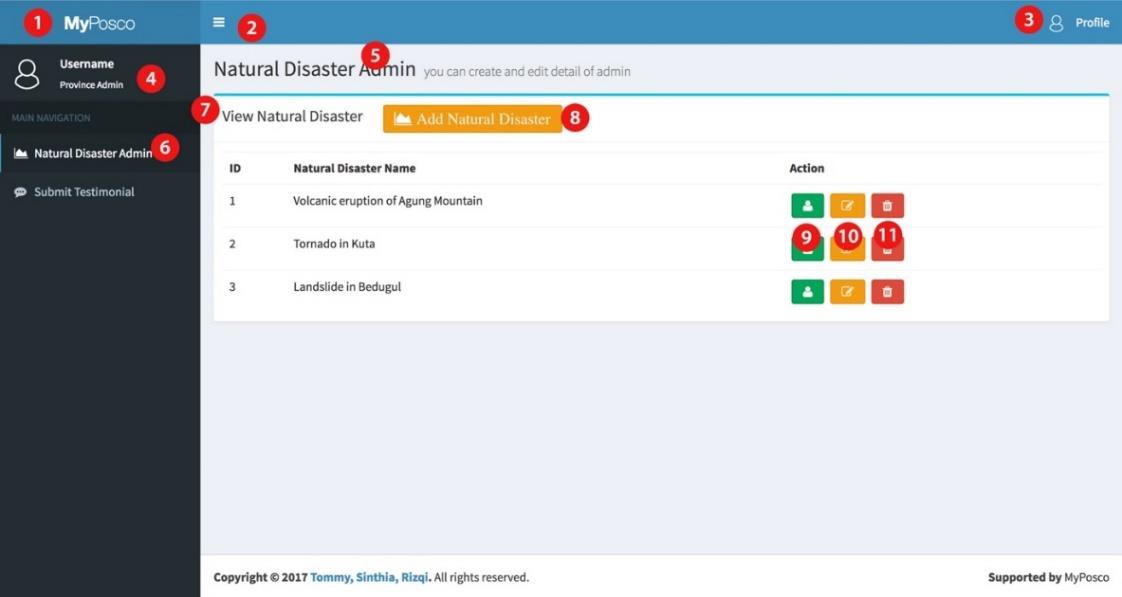
*Figure 3.8.4.4.12 Web UI for Admin: Super Admin Panel (View*

*Command Post Detail)*

Explanation:

1. Command post detail

This button will direct admin to command post detail.



*Figure 3.8.4.4.13 Web UI for Admin: Province Admin Panel (View*

*Natural Disaster)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

This will show the navigation which is provide by natural disaster admin such as view province, view admin, add admin, edit admin, etc.

1. Add natural disaster

After click add natural disaster button, admin will direct to add natural disaster page.

1. View Admin

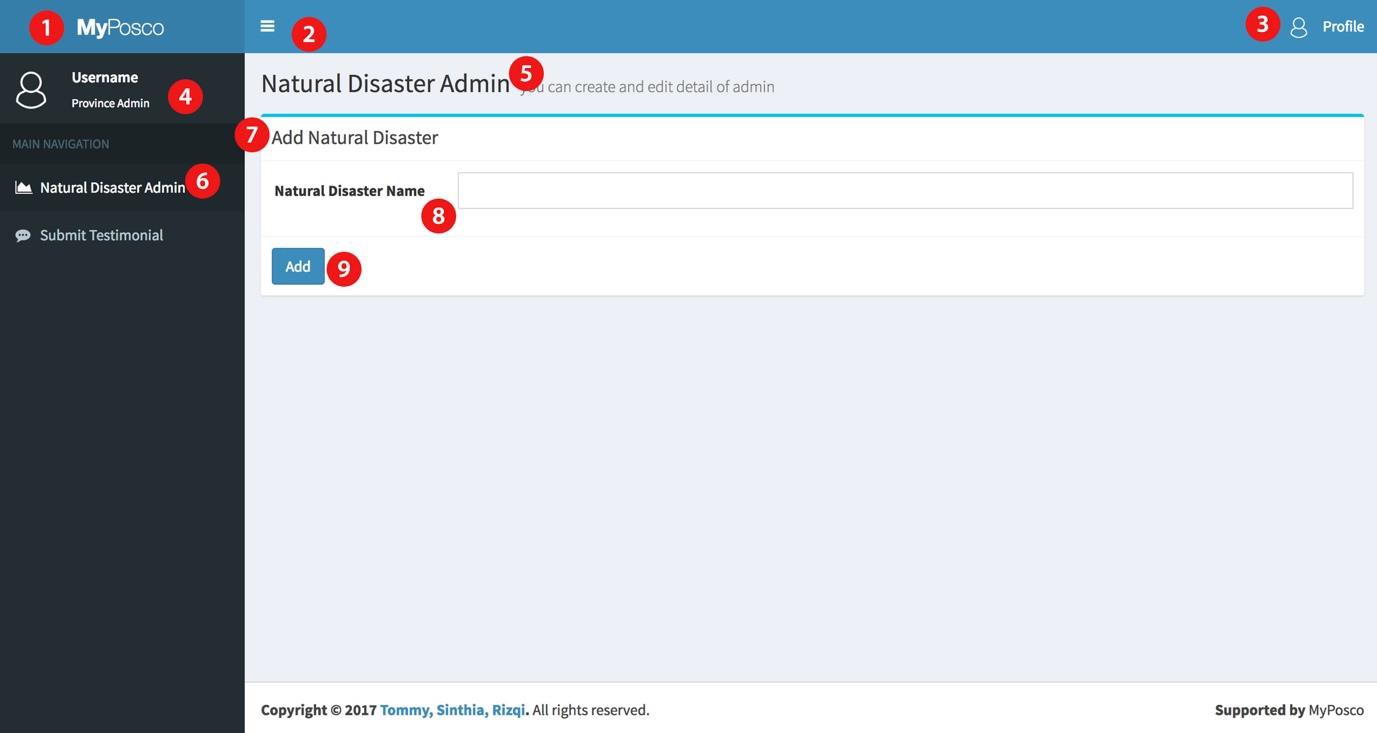
It will show the list of admin in disaster selected.

1. Edit natural disaster

After click edit natural disaster button, admin will direct to edit natural disaster page

1. Delete natural disaster

After click edit admin button, admin will be deleted



*Figure 3.8.4.4.14 Web UI for Admin: Province Admin Panel (Add Natural*

*Disaster)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

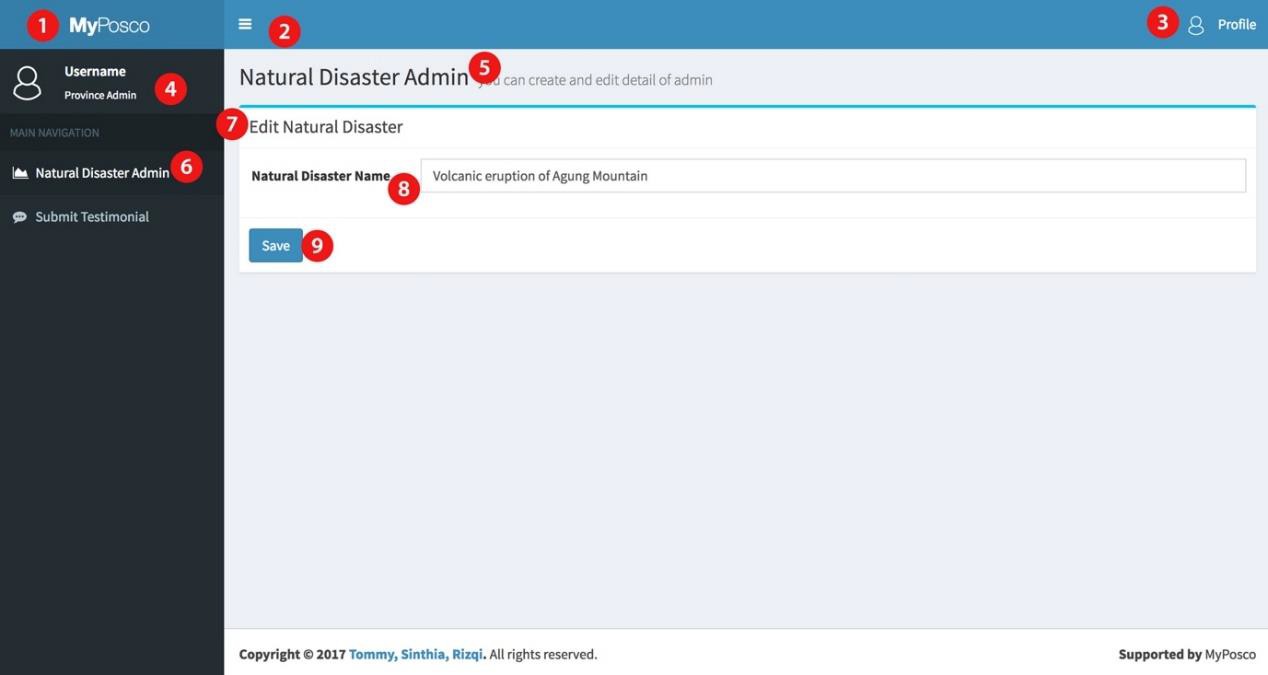
This will show the navigation which is provide by natural disaster admin such as view province, view admin, add admin, edit admin, etc.

1. Add natural disaster field

There are a field that required to be filled up before adding the natural disaster.

1. Add

Click add and natural disaster will be added to view natural disaster.



*Figure 3.8.4.4.15 Web UI for Admin: Province Admin Panel (Edit Natural*

*Disaster)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

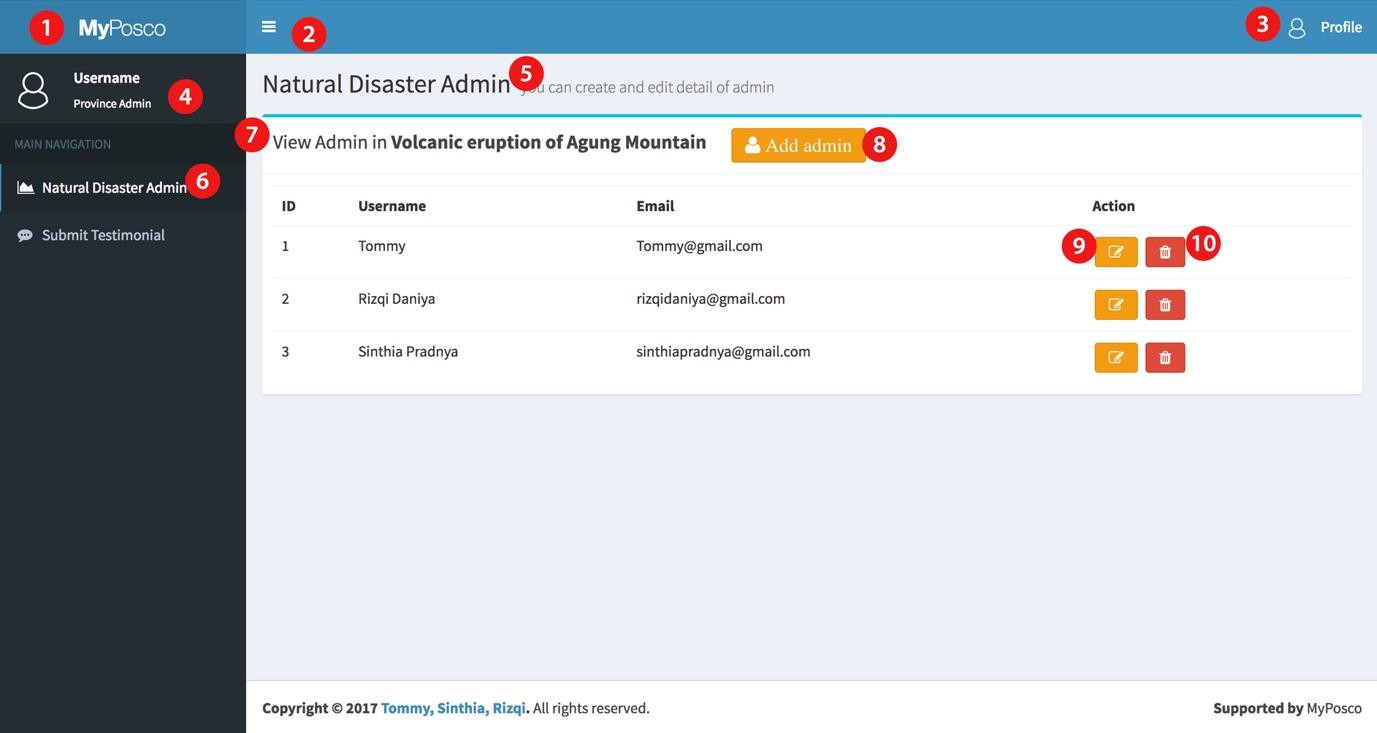
This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Edit natural disaster field

admin can make change or edit natural disaster name, and this field will show the last change.

1. Save

Click save and natural disaster will update the natural disaster.



*Figure 3.8.4.4.16 Web UI for Admin: Province Admin Panel (View*

*Admin)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Add admin

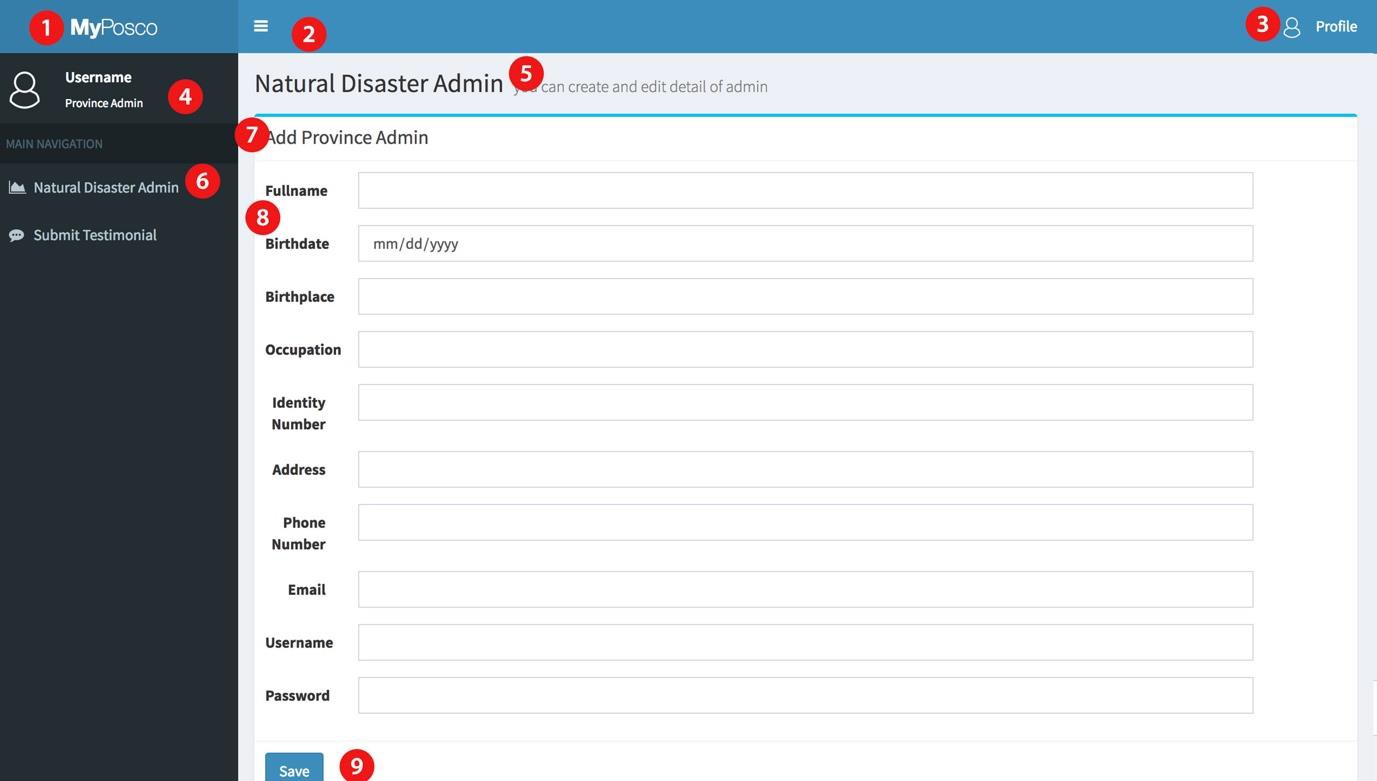
After click add admin button, admin will direct to add admin page.

1. Edit admin

After click edit admin button, admin will direct to edit admin page

1. Delete admin

After click edit admin button, admin will be deleted.



*Figure 3.8.4.4.17 Web UI for Admin: Province Admin Panel (Add Admin*

*Natural Disaster)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

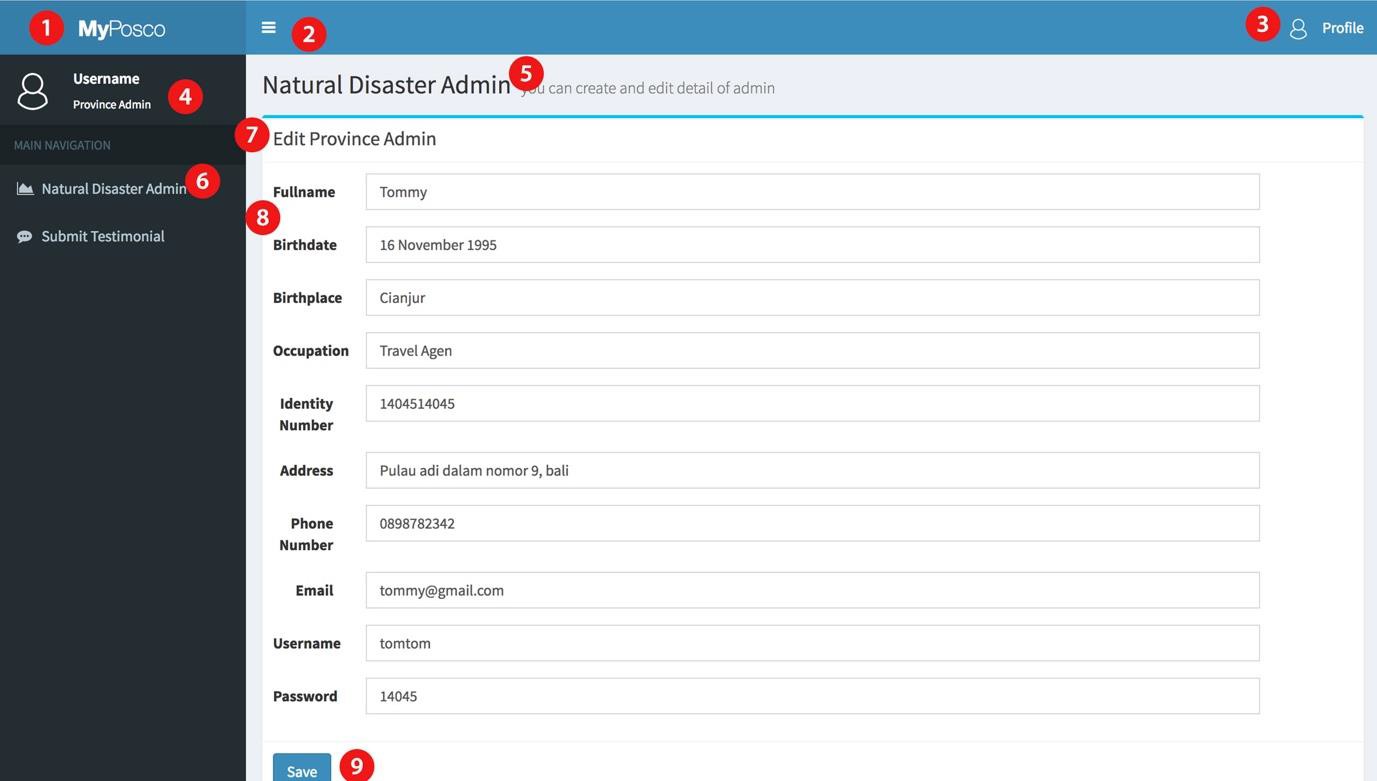
This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Add admin form

There are some field that required to be filled up before adding the admin.

1. Save admin

Click save and admin will be added to view admin.



*Figure 3.8.4.4.18 Web UI for Admin: Province Admin Panel (Edit Admin*

*Natural Disaster)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

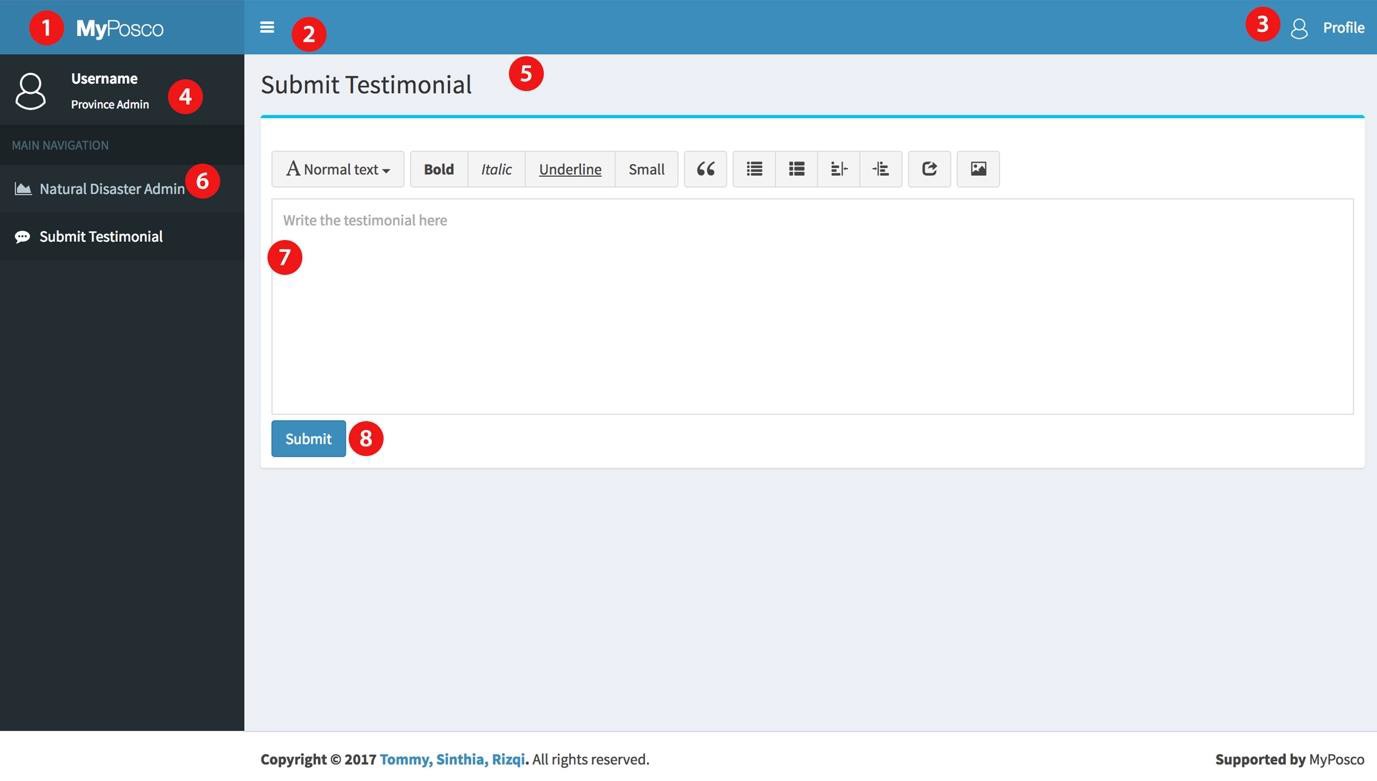
This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Edit admin form

admin can make changes for each admin and edit province admin form will show the last save.

1. Save admin

Click save and admin will update the admin.



Explanation:

*Figure 3.8.4.4.19 Web UI for Admin: Province Admin Panel*

*(Testimonial)*

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

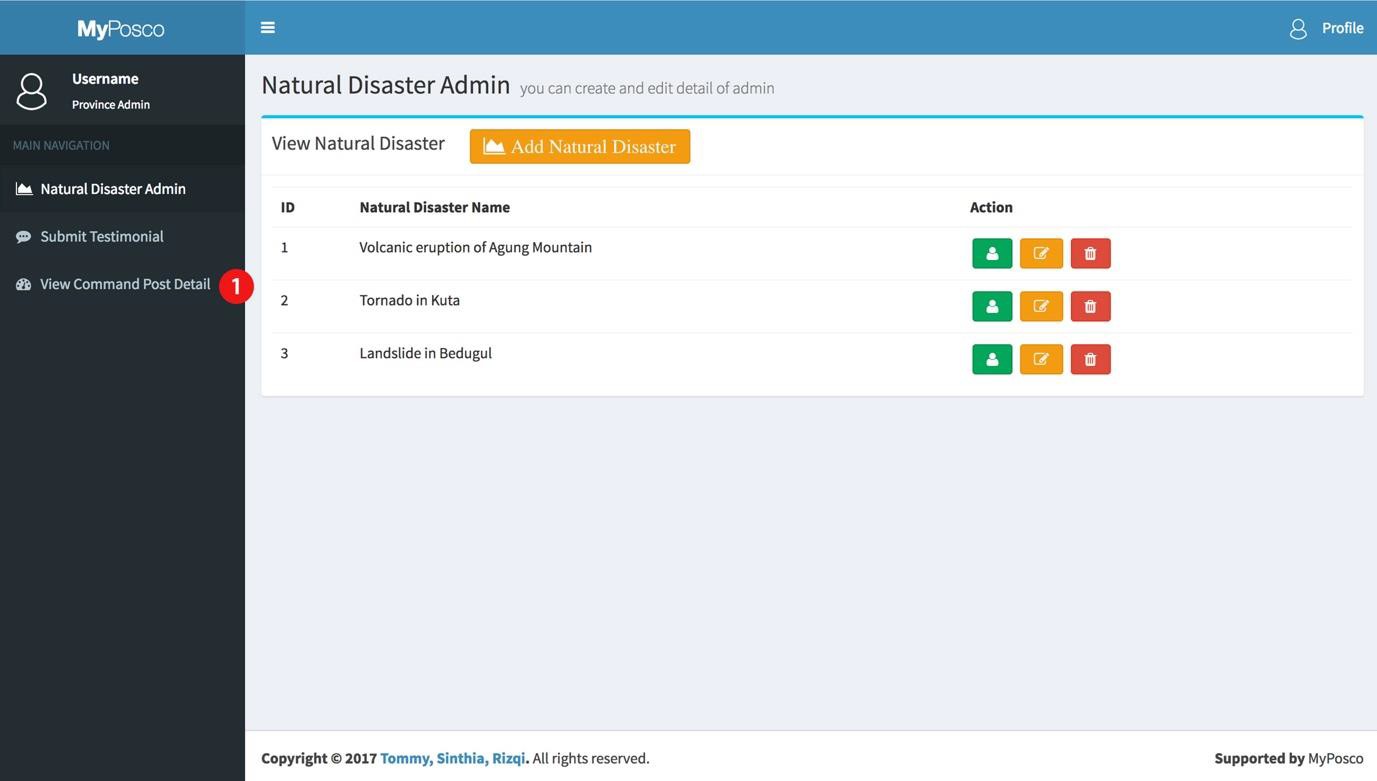
This is navigation which can be clicked and show what content which you want to be managed.

1. Testimonial Field

This is a field for province admin give the testimonial about this web.

1. Submit

The testimonial will submit to database and wait to approval from super admin.



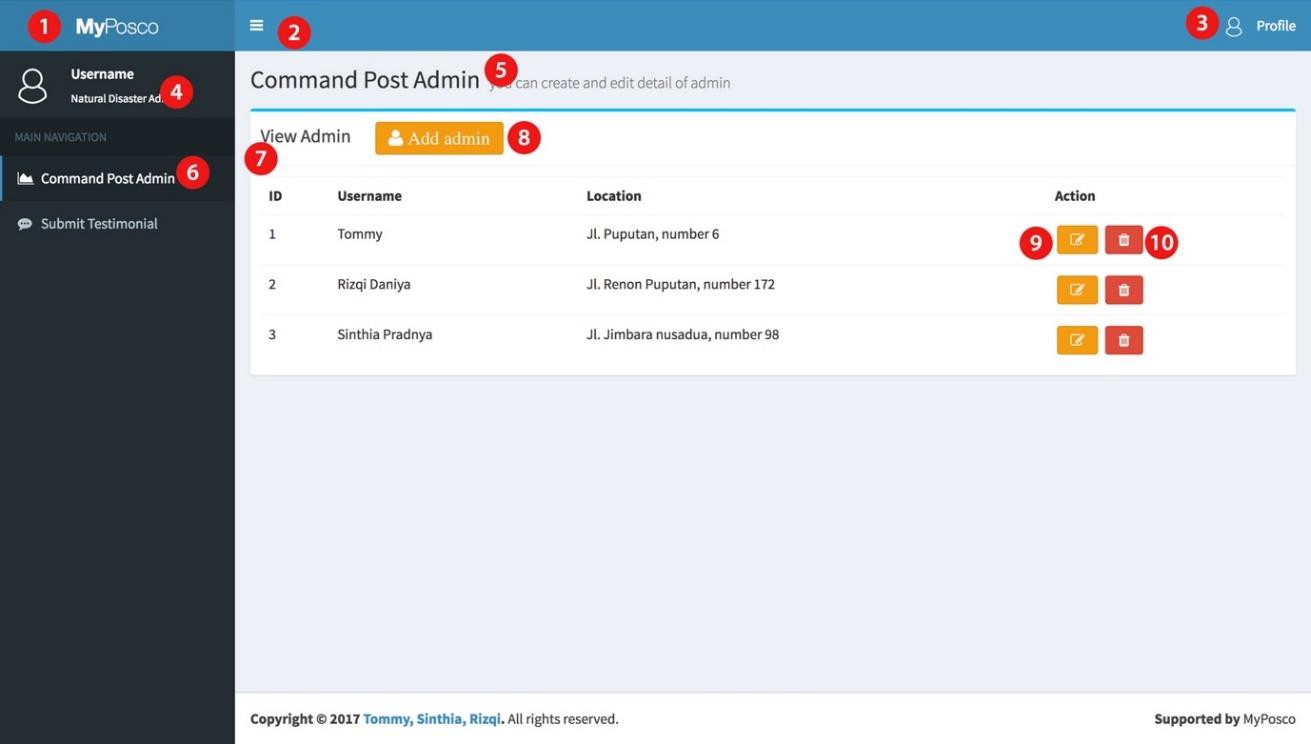
*Figure 3.8.4.4.20 Web UI for Admin: Province Admin Panel (View*

*Command Post Detail)*

Explanation:

1. Command post detail

This button will direct admin to command post detail.



*Figure 3.8.4.4.21 Web UI for Admin: Natural Disaster Admin Panel*

*(View Admin Command Post*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Add admin

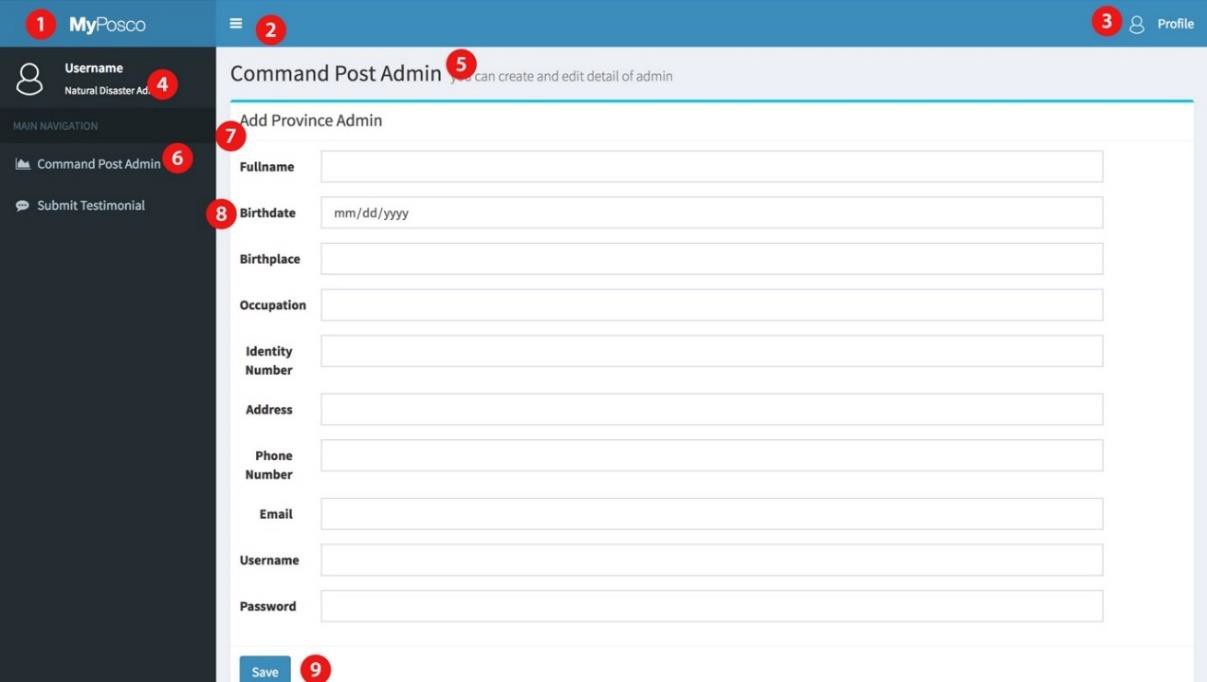
After click add admin button, admin will direct to add admin page.

1. Edit admin

After click edit admin button, admin will direct to edit admin page

1. Delete admin

After click edit admin button, admin will be deleted.



*Figure 3.8.4.4.22 Web UI for Admin: Natural Disaster Admin Panel (Add*

*Admin)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

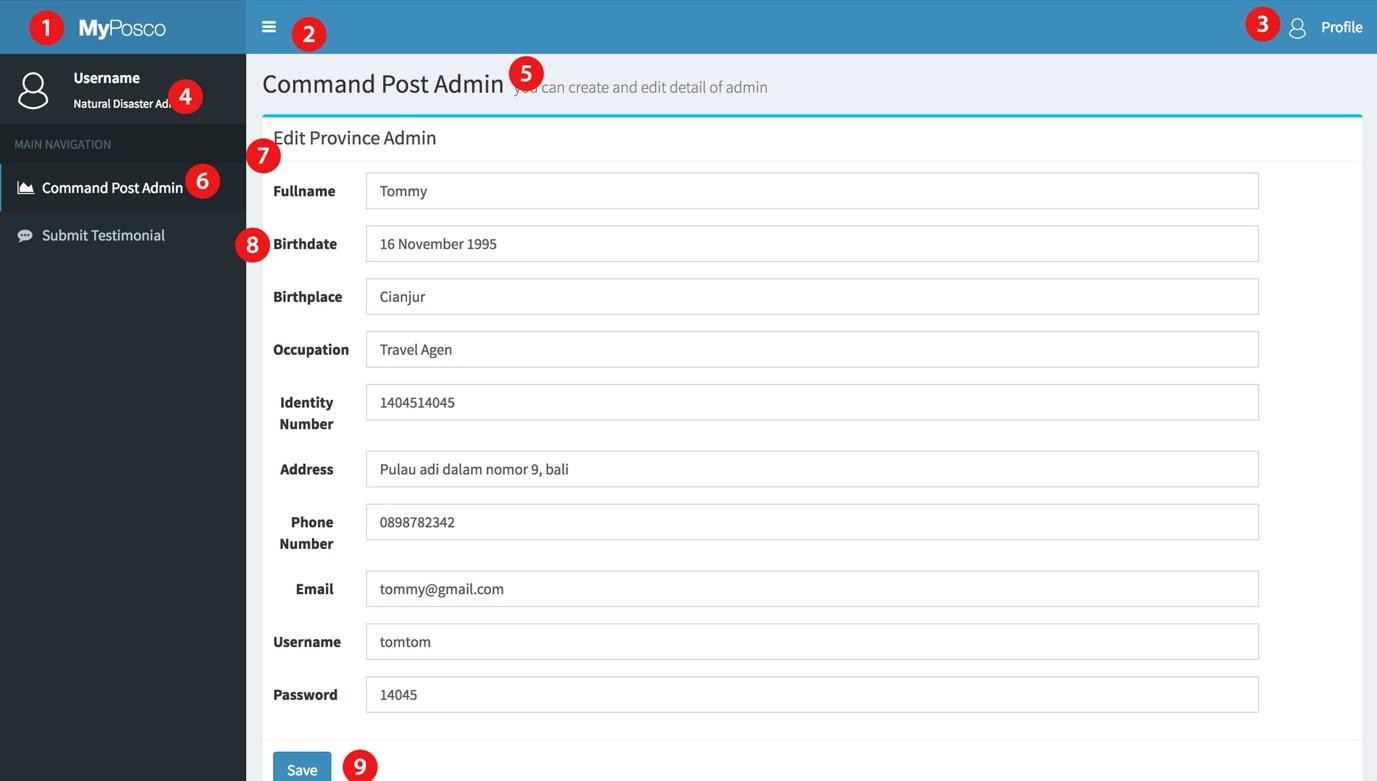
This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Add admin form

There are some field that required to be filled up before adding the admin.

1. Save admin

Click save and admin will be added to view admin.



Explanation:

1. Logo

*Figure 3.8.4.4.23 Web UI for Admin: Natural Disaster Admin Panel (Edit Admin Command Post)*

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

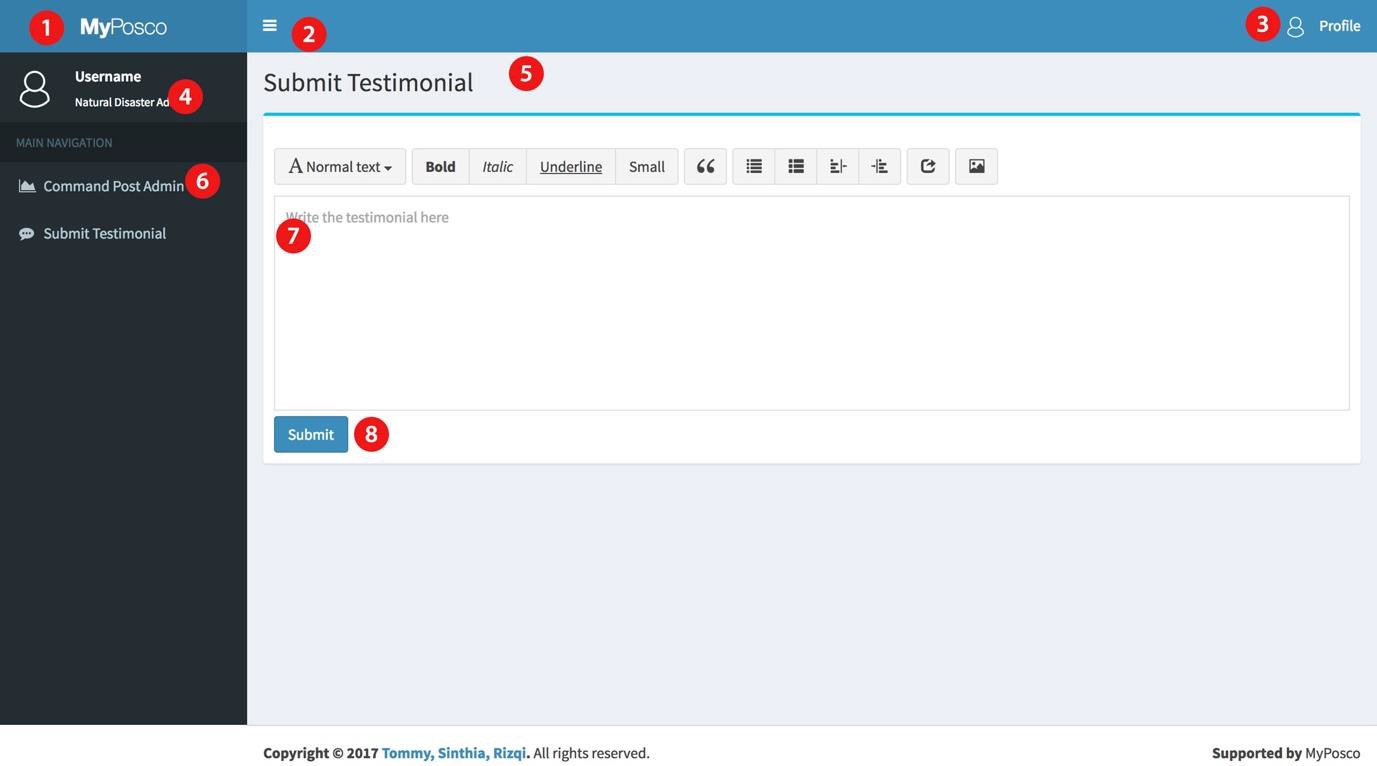
This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Edit admin form

admin can make changes for each admin and edit province admin form will show the last save.

1. Save admin

Click save and admin will update the admin.



*Figure 3.8.4.4.24 Web UI for Admin: Natural Disaster Admin Panel*

*(Testimonial)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

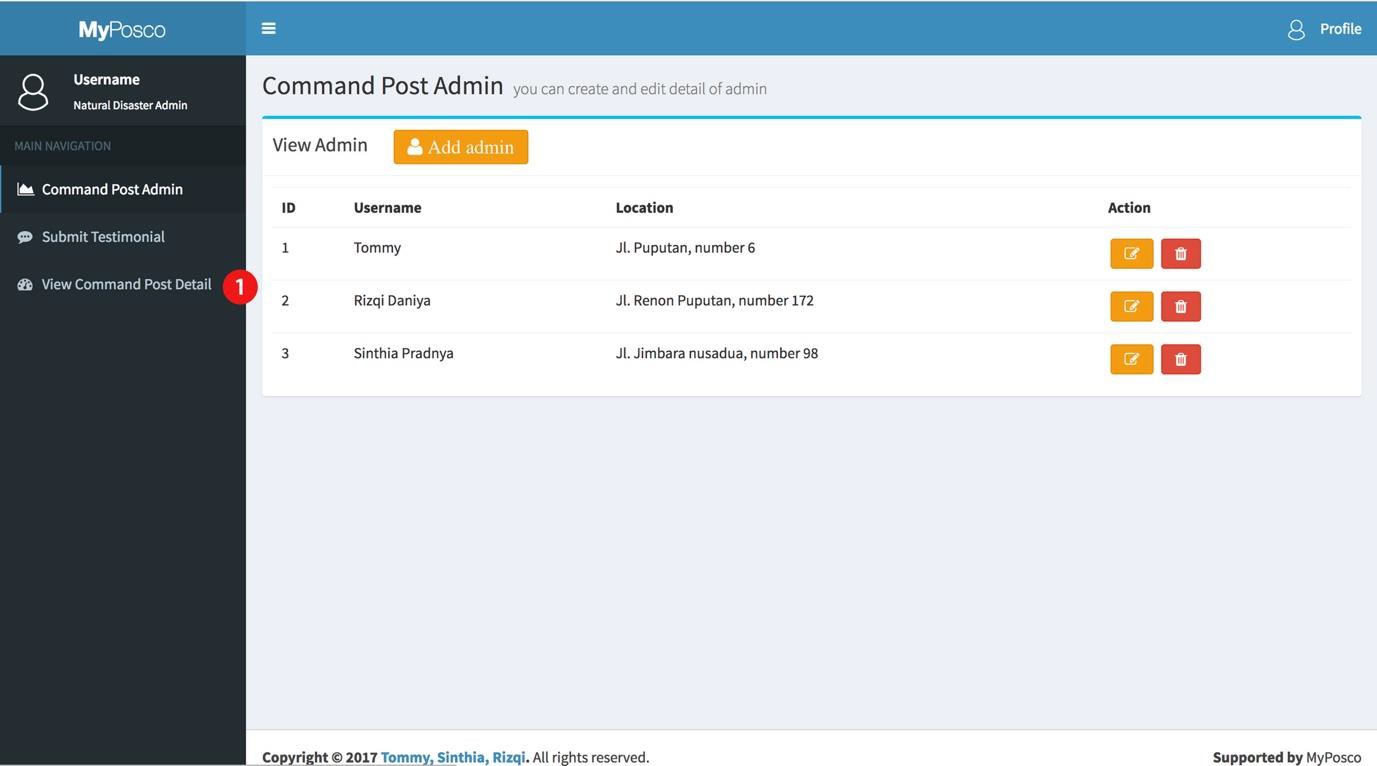
This is navigation which can be clicked and show what content which you want to be managed.

1. Testimonial Field

This is a field for natural disaster admin give the testimonial about this web.

1. Submit

The testimonial will submit to database and wait to approval from super admin.



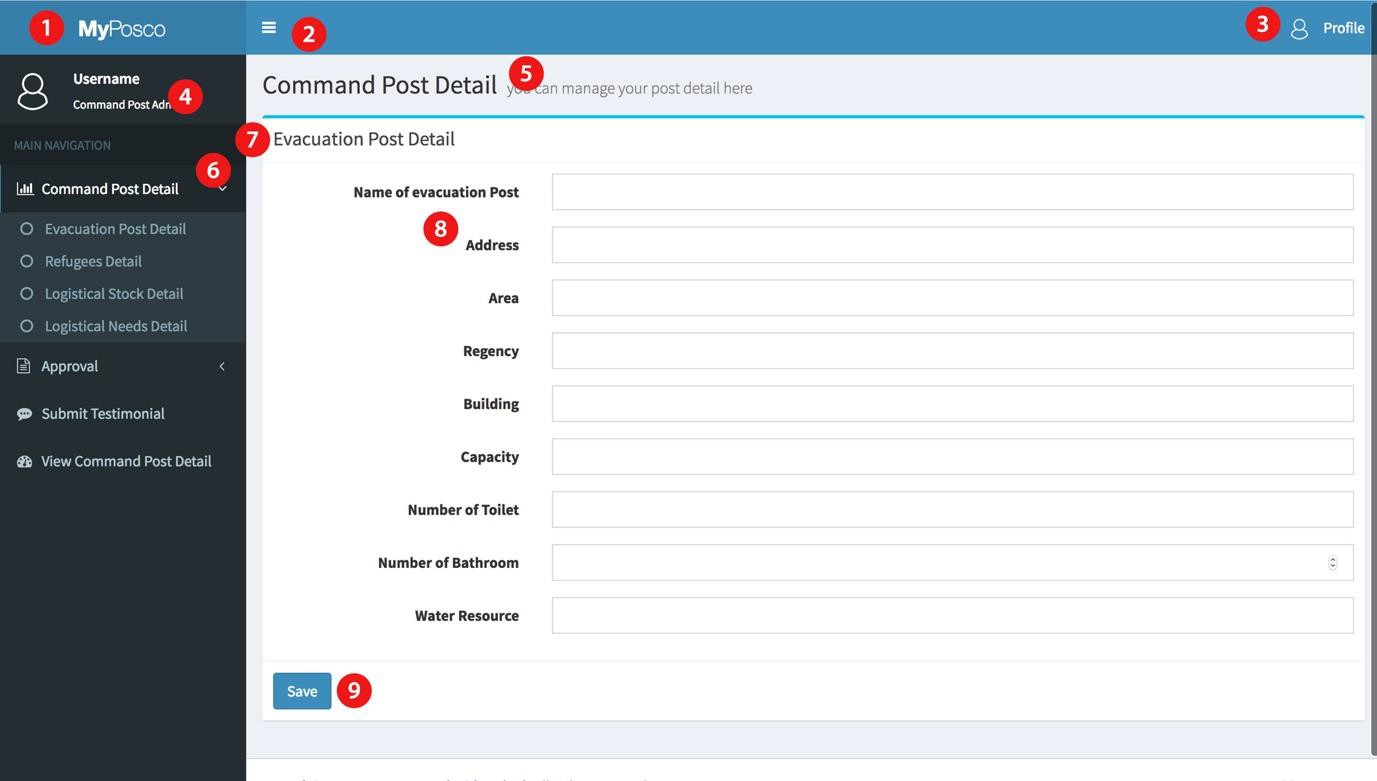
*Figure 3.8.4.4.25 Web UI for Admin: Natural Disaster Admin Panel*

*(View Command Post Detail)*

Explanation:

1. Command post detail

This button will direct admin to command post detail.



*Figure 3.8.4.4.26 Web UI for Admin: Command Post Admin Panel*

*(Evacuation Post Detail)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

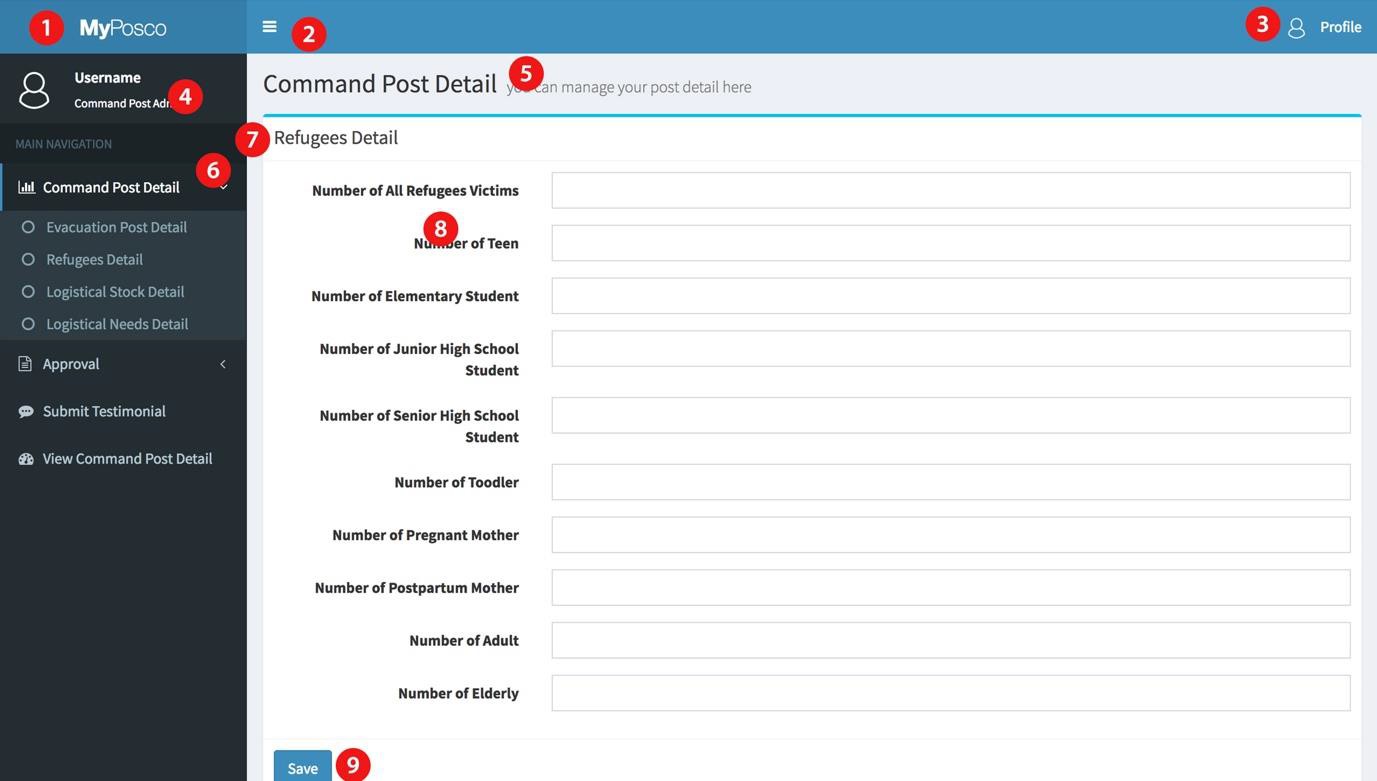
This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Evacuation Post Form

There are some field to input the evacuation Post Detail.

1. Save

This button will process the form and input it into database.



*Figure 3.8.4.4.27 Web UI for Admin: Command Post Admin Panel*

*(Refugees Detail)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

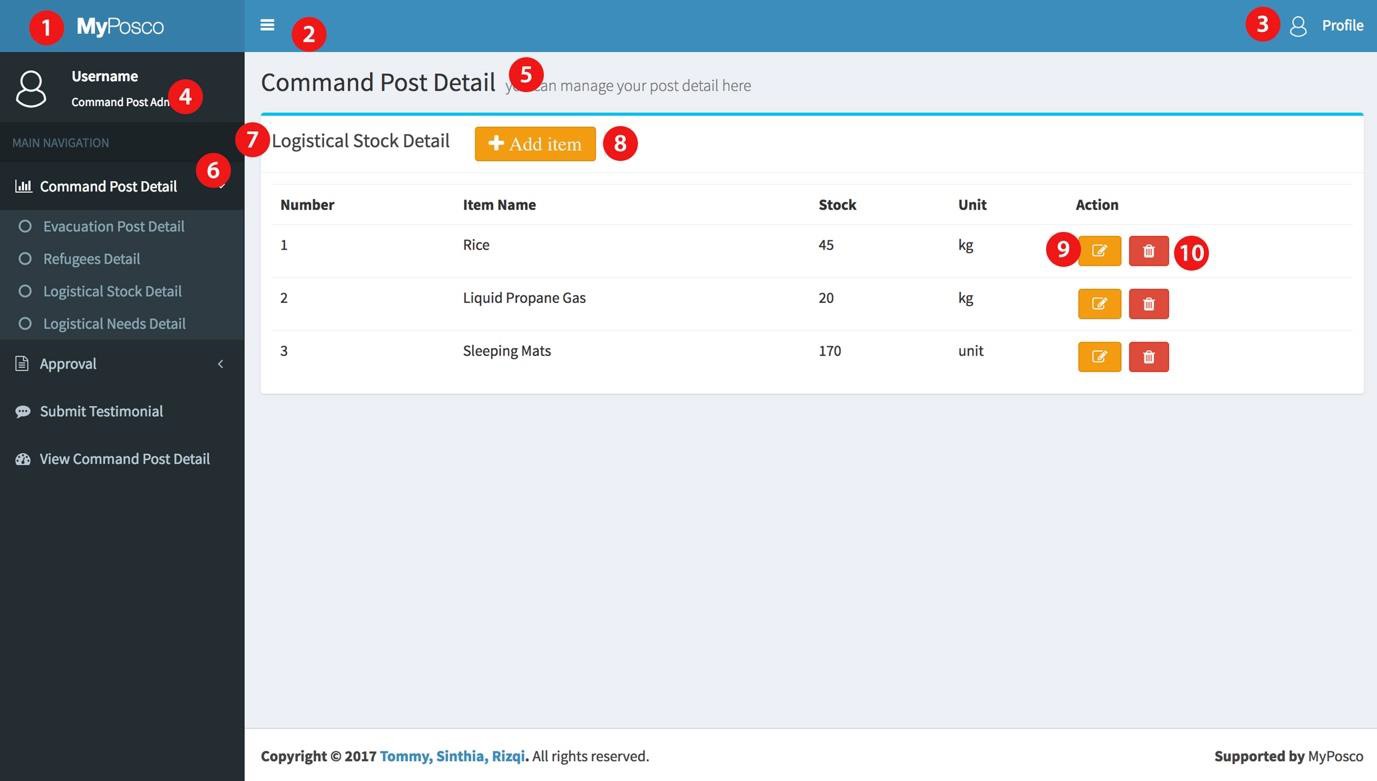
This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Refugees Form

There are some field to input the Refugees Detail.

1. Save

This button will process the form and input it into database.



*Figure 3.8.4.4.28 Web UI for Admin: Command Post Admin Panel*

*(Logistical Aids)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Add item

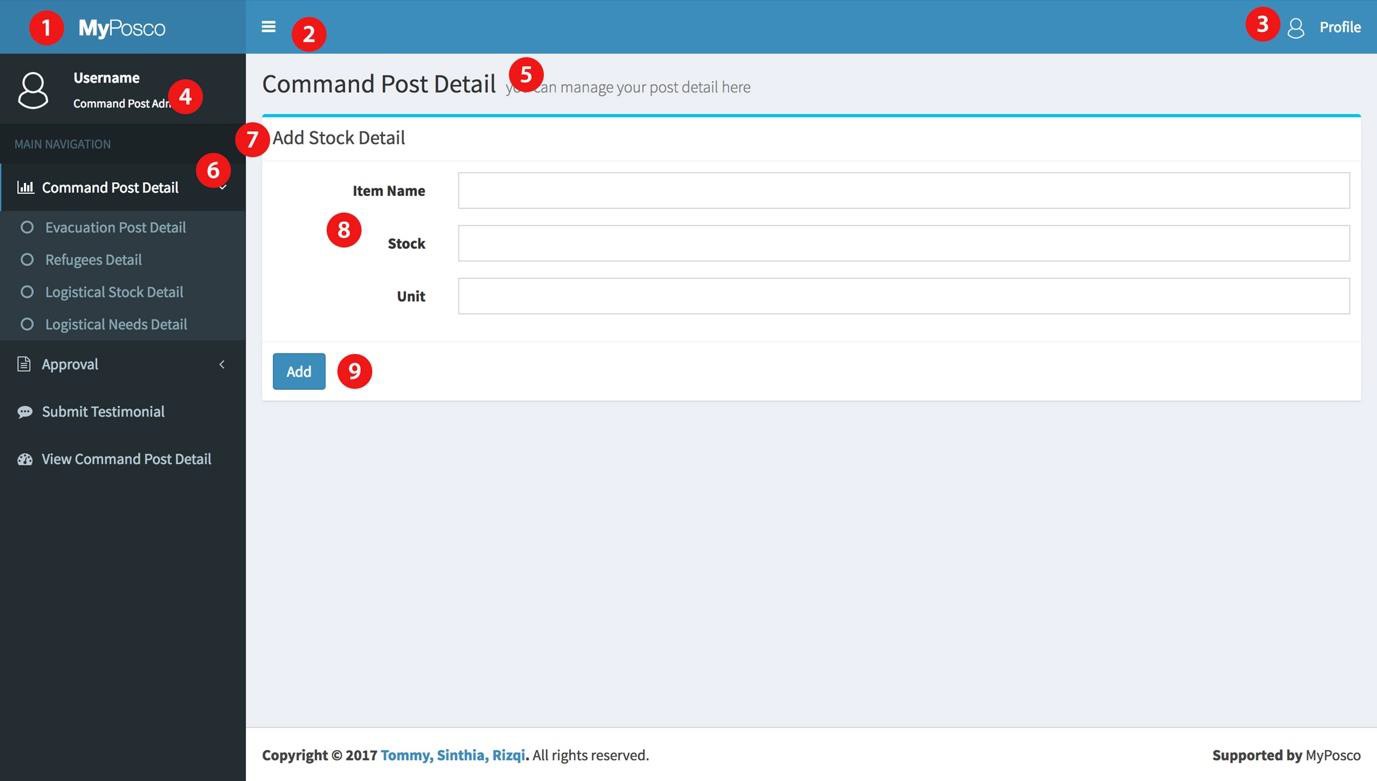
After click add item button, admin will direct to add item page.

1. Edit item

After click edit admin button, admin will direct to edit item page

1. Delete item

After click edit item button, item will be deleted.



*Figure 3.8.4.4.29 Web UI for Admin: Command Post Admin Panel (Logistical*

*Stock Detail – Add Stock)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

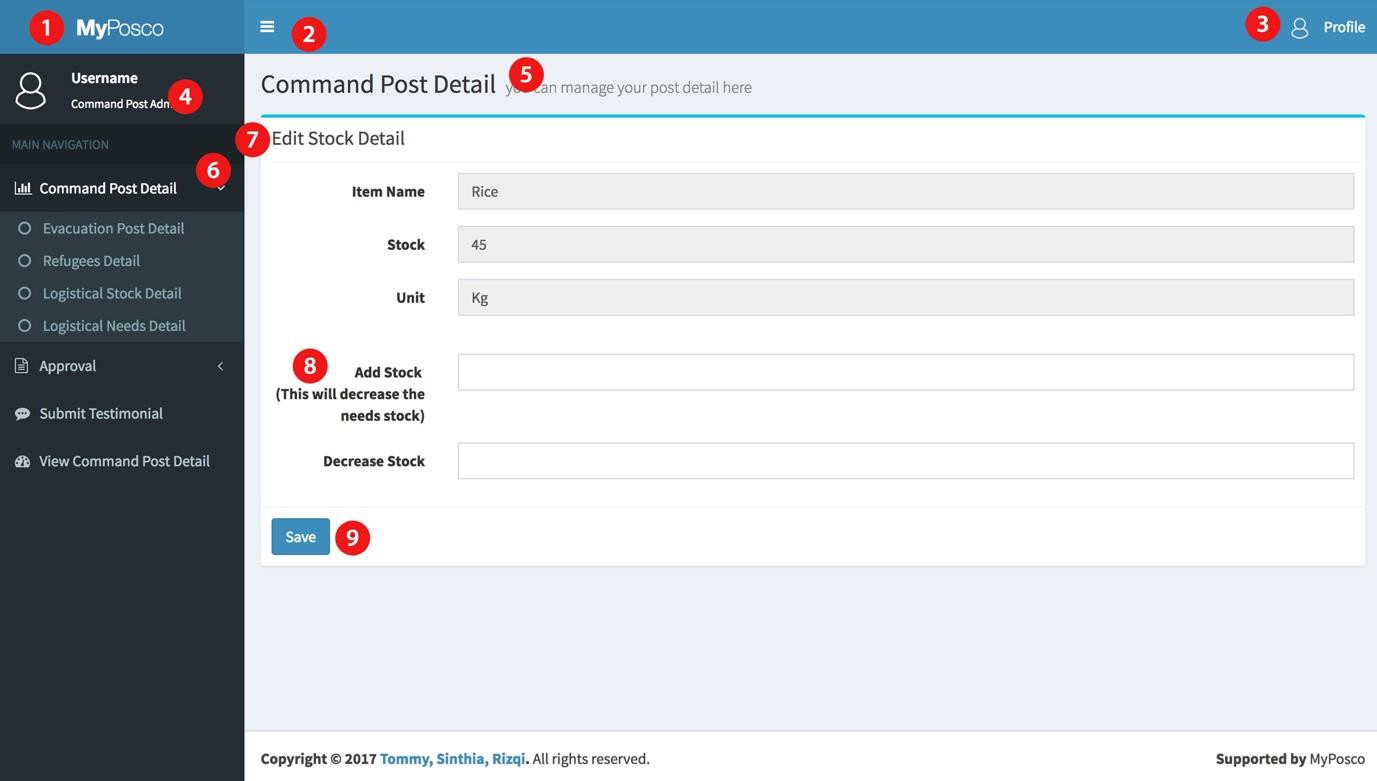
This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Logistical Stock Form

There are some field to input the Logistical Stock Detail.

1. Save

This button will process the form and input it into database.



*Figure 3.8.4.4.30 Web UI for Admin: Command Post Admin Panel (Logistical*

*Stock Detail-Edit Stock)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

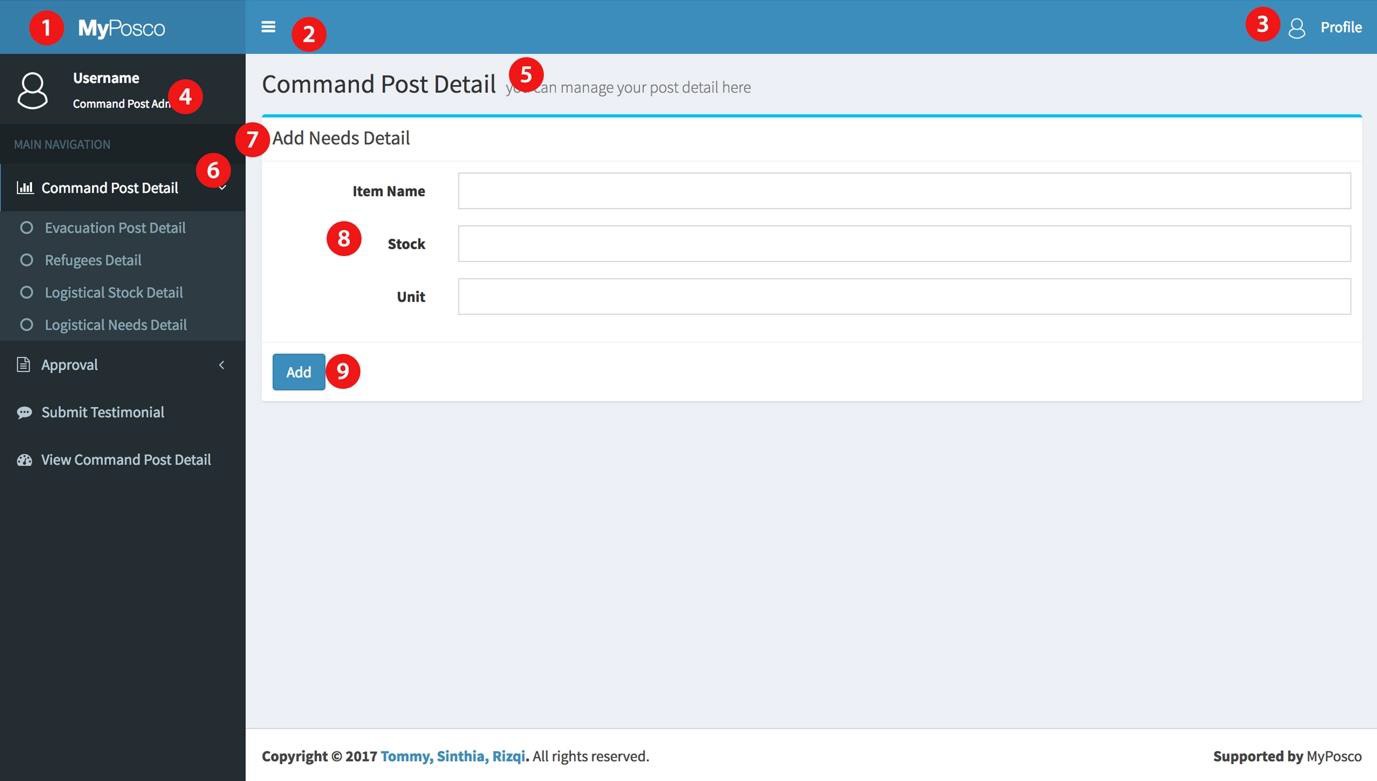
This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Logistical Stock Form

Admin can add stock or decrease stock. If admin add stock, it will decrease the logistical needs.

1. Save

This button will process the form and input it into database.



*Figure 3.8.4.4.31 Web UI for Admin: Command Post Admin Panel*

*(Logistical Needs Detail-Add Item)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. Add item

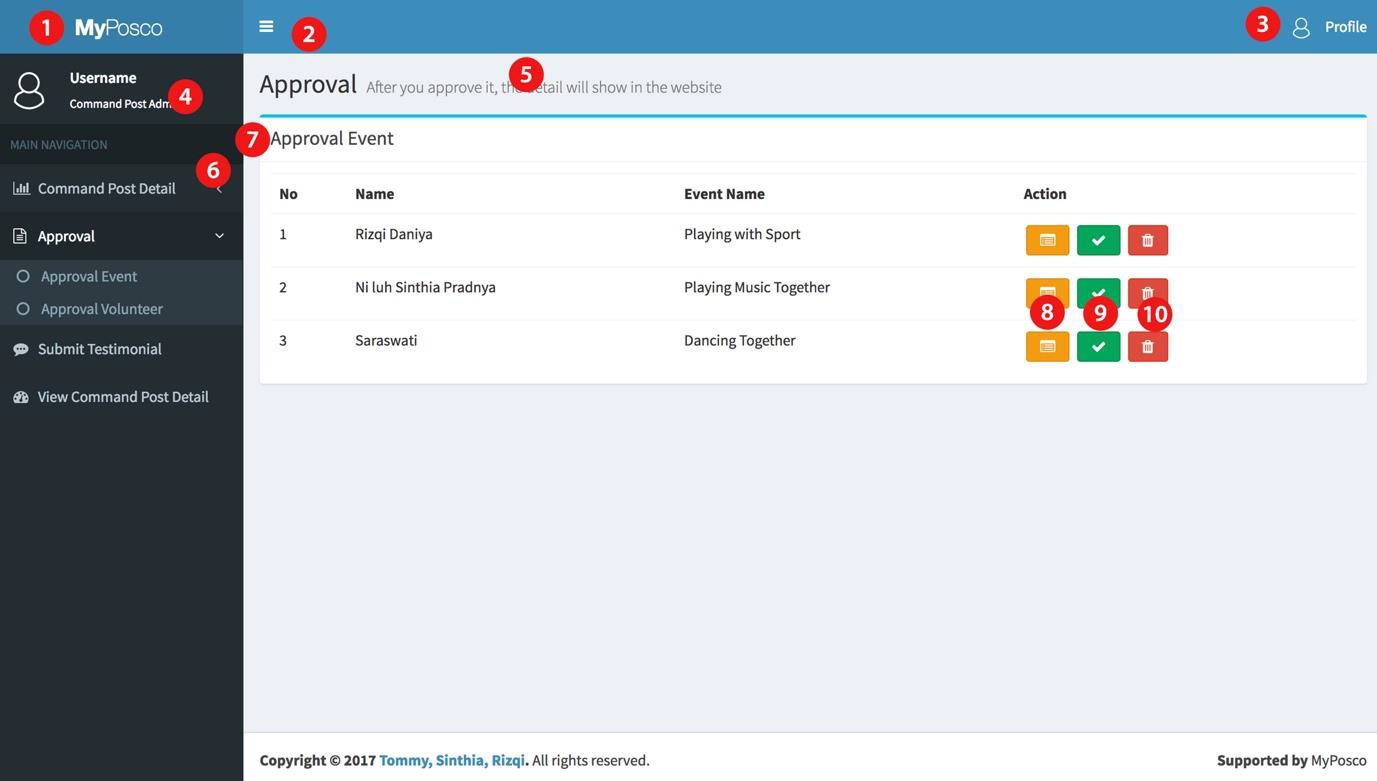
After click add item button, admin will direct to add item page.

1. Edit item

After click edit admin button, admin will direct to edit item page

1. Delete item

After click edit item button, item will be deleted.



*Figure 3.8.4.4.32 Web UI for Admin: Command Post Admin Panel*

*(Approval Event and Volunteer)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

This is navigation which can be clicked and show what content which you want to be managed.

1. Sub-title page

This will show the navigation which is provide by province admin such as view province, view admin, add admin, edit admin, etc.

1. See Detail Event

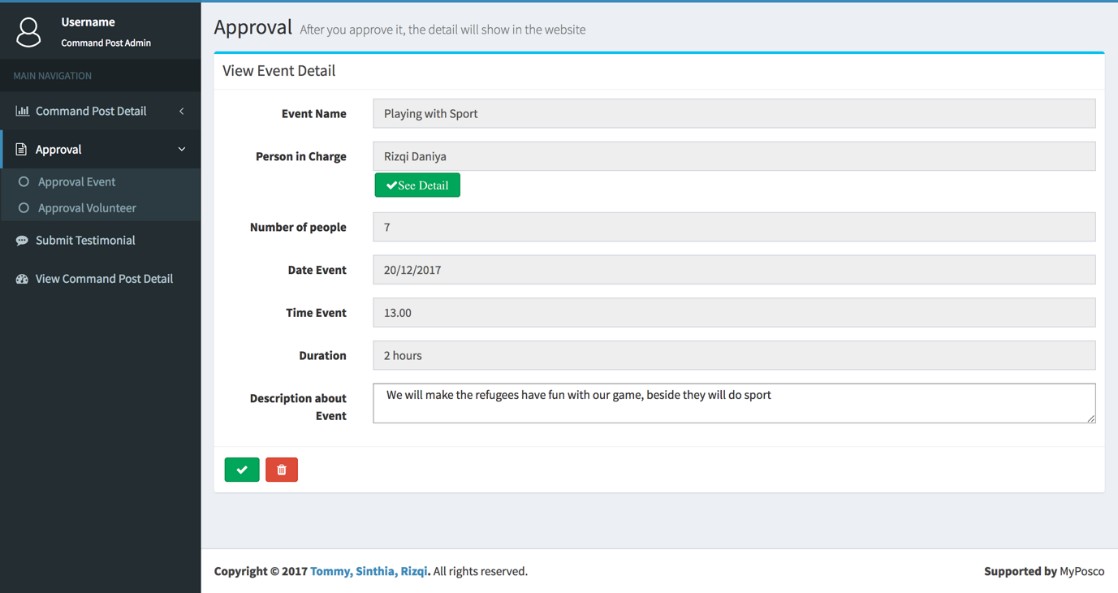
Admin can see the detail of event.

1. Accept Event

By click this button, admin can accept the event or not, if admin accept this event, it will be showed in the webpage.

1. Ignore event

If admin ignore the event, it will be deleted from the event list.

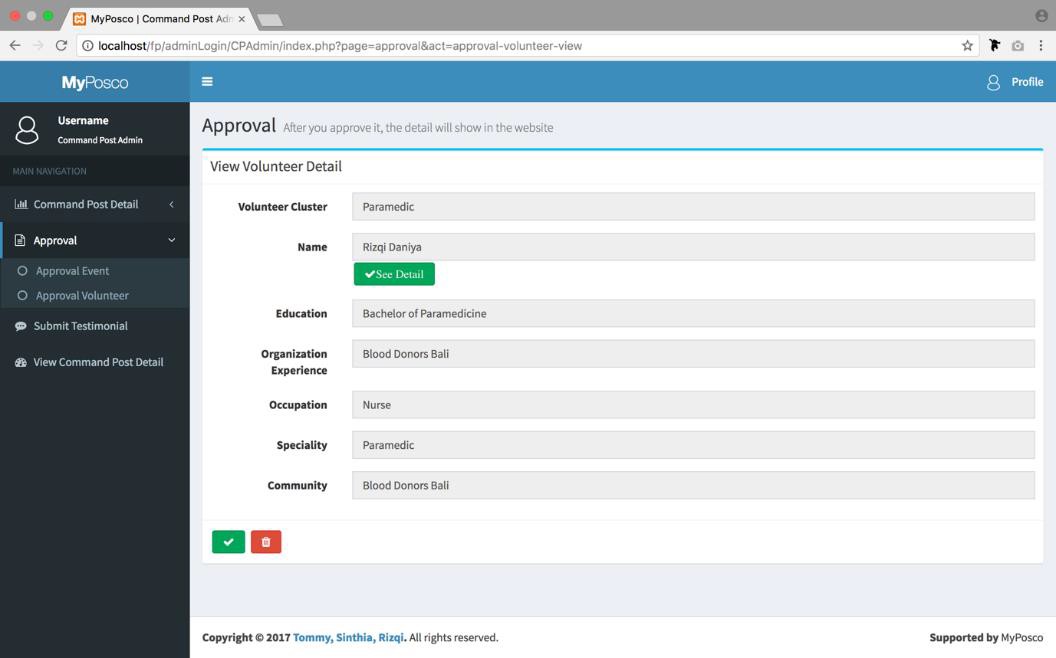


*Figure 3.8.4.4.33 Web UI for Admin: Command Post Admin Panel (See*

*Detail of Event)*

Explanation:

When the admin clicks the see event detail icon, this page will show the information about event. The Admin also can see the detail of person in charge base on registration detail of user.

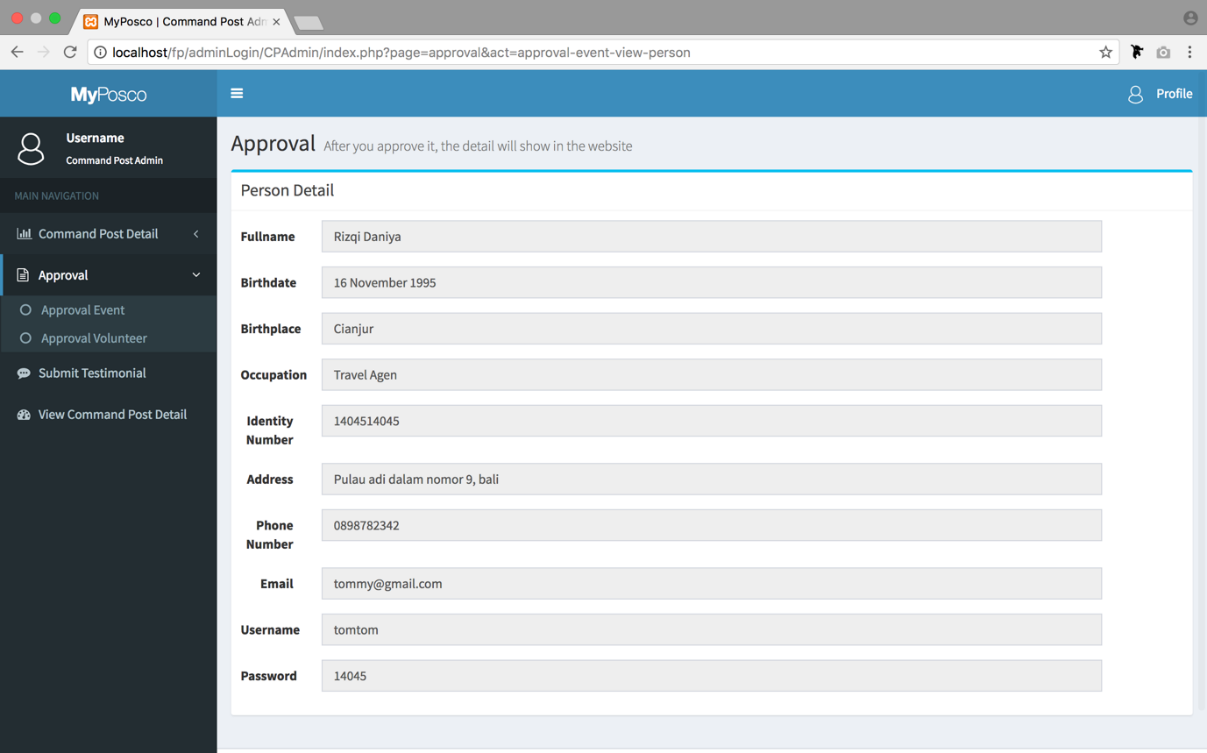


*Figure 3.8.4.4.34 Web UI for Admin: Command Post Admin Panel (See*

*Detail of Volunteer)*

Explanation:

When the admin clicks the see volunteer detail icon, this page will show the information about volunteer. The Admin also can see the detail of person in charge base on registration detail of user.

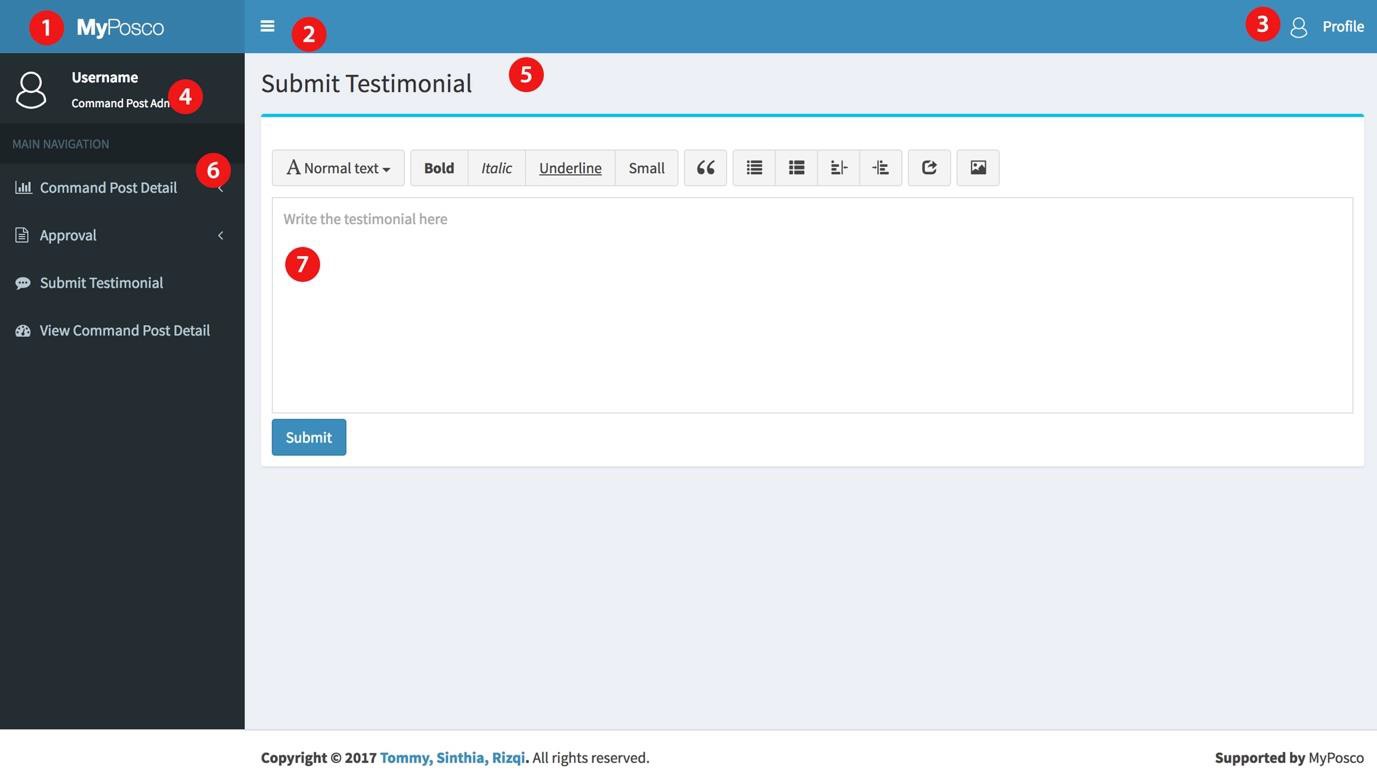


*Figure 3.8.4.4.35 Web UI for Admin: Command Post Admin Panel (Person*

*Detail)*

Explanation:

This page will show after the admin click the see detail button in the approval event or volunteer. This page show about the information about person in charge base on the registration detail of user.



*Figure 3.8.4.4.36 Web UI for Admin: Command Post Admin Panel*

*(Testimonial)*

Explanation:

1. Logo

We can put domain name, name of application or put a logo here.

1. Menu show/hide

We provide the button to hide or show sidebar navigation.

1. Profile button

The admin can look their profile and we provide log out button here.

1. Detail admin

This is detail of admin in form of username and the rule of admin

1. Title page

After the navigation is clicked, the title page will show base on navigation

1. Sidebar navigation

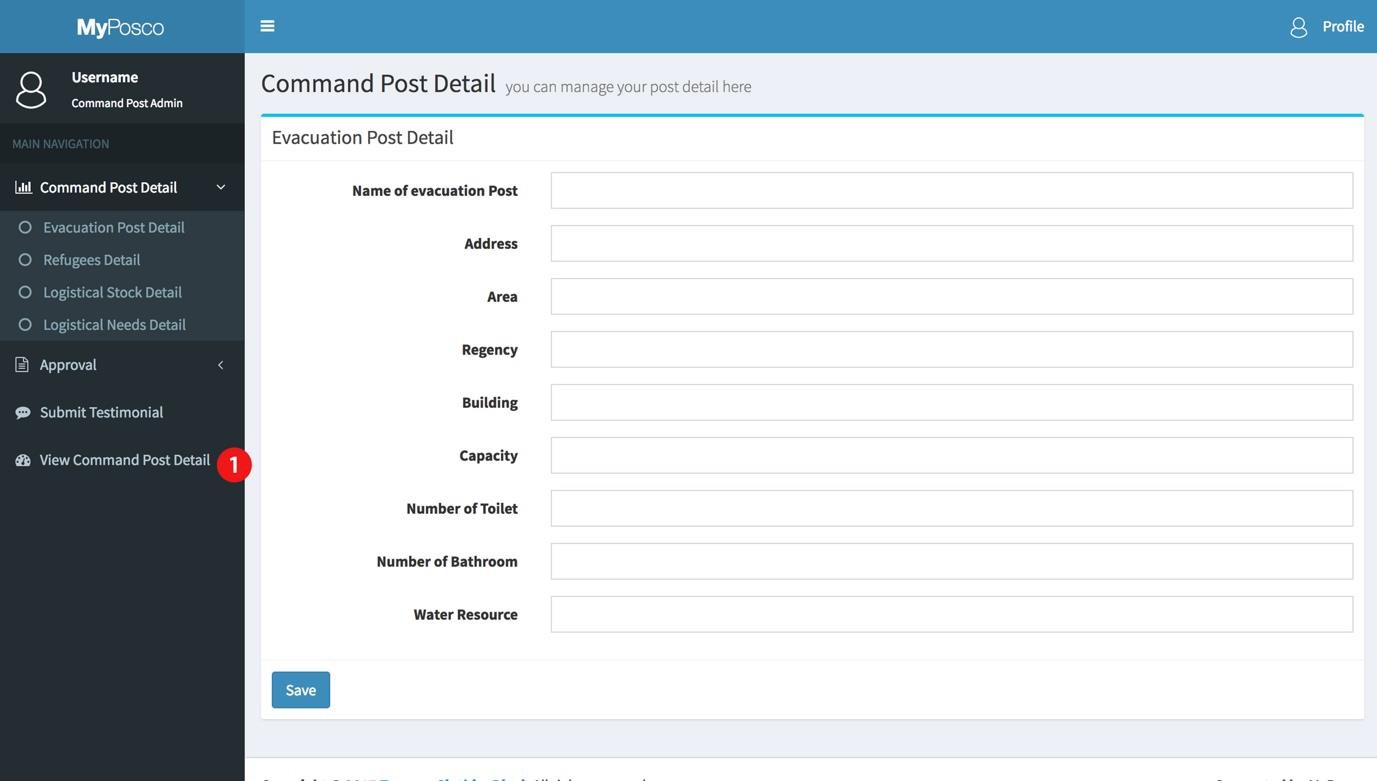
This is navigation which can be clicked and show what content which you want to be managed.

1. Testimonial Field

This is a field for province admin give the testimonial about this web.

1. Submit

The testimonial will submit to database and wait to approval from super admin.



*Figure 3.8.4.4.37 Web UI for Admin: Command Post Admin Panel (View*

*Command Post Detail)*

Explanation:

1. Command post detail

This button will direct admin to command post detail.

### Implementation/Testing

* + 1. **Introduction**

In the development of software, it is needed to do some testing to ensure the quality of software meet the goals that has been set. The other reason is the testing could specifies the strength of software compared to other existing product. For our project, we have done the testing phase of our prototype which is in form of web-based platform. The use case we have tested in our website are login for all division of admin, login and register for user.

### Implementation Methodology

The methodology that we implemented in our software of natural disaster information system is waterfall with prototyping. Waterfall with prototyping allow us to understand how to make the project in structural way and with prototyping methodology it teaches us to learn to develop the project in accordance with user expectation because during the testing phase there are validation and verification we conducted to ensure the quality of software. To create the software project of natural disaster information system, we are using Sublime Text 3 which is the current version of sublime text. Sublime text 3 offer use many easiness in coding the website and also easy to learn because there are many forums that currently use that application in building the web project. Meanwhile, there are still some challenges that that hinder us during the development of project. Lack of knowledge to build big project like what we currently conduct. It is because the application needs many features that assure user especially society to get appropriate information about natural disaster that happen in their area. Then for the accuracy and validity of information we need to include 3 admins who are admin of province, admin of natural disaster and also admin of post. But all problems have finished to be solved because we are able to specify the features that owns by one admin to another.

### Unit Testing

|  |  |
| --- | --- |
| **Unit Test Plan** | |
| Module Name/ID: Register User | Program Name/ID: MyPosco.com |

* + - 1. **Module Overview**

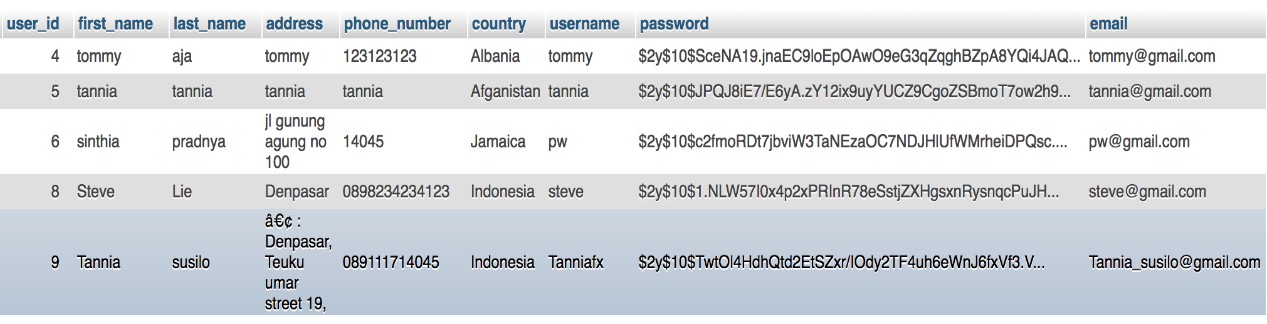
Register a user and save the detail of user to database

|  |
| --- |
|  |
| **Unit Test Plan** |
|  |
| **1.1 Inputs to Module** |
| * First Name (Varchar) * Last Name (Varchar) * Address (Varchar) * Phone Number (Varchar) * Country (Varchar) * Username (Varchar) * Password (Varchar) * Email Address (Varchar) |
|  |
| **1.2 Outputs from Module** |
| User has been registered correctly and save the detail of user base on the input into database. |
|  |
| **1.3 Segment of Code** |
| public function register($firstName, $lastName, $address, $phone, $country, $uname,  $upass, $umail)  {  try  {  $stmt = $this->conn->prepare("SELECT username, email FROM user\_account WHERE username=:uname OR email=:umail");  $stmt->execute(array(':uname'=>$uname, ':umail'=>$umail));  $row=$stmt->fetch(PDO::FETCH\_ASSOC); if($row['username']==$uname) {  return "username is already exist";  }  else if($row['email']==$umail) { return "email is already exist";  }  else{  $new\_password = password\_hash($upass,  PASSWORD\_DEFAULT); |

|  |
| --- |
|  |
| **Unit Test Plan** |
| $stmt = $this->conn->prepare("INSERT INTO  user\_account(first\_name, last\_name, address, phone\_number, country, username,password,email)  VALUES(:first\_name, :last\_name, :address, :phone, :country, :uname, :upass,  :umail)");  $stmt->bindparam(":first\_name", $firstName);  $stmt->bindparam(":last\_name", $lastName);  $stmt->bindparam(":address", $address);  $stmt->bindparam(":phone", $phone);  $stmt->bindparam(":country", $country);  $stmt->bindparam(":uname", $uname);  $stmt->bindparam(":upass", $new\_password);  $stmt->bindparam(":umail", $umail);  $stmt->execute(); return true;  }  }  catch(PDOException $e)  {  echo $e->getMessage();  }  } |
|  |
| **2. Test Data** |
| 1. Valid data    * First Name : Tannia    * Last Name : Susilo    * Address : Denpasar, Teuku umar street 19,    Phone : 089111714045   * + Country : Indonesia   + Username : Tanniafx   + Password : fxtansil   + Email : [Tannia\_susilo@gmail.com](mailto:Tannia_susilo@gmail.com)  1. Invalid data |

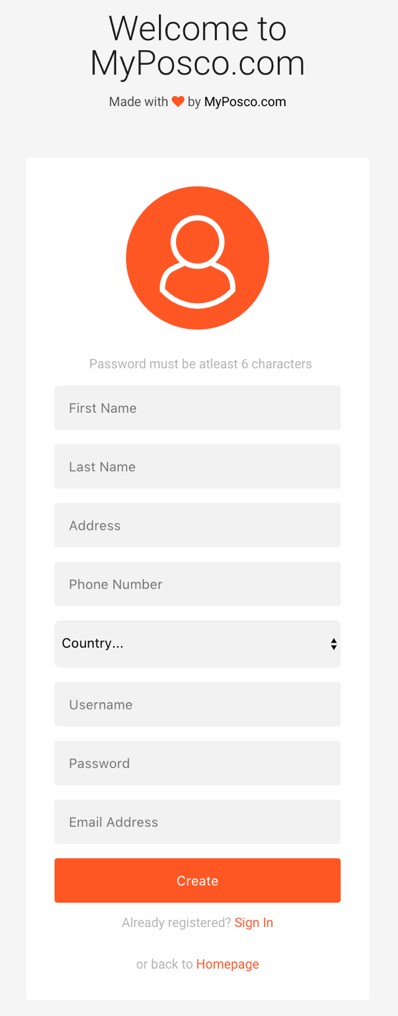
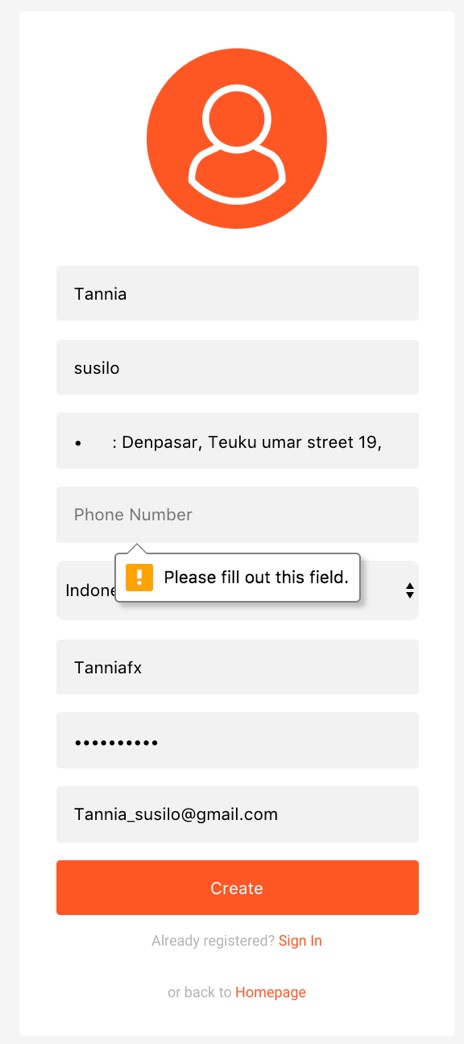
|  |
| --- |
|  |
| **Unit Test Plan** |
| * First Name : 123Karno * Last Name : Tuwodo * Address : Badung, Merpati street 234,    Phone : 08123456234   * Country : Indonesia * Username : 123Karno * Password : a123 * Email : [kardotuwodo@gmail.com](mailto:kardotuwodo@gmail.com) |
| **2.1 Positive Test Cases** |
| * First Name : Tannia * Last Name : Susilo * Address : Denpasar, Teuku umar street 19,    Phone : 089111714045   * Country : Indonesia * Username : Tanniafx * Password : fxtansil * Email : [Tannia\_susilo@gmail.com](mailto:Tannia_susilo@gmail.com)   This is a valid data. When this data is inputted, the data of every field will be inputted into database, and there is a message that said **“Successfully registered, please login here”** |
|  |
| **2.2 Negative Test Cases** |
| * First Name : Tannia * Last Name : Susilo * Address : Denpasar, Teuku umar street 19,    Phone : 089111714045   * Country : Indonesia * Username : Tanniafx * Password : test * Email : [Tannia\_susilo@gmail.com](mailto:Tannia_susilo@gmail.com)   This is uninvalid data. When this data is inputted, there is an error message from the website. Because the password should be at least 6 characters. Every field also will show an error message when there is an unfilled field of this registration form. There also have a validation of email that the email field should contain the right email content. |

|  |
| --- |
|  |
| **Unit Test Plan** |
|  |
| **3. Interface Modules** |
| **Positive Test Case**    **Data inputted into database** |



**Unit Test Plan**

**Negative Test Case**



**4. Test Tools**

This system was tested in Chrome, Safari and Mozilla Firefox.

*Table 3.9.3.1 Unit Testing: Register User*

|  |  |
| --- | --- |
| **Unit Test Plan** | |
| Module Name/ID: Login User | Program Name/ID: MyPosco.com |

1. **Module Overview**

Login user base on data in database.

* 1. **Inputs to Module**
     + Username (Varchar)
     + Password (Varchar)
  2. **Outputs from Module**

User can access the logged page of user.

* 1. **Logic Flow Diagram/Segment of Code**

public function doLogin($uname,$upass)

{

try

{

$stmt = $this->conn->prepare("SELECT user\_id, username, email,

password FROM user\_account WHERE username=:uname OR email=:umail ");

$stmt->execute(array(':uname'=>$uname, ':umail'=>$uname));

$userRow=$stmt->fetch(PDO::FETCH\_ASSOC); if($stmt->rowCount() == 1)

{

if(password\_verify($upass, $userRow['password']))

{

}

else

{

}

}

}

$\_SESSION['user\_session'] = $userRow['user\_id']; return true;

return false;

catch(PDOException $e)

{

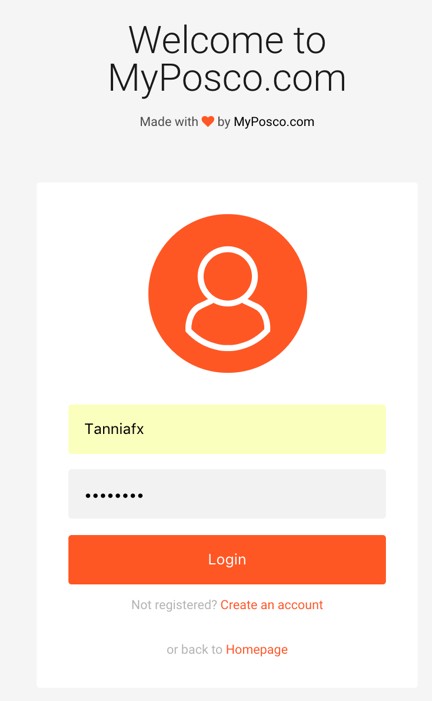
echo $e->getMessage();

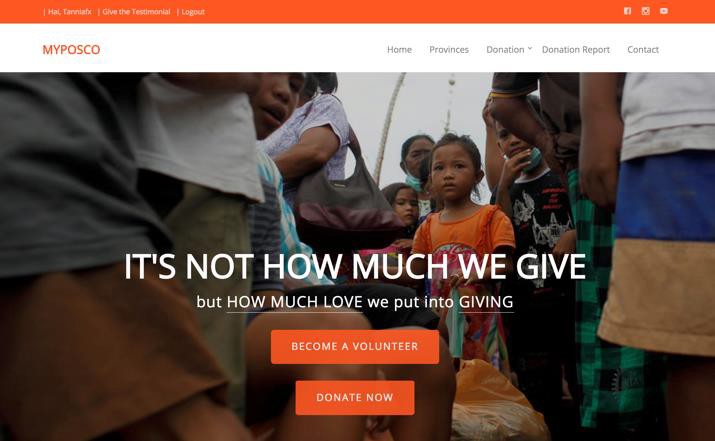
|  |
| --- |
|  |
| **Unit Test Plan** |
| }  } |
|  |
| **2. Test Data** |
| 1. Invalid    * Username : tanniafx    * Password : fxtansil 2. Invalid data    * Username : ariniwati    * Password : testedpassword |
|  |
| **2.1 Positive Test Cases** |
| * Username : tanniafx * Password : fxtansil   This is a valid data. When this data is inputted, the user logged page will be showed. |
|  |
| **2.2 Negative Test Cases** |
| * Username : ariniwati * Password : testedpassword   This is invalid data. This data is doesn’t match with database. When this data is inputted, there is an error message from the form. |

**Unit Test Plan**

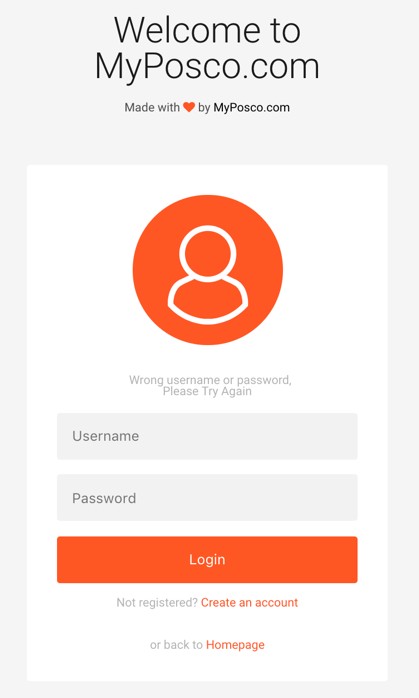
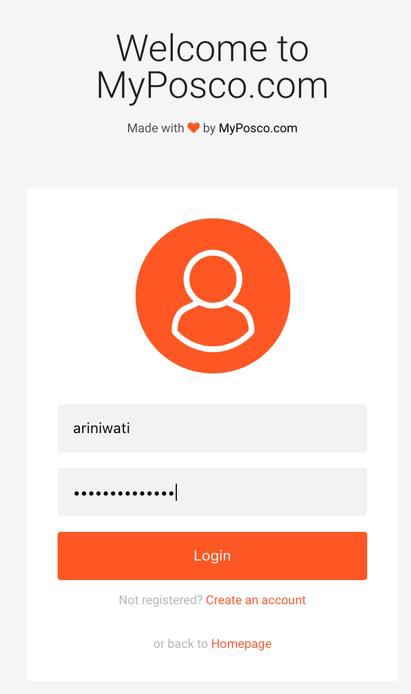
1. **Interface Modules**

**Positive test case**





**Negative test case**



1. **Test Tools**

This system was tested in Chrome, Safari and Mozilla Firefox.

Table 3.9.3.2 *Unit Testing: Login User*

|  |  |
| --- | --- |
| **Unit Test Plan** | |
| Module Name/ID: Login Admin | Program Name/ID: MyPosco.com |

1. **Module Overview**

Login user

* 1. **Inputs to Module**
     + Username (Varchar)
     + Password (Varchar)
  2. **Outputs from Module**

Login can access the logged page of user.

* 1. **Logic Flow Diagram/Segment of Code**

public function doLogin($uname,$upass)

{

try

{

$stmt = $this->conn->prepare("SELECT id, username, email,

password, admin\_category FROM admin WHERE username=:uname OR email=:umail ");

$stmt->execute(array(':uname'=>$uname, ':umail'=>$uname));

$userRow=$stmt->fetch(PDO::FETCH\_ASSOC); if($stmt->rowCount() == 1)

{

if($upass == $userRow['password'])

{

admin")

admin")

$\_SESSION['user\_session'] = $userRow['id']; if($userRow['admin\_category']=="super admin")

return 1;

else if($userRow['admin\_category']=="province

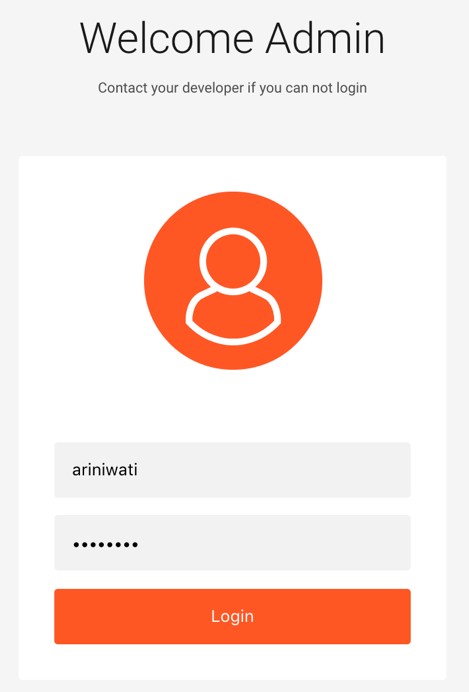
return 2;

else if($userRow['admin\_category']=="disaster

return 3;

else if($userRow['admin\_category']=="post admin") return 4;

|  |
| --- |
|  |
| **Unit Test Plan** |
| else  return false;  }  else  {  return false;  }  }  }  catch(PDOException $e)  {  echo $e->getMessage();  }  } |
|  |
| **2. Test Data** |
| 1. Invalid    * Username : super\_admin    * Password : superadmin 2. Invalid data    * Username : ariniwati    * Password : testedpassword |
|  |
| **2.1 Positive Test Cases** |
| * Username : super\_admin * Password : superadmin   This is a valid data. When this data is inputted, the admin logged page will be showed. |
|  |
| **2.2 Negative Test Cases** |
| * Username : ariniwati * Password : testedpassword   This is invalid data. This data is doesn’t match with database. When this data is inputted, there is an error message from the form. |



|  |
| --- |
|  |
| **Unit Test Plan** |
| **3. Interface Modules** |
| **Positive test case**    **Negative test case** |
|  |
| **4. Test Tools** |
| This system was tested in Chrome, Safari and Mozilla Firefox. |
| Table 3.9.3.3 *Unit Testing: Login Admin* |

### System Testing

The use case that we have finished and tested in our prototype are:

Use Case 1: Login Admin

* + - 1. Super Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case Tested:** | | Log in Super Admin | | |
| **Test Description:** | | Testing the super admin can login to the system | | |
| **Pre-conditions** | | Super admin must have a valid username and password | | |
| **Post-conditions** | | Super admin has successfully login and system display the main page of website. | | |
| **Notes:** | |  | | |
| **Result(Pass/Fail/warning/ Incomplete)** | | Pass | | |
| **No.** | **Test Step** | **Expected Test Result** | **Pass** | **Fail** |
| 1. | Login | Super admin open the form of super admin login page. | √ |  |
| 2. | Super admin enter their username and password | Super admin has successfully login to the system. | √ |  |
| 3. | If super admin enter wrong username and password | The system display the message of wrong username or password | √ |  |

*Table 3.9.4.1 Website Platform: Super Admin Login*

* + - 1. Admin of Province

|  |  |
| --- | --- |
| **Use Case Tested:** | Login of Admin of Province |
| **Test Description:** | Testing admin of province can login into the system |
| **Pre-conditions** | Admin of province must have a valid username and password |
| **Post-conditions** | Admin of province has successfully login and system display the main page of website |
| **Notes:** |  |
| **Result(Pass/Fail/warning/** | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Incomplete)** | |  | | |
| **No.** | **Test Step** | **Expected Test Result** | **Pass** | **Fail** |
| 1 | Login | Admin of province open the form of login page | √ |  |
| 2 | Admin of province enter their username and password | Admin of province has successfully login to the system | √ |  |
| 3 | If admin of province enter the wrong username and password | The system display the message of wrong username or password. | √ |  |

*Table 3.9.4.2 Website Platform: Admin of Province Login*

* + - 1. Admin of Natural Disaster

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case Tested:** | | Login of Admin of Natural Disaster | | |
| **Test Description:** | | Testing admin of natural disaster can login into the system | | |
| **Pre-conditions** | | Admin of natural disaster must have a valid username and password | | |
| **Post-conditions** | | Admin of natural disaster has successfully login and system display the main page of website | | |
| **Notes:** | |  | | |
| **Result(Pass/Fail/warning/ Incomplete)** | | Pass | | |
| **No.** | **Test Step** | **Expected Test Result** | **Pass** | **Fail** |
| 1. | Login | Admin of natural disaster open the form of login page | √ |  |
| 2. | Admin of natural disaster enter their username and password | Admin of natural disaster has successfully login to the system | √ |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3. | If admin of natural disaster enter wrong username and password | The system display the message of wrong username and password | √ |  |

*Table 3.9.4.3 Website Platform: Admin of Natural Disaster Login*

* + - 1. Admin of Command Post

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case Tested:** | | Login of Admin Of Command Post | | |
| **Test Description:** | | Testing admin of command post can login into the system | | |
| **Pre-conditions** | | Admin of command post must have a valid username and password | | |
| **Post-conditions** | | Admin of command post has successfully login and system display the main page of website | | |
| **Notes:** | |  | | |
| **Result(Pass/Fail/warning/ Incomplete)** | | Pass | | |
| **No.** | **Test Step** | **Expected Test Result** | **Pass** | **Fail** |
| 1. | Login | Admin of command post open the form of login page | √ |  |
| 2. | Admin of command post enter their username and password | Admin of command post has successfully login to the system | √ |  |
| 3. | If admin of command post enter wrong username and password | The system display the message of wrong username of password | √ |  |

*Table 3.9.4.4 Website Platform: Admin of Command Post Login*

Use Case 2: Register User

|  |  |
| --- | --- |
| **Use Case Tested:** | Register User |
| **Test Description:** | The user can register to be registered users in this web application |
| **Pre-conditions** | The user must have fulfilled the form of user register |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Post-conditions** | | The user can register as a registered user in system and system can access all information related to natural disaster in the website. | | |
| **Notes:** | |  | | |
| **Result(Pass/Fail/warning/ Incomplete)** | | Pass | | |
| **No.** | **Test Step** | **Expected Test Result** | **Pass** | **Fail** |
| 1. | Register | User open the register form page | √ |  |
| 2. | User complete the forms with the appropriate data | The data saved to database and system display the identity number to user and will be directed to user’s page | √ |  |
| 3. | If user does not fulfill the form register completely | System display the warning message for user to input all their identity data in form register. | √ |  |

*Table 3.9.4.5 Website Platform: Register User*

Use Case 3: Login User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case Tested:** | | Login User | | |
| **Test Description:** | | Testing user can login into the system | | |
| **Pre-conditions** | | The user must have a valid username and password | | |
| **Post-conditions** | | The user has successfully login and display the main page of website | | |
| **Notes:** | |  | | |
| **Result(Pass/Fail/warning/ Incomplete)** | | Pass | | |
| **No.** | **Test Step** | **Expected Test Result** | **Pass** | **Fail** |
| 1. | Login | The user open the form of login page | √ |  |
| 2. | User enter their username and password | The user has successfully login to the system and access all | √ |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | information related to natural disaster |  |  |
| 3. | If user enter wrong username and password | System display the message of wrong username or password. | √ |  |

### Testing Phase Conclusion

*Table 3.9.4.6 Website Platform: Login User*

The importance of prototype testing is to ensure there is no possible error or debug happen and the code flow which is implemented can work properly. For the methodology we implement to our project is waterfall with prototyping. The reason is that the methodology teach us to build the project in structural approach and it also allow us to do some validation and verification to our prototype with the user during the testing phase. And at this stage of testing we have done to implement three use cases of login admin, register user and login user.

### 3.10 References

Sharma, A. (2017). *Create Login and Registration Form in PHP Using OOP*. Retrieved from c-sharpcorner: http://www.c- sharpcorner.com/UploadFile/4ab77c/create-login-and-registration-form-in- php-by-using-oops-conc/

informasitips.com. (2017). *informasitips*. Retrieved from Apa Arti Warna Orange?:<http://informasitips.com/apa-arti-warna-orange>

Sharma, A. (2017). *Create Login and Registration Form in PHP Using OOP*. Retrieved from c-sharpcorner.com: http://www.c- sharpcorner.com/UploadFile/4ab77c/create-login-and-registration-form-in- php-by-using-oops-conc/

stackoverflow.com. (2017). *PHP form submission using oop* . Retrieved from stackoverflow.com: https://stackoverflow.com/questions/11573478/php- form-submission-using-oop

Faliskie, D. (2017). *Using HTML form input with OOP in PHP*. Retrieved from restaurant belum punya nama: https://teamtreehouse.com

Faliskie, D. (2017). *Using HTML form input with OOP in PHP*. Retrieved from teamtreehouse.com: https://teamtreehouse.com/community/using-html-form- input-with-oop-in-php

## Chapter 4: Conclusion

### Introduction

The project we conducted for final project is natural disaster information system that implement the technology of web application. Natural disaster information system has a purpose to assist and provide information about the command post of current natural disaster which is happening in Indonesia. This web application is intended to people who wants to contribute themselves as volunteer or give aids in form of money, logistical aid or event. During three months of project development we have faced many challenges and also many achievement of several task within the project. There are 38 use cases that we proposed for our project development. In addition all four objectives, analysis artifact, design artifact, and three use cases of prototype has been defined and tested perfectly. In this chapter of project evaluation mainly discussed about shortcoming and solution, what thing that has been done in right or wrong way, lesson learned during the project development and along with further improvement and future work of our project.

### Shortcoming and Solutions

* + 1. Little Knowledge of Information System Development

In developing the project, each of us still has a little knowledge and experience in building the information system, even for the natural disaster related. At first we are still confuse to decide what functional requirement that could pursue our project objectives and aims. But after taking numerous consultation with our mentor and people who are in charge of recovering natural disaster, one by one functionality and use cases proposed for our project can be defined well. Not only by taking a direct discussion but also doing some research in internet for some journal or books also supported us in completing this project document and application.

* + 1. Google API operation

In our natural disaster information system, we provide the navigation functionality to user to find the area of command post which still need reinforcement. The navigation can be shown by using Google API technology. The problem is that Google API is a new thing to us and the implementation has never been learned by us in the college.

Therefore to overcome it, we do some research in internet especially tutorial video or books of Google API application in our website.

* + 1. Communication

During the project development, communication between teams is beneficial to overcome the clarity of project. It is also used to ensure the quality of project in accordance with our mutual agreement and of course the project goal. Sometimes many uncertainty meaning arises caused of different opinion to one another. For example in determining the functions and features to be included in our project, interface design for admin and user, developing the use case and sequence diagram, which still make us feel doubt whether we already make it right or wrong. Therefore to overcome it, we always do direct discussion for one another until we can understand perfectly and there is no sense of ambiguous meaning spread through our team. It is really critical to make the workmanship of document and application can run smoothly. It is also coupled with the consultation that we conduct with mentor to help us to sort out which is more appropriate or not to be implemented in our software project.

### What Went Right

During the process of project development, many task has been finished and turn out like what we have planned, which are:

* + 1. Have Enough Knowledge in Website Development

For the development of project, we choose the platform of web technology. It is because we already have enough knowledge in developing website by using HTTP, CSS and database. So in the making of website can be done easier and we can focus in uncommon things such as Google API implementation. Not only that, we also have learned about OOP Programming technique which really help us to build the website and for the future maintenance and modification of existing code inside the project will be easier.

* + 1. Interface Design

The interface design that we made has been made like what we have expected. But in the future, there is possibility that our design might be changed due to user requirement or to achieve the objectives of our project.

### What Went Wrong

During the development, there are some activities that we struggle to make and caused some issues which need to be communicated by each of team member. There are:

* + 1. Use Case Diagram

At first we already list all of use case that owned by each actor. But to divide the use case for four admins which are super admin, admin of province, admin of natural disaster, and admin command post is really confusing. So during the project development, we already re-create use case diagram until it gets what we have expected.

* + 1. Analysis Class Diagram

To plan the class diagram is quite hard for us. Because we have a little knowledge to build them in appropriate way even we have searched them in internet. So during the analysis class diagram development, we spend a lot of time to think and re-creating the variable and method to be included in the project.

### Lesson Learned

There are so many lesson that we got from conducting this project and also experiences of using new technology and uncommon software that included in project development. The lesson learned by us in detail are:

* + 1. Communication

In gathering all requirement or design of application, communication is a key to one another to avoid the uncertainty meaning in developing the project. Through the project development, we have improve our communication skill along with understanding that each team member has their own characteristic that should be differ in their social approach. We know we always face many miscommunication, but luckily each of team member is willing to explain in detail until one another understand perfectly.

* + 1. Time Management

In conducting the project, we are able to manage our time to focus on what is more important than the other. We also can divide the task to achieve the project goals and all the requirement can be completed even just for three use cases and interface that we made. For all we feel satisfy with what we have been faced and developed.

* + 1. PHP OOP

This project is implementing the PHP OOP technique. PHP OOP is a new thing for us but the OOP Programming lesson which have been taught in the college really help us to create the application easier. This new technique become our new experience and lesson in programming the website using PHP OOP Programing technique.

* + 1. Bootstrap

Through the development of project, we also gain so many knowledge in creating the responsible website using Bootstrap. Bootstrap is a free and open source front-end web framework for web design application.

### Further Improvement

Through the development we face many issues especially in creating use case diagram and also analysis class diagram because we need to specify all the features owned by each admin and also user. For several admin such as super admin, admin of province, admin of natural disaster and command post admin has their own features, it makes us to spend a lot of time to think about how to make the use case diagram and also what variable and method to be included in class diagram. Then for the design of website, we might change it to fulfill the user requirement or features that needed in admin requirement.

### Future Work

There are still many activities to go to achieve the project goals. For what we have achieved by this time are three prototypes of 38 use cases proposed in our project. Then the project aims, objectives, analysis class diagram and design artefact has been defined perfectly. But there are still possibility to make changes in those requirement and we will finish the rest of system requirement based on use case and sequence diagram that we made. For the future improvement, we would like to add navigation functionality using Google API technology for the user. So that they can directly go to the command post by the direction given by our navigation. It is also proposed to fulfill our objective of project, to facilitate the user to spend shorter time to find the right command post which still need some reinforcement.

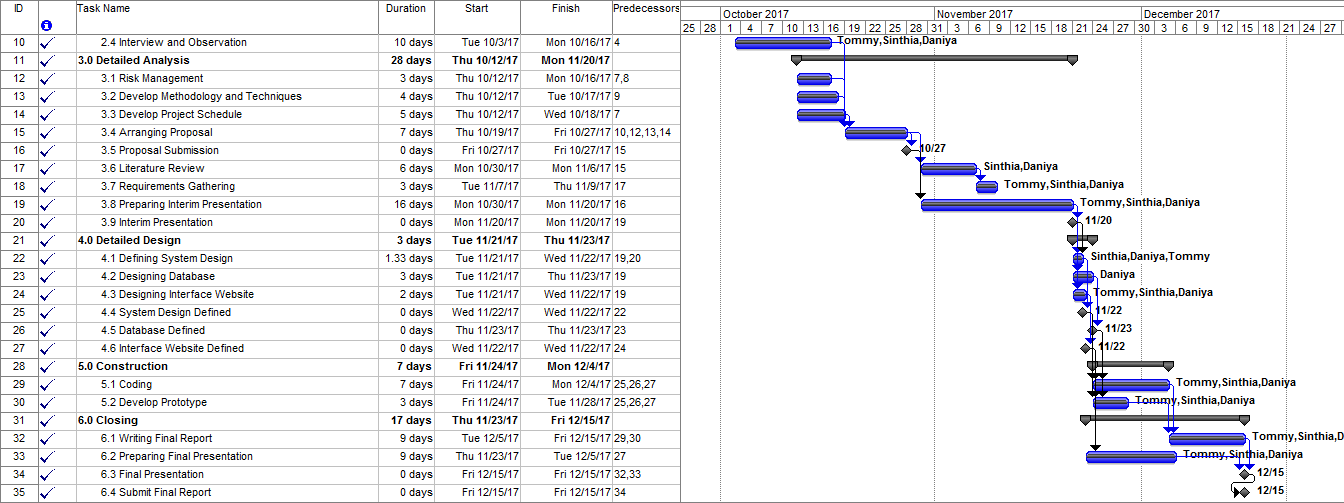
### Conclusion

Natural Disaster Information System is a web-based system which is designed to serve five actors, there are super admin, admin of province, admin of natural disaster, admin of command post and user. The reason to implement the system to several admin because we need to do approval for the information about people who is in charge in other division of admin, information about natural disaster and also logistical aid status for each post. So far during the three months of developing this project we already has proposed four objectives, analysis artifact, design artifact of project and also develop thirty eight use cases. There are three use cases that is implemented and has been tested, such as login for all division of admin, register user and login user.

During the development of project, we as a team gain a lot of lessons and experiences. This is our first time to build such a big project, even sometimes we still feel confuse of what we have included in the project requirement. But fortunately, with the aid of consultation with several people such as our mentor, volunteer, donator and also the agency of natural disaster recovery, those really help us to develop the functionality and features that will be implemented in our project. Not only by consultation, but also our communication skill to one another is also improved. Sometimes we face many ambiguous meaning, but we can discuss them in detail until one other understand perfectly. Another important lesson we learn is time management. This project make us to utilize our time wisely, because beside of the final project documentation, we need to complete the other three assignments given. But after all, we can follow our project schedule and stay on track to finish it on time. We hope in the future work, we can complete the rest of all requirement together and continue our commitment to produce a good quality of natural disaster information system which can help refugees that affected by natural disaster and help the user to access the updated information directly from the admin command post of each post easier.

### Appendix A: Updated Gantt Chart





**Appendix B: Progress Reports**

**B.1 September Project Report**

|  |
| --- |
| **Project Name:** Natural Disaster Information System  **Team Member:** Tommy, Daniya, Sinthia  **Date:** 30th September 2017  **Reporting Date:** September 2017 |
| **Work completed this reporting period:**  - Kick of meeting |
| **Work to complete next reporting period:**   * Inquire Idea * Consultation * Final Project Theme and Description Submission * Develop project background * Develop project aims and objectives * Develop resource plans * Interview and observation * Risk management * Develop methodology and techniques * Develop project schedule * Arranging proposal * Proposal submission |
| **What is going well and why:**  During the kick off meeting, we gather to list all ideas of what kind of project that we want to develop. In this stage communication through the team is going well because one another want to open their mind and give opinion about ideas of project. |
| **What is not going well and why:**  In choosing the ideas of project we want to develop because too many consideration and we feel worry about the risk that could inhibit us to develop the project. |
| **Suggestion/Issues:**  No suggestion |
| **Project Changes:** |

|  |
| --- |
|  |
| No changes |

### 2October Project Report

|  |
| --- |
| **Project Name:** Natural Disaster Information System  **Team Member:** Tommy, Daniya, Sinthia  **Date:** 28th October 2017  **Reporting Date:** October 2017 |
| **Work completed this reporting period:**   * Inquire Idea * Consultation * Final Project Theme and Description Submission * Develop project background * Develop project aims and objectives * Develop resource plans * Interview and observation * Risk management * Develop methodology and techniques * Develop project schedule * Arranging proposal * Proposal submission |
| **Work to complete next reporting period:**   * Literature review * Requirement gathering * Preparing interim presentation * Interim presentation * Defining system design * Designing database * Designing interface website * System design defined * Database defined * Interface website defined |

|  |
| --- |
| - Develop prototype |
| **What is going well and why:**  The consultation and interview with supervisor and mentor were conducted well so that we can define all part of proposal documentation |
| **What is not going well and why:**  Lack of knowledge in building information system, even related to natural disaster, so that we need to do some consultation and interview for brighter understanding. |
| **Suggestion/Issues:**  No suggestion |
| **Project Changes:**  No changes |

* 1. **November Project Report**

|  |
| --- |
| **Project Name:** Natural Disaster Information System  **Team Member:** Tommy, Daniya, Sinthia  **Date:** 29th November 2017  **Reporting Date:** November 2017 |
| **Work completed this reporting period:**   * Literature review * Requirement gathering * Preparing interim presentation * Interim presentation * Defining system design * Designing database * Designing interface website * System design defined * Database defined * Interface website defined * Develop prototype |
| **Work to complete next reporting period:**  - Coding |

|  |
| --- |
| * Writing final report * Preparing final presentation * Final presentation * Submit final report |
| **What is going well and why:**  In this month we have done the chapter 1 and 2, so we can focus in making the next chapter about analyzing and designing our website prototype. |
| **What is not going well and why:**  We face the difficulty in making analysis class diagram and use case diagram but after do some research we can solve it properly. |
| **Suggestion/Issues:**  No suggestion |
| **Project Changes:**  No changes |

### December Project Report

|  |
| --- |
| **Project Name:** Natural Disaster Information System  **Team Member:** Tommy, Daniya, Sinthia  **Date:** 14th December 2017  **Reporting Date:** December 2017 |
| **Work completed this reporting period:**   * Coding * Writing final report * Preparing final presentation |
| **Work to complete next reporting period:**   * Final presentation * Submit final report * Final Project 2 |
| **What is going well and why:**  We can finish all requirement of final project 1 documentation and also its presentation. |
| **What is not going well and why:** |

|  |
| --- |
| In developing the project along with the documentation are really consuming our energy and effort to finish the final project. But because we have strong commitment and teamwork to one another we can finish the project on time. |
| **Suggestion/Issues:**  No suggestion |
| **Project Changes:**  No changes |

**Appendix C: Minutes Meeting**

**BIT304 Final Year Project I**

**MyPosko.com: Natural Disaster Information System using Web-Based Minutes No. 1**

Date : 30th September 2017

Time : 9.00 am - 1.00 pm

Venue : International Class 2 Present : Tommy *(Project Leader)*

Ni Made Sinthia Pradnya Wednathi Rizqi Daniya

|  |  |  |
| --- | --- | --- |
| *No.* | **Matters Discussed** | *Action By* |
| **1.0** | *Topic 1: Discussing System Design* |  |
| 1.1 | ***Discussing overall system*** | All Members |
| **2.0** | *Topic 2: Discussing Requirements* |  |
| 2.1 | ***Discussing general requirements for the system.*** | All Members |
| **3.0** | **AOB** |  |
|  | Discussion end. The meeting has finished at 1.00 pm |  |

Minuted by: Rizqi Daniya Verified by: Tommy

### BIT304 Final Year Project I

**MyPosko.com: Natural Disaster Information System using Web-Based Minutes No. 2**

Date : 3rd October 2017

Time : 10.00 am - 2.00 pm

Venue : International Class 2 Present : Tommy *(Project Leader)*

Ni Made Sinthia Pradnya Wednathi Rizqi Daniya

|  |  |  |
| --- | --- | --- |
| *No.* | **Matters Discussed** | *Action By* |
| **1.0** | *Topic 1: Discussing System Design* |  |
| 1.1 | ***Discussing unit design for user sides*** | All Members |
| 1.2 | Discussing the system flow | All Members |
| **2.0** | **AOB** |  |
|  | Discussion end. The meeting has finished at 2.00 pm |  |

Minuted by: Rizqi Daniya Verified by: Tommy

### BIT304 Final Year Project I

**MyPosko.com: Natural Disaster Information System using Web-Based Minutes No. 3**

Date : 11th October 2017

Time : 10.00 am - 4.00 pm

Venue : International Class 2 Present : Tommy *(Project Leader)*

Ni Made Sinthia Pradnya Wednathi Rizqi Daniya

|  |  |  |
| --- | --- | --- |
| *No.* | **Matters Discussed** | *Action By* |
| **1.0** | *Topic 1: Discussing System Design* |  |
| 1.1 | ***Discussing unit design for admin sides*** | All Members |
| 1.2 | Discussing the requirements of system that should be completed and consulted to the Department of Natural Disaster Countermeasures in Bali. | All Members |
| **2.0** | **Topic 2: Discussing the Proposal Document** |  |
| 2.1 | Discussing the project background. | Dania, Tommy |
| 2.2 | Discussing project aims and objectives. | Daniya, Sinthia |
| 2.3 | Discussing resource plans. | Tommy, Sinthia |
| **3.0** | **AOB** |  |
|  | Discussion end. The meeting has finished at 4.00 pm |  |

Minuted by: Rizqi Daniya Verified by: Tommy

### BIT304 Final Year Project I

**MyPosko.com: Natural Disaster Information System using Web-Based Minutes No. 4**

Date : 12th October 2017

Time : 10.00 am - 2.00 pm

Venue : International Class 2 Present : Tommy *(Project Leader)*

Ni Made Sinthia Pradnya Wednathi Rizqi Daniya

|  |  |  |
| --- | --- | --- |
| *No.* | **Matters Discussed** | *Action By* |
| **1.0** | *Topic 1: Discussing System Requirement* |  |
| 1.1 | ***Discussing the volunteers and refugees’ administration according to the command post.*** | All Members |
| 1.2 | Discussing the logistical aids distribution from donators to refugees. | All Members |
| **2.0** | **Topic 2: Discussing the Risk Management** |  |
| 2.1 | Discussing the risk management that might occurred during and after the project development. | Sinthia, Tommy |
| **3.0** | **Topic 3: Discussing the Technology Used** |  |
| 3.1 | Discussing the programming language that will be used in the project. | All Members |
| **4.0** | **AOB** |  |
|  | Discussion end. The meeting has finished at 2.00 pm |  |

Minuted by: Rizqi Daniya

Verified by: Tommy

### BIT304 Final Year Project I

**MyPosko.com: Natural Disaster Information System using Web-Based Minutes No. 5**

Date : 16th October 2017

Time : 10.00 am - 5.00 pm

Venue : International Class 2 Present : Tommy *(Project Leader)*

Ni Made Sinthia Pradnya Wednathi Rizqi Daniya

|  |  |  |
| --- | --- | --- |
| *No.* | **Matters Discussed** | *Action By* |
| **1.0** | *Topic 1: Discussing Methodology* |  |
| 1.1 | ***Discussing the methodology about waterfall and prototyping and its phases.*** | Sinthia, Daniya |
| **2.0** | **Topic 2: Discussing the Schedule** |  |
| 2.1 | Discussing the schedule of the project and the tasks of waterfall and prototyping methodology. | All Members |
| **3.0** | **Topic 3: Discussing Proposal Document** |  |
| 3.1 | Discussing the proposal parts that have been completed and the problem that occurred in document writing. | All Members |
| **4.0** | **AOB** |  |
|  | There is no additional issue. The meeting ended at 5.00 pm |  |

Minuted by: Rizqi Daniya Verified by: Tommy

### BIT304 Final Year Project I

**MyPosko.com: Natural Disaster Information System using Web-Based Minutes No. 6**

Date : 25th October 2017

Time : 3.00 pm - 6.00 pm

Venue : International Class Laboratory 2. Present : Tommy *(Project Leader)*

Ni Made Sinthia Pradnya Wednathi Rizqi Daniya

|  |  |  |
| --- | --- | --- |
| *No.* | **Matters Discussed** | *Action By* |
| **1.0** | *Topic 1: Discussing the System Design* |  |
| 1.1 | ***Discussing the results of interview with the responsibility of Natural Disaster Countermeasures Department of Bali.*** | All Members |
| **2.0** | **Topic 2: Discussing Literature Reviews** |  |
| 2.1 | Discussing and distributing task of literature reviews. | All Members |
| **3.0** | **AOB** |  |
|  | There is no additional issue. The meeting ended at 5.00 pm |  |

Minuted by: Rizqi Daniya Verified by: Tommy

### BIT304 Final Year Project I

**MyPosko.com: Natural Disaster Information System using Web-Based Minutes No. 7**

Date : 25th October 2017

Time : 3.00 pm - 6.00 pm

Venue : International Class Laboratory 2. Present : Tommy *(Project Leader)*

Ni Made Sinthia Pradnya Wednathi Rizqi Daniya

|  |  |  |
| --- | --- | --- |
| *No.* | **Matters Discussed** | *Action By* |
| **1.0** | *Topic 1: Discussing the System Design* |  |
| 1.1 | ***Discussing the results of interview with the responsibility of Natural Disaster Countermeasures Department of Bali.*** | All Members |
| **2.0** | **Topic 2: Discussing Literature Reviews** |  |
| 2.1 | Discussing and distributing task of literature reviews. | All Members |
| **3.0** | **AOB** |  |
|  | There is no additional issue. The meeting ended at 6.00 pm |  |

Minuted by: Rizqi Daniya Verified by: Tommy

### BIT304 Final Year Project I

**MyPosko.com: Natural Disaster Information System using Web-Based Minutes No. 8**

Date : 8th November 2017

Time : 3.00 pm - 6.00 pm

Venue : International Class Laboratory 2. Present : Tommy *(Project Leader)*

Ni Made Sinthia Pradnya Wednathi Rizqi Daniya

|  |  |  |
| --- | --- | --- |
| *No.* | **Matters Discussed** | *Action By* |
| **1.0** | *Topic 1: Discussing Literature Reviews* |  |
| 1.1 | ***Discussing and following up literature reviews*** | All Members |
| **2.0** | **Topic 2: Discussing Interim Presentation** |  |
| 2.1 | Discussing about the interim presentation and the contents that would be presented. | All Members |
| **3.0** | **AOB** |  |
|  | There is no additional issue. The meeting ended at 6.00 pm |  |

Minuted by: Rizqi Daniya Verified by: Tommy

### BIT304 Final Year Project I

**MyPosko.com: Natural Disaster Information System using Web-Based Minutes No. 9**

Date : 20th November 2017

Time : 1.00 am - 4.00 pm

Venue : International Class Laboratory 2. Present : Tommy *(Project Leader)*

Ni Made Sinthia Pradnya Wednathi Rizqi Daniya

|  |  |  |
| --- | --- | --- |
| *No.* | **Matters Discussed** | *Action By* |
| **1.0** | *Topic 1: Discussing Interim Presentation* |  |
| 1.1 | ***Discussing and preparing the contents of interim presentation*** | All Members |
| 1.2 | Reviewing the video of interim presentation | All Members |
| **2.0** | **AOB** |  |
|  | There is no additional issue. The meeting ended at 4.00 pm |  |

Minuted by: Rizqi Daniya Verified by: Tommy

### BIT304 Final Year Project I

**MyPosko.com: Natural Disaster Information System using Web-Based Minutes No. 10**

Date : 21st November 2017

Time : 6.00 pm - 8.00 pm

Venue : International Class Laboratory 1. Present : Tommy *(Project Leader)*

Ni Made Sinthia Pradnya Wednathi Rizqi Daniya

|  |  |  |
| --- | --- | --- |
| *No.* | **Matters Discussed** | *Action By* |
| **1.0** | *Topic 1: Discussing Detailed Design* |  |
| 1.1 | ***Discussing all parts in detailed design*** | All Members |
| 1.2 | Distributing tasks to each member in detailed design | All Members |
| **2.0** | **AOB** |  |
|  | There is no additional issue. The meeting ended at 8.00 pm |  |

Minuted by: Rizqi Daniya Verified by: Tommy

### BIT304 Final Year Project I

**MyPosko.com: Natural Disaster Information System using Web-Based Minutes No. 11**

Date : 28th November 2017

Time : 2.00 pm - 8.00 pm

Venue : International Class Laboratory 2. Present : Tommy *(Project Leader)*

Ni Made Sinthia Pradnya Wednathi Rizqi Daniya

|  |  |  |
| --- | --- | --- |
| *No.* | **Matters Discussed** | *Action By* |
| **1.0** | *Topic 1: Discussing Prototype* |  |
| 1.1 | ***Discussing the prototype*** | Sinthia, Daniya |
| 1.2 | Discussing expanded use case diagram | Sinthia, Daniya |
| 1.3 | Discussing sequence diagrams | Daniya |
| 1.4 | Discussing web interface | Tommy |
| **2.0** | **AOB** |  |
|  | There is no additional issue. The meeting ended at 8.00 pm |  |

Minuted by: Rizqi Daniya Verified by: Tommy

### BIT304 Final Year Project I

**MyPosko.com: Natural Disaster Information System using Web-Based Minutes No. 12**

Date : 6th December 2017

Time : 2.00 pm - 8.00 pm

Venue : International Class Laboratory 2. Present : Tommy *(Project Leader)*

Ni Made Sinthia Pradnya Wednathi Rizqi Daniya

|  |  |  |
| --- | --- | --- |
| *No.* | **Matters Discussed** | *Action By* |
| **1.0** | *Topic 1: Discussing Detailed Design* |  |
| 1.1 | ***Reviewing detailed design*** | All Members |
| **2.0** | **Topic 2: Discussing Testing** |  |
| 2.1 | Discussing testing procedures for use case that have been completed. | All Members |
| **3.0** | **AOB** |  |
|  | There is no additional issue. The meeting ended at 8.00 pm |  |

Minuted by: Rizqi Daniya Verified by: Tommy

### BIT304 Final Year Project I

**MyPosko.com: Natural Disaster Information System using Web-Based Minutes No. 13**

Date : 12th December 2017

Time : 10.00 am - 4.00 pm

Venue : International Class 2. Present : Tommy *(Project Leader)*

Ni Made Sinthia Pradnya Wednathi Rizqi Daniya

|  |  |  |
| --- | --- | --- |
| *No.* | **Matters Discussed** | *Action By* |
| **1.0** | *Topic 1: Discussing Testing* |  |
| 1.1 | ***Reviewing detailed design*** | All Members |
| **2.0** | **Topic 2: Discussing Evaluation** |  |
| 2.1 | Discussing evaluation document | All Members |
| 2.2 | Reviewing all parts of document |  |
| **3.0** | **AOB** |  |
|  | There is no additional issue. The meeting ended at 4.00 pm |  |

Minuted by: Rizqi Daniya Verified by: Tommy

### Appendix D: Team Contract Team Member

Tommy E1400464

* + Rizqi Daniya E1400459
  + Ni Made Sinthia Pradny Wedanthi E1400454

### Code of Conduct

As a project team, we will :

1. Overcome the problem during the project
2. Build a strong teamwork and give support to each other to finish the project on time
3. Generate the task division to each member evenly
4. Concern in individual’s approach

### Participation

As a project team, we will:

1. Take a part in every meeting of project
2. Being responsible for the given task
3. Provide solution and suggestion when facing the project’s problem
4. Accept any recommendation to achieve our project goals

### Communication

As a project team, we will:

1. Communicate clearly and in detail about our project goals and ideas
2. Speak out the problem we face and discuss it to find the solution
3. Respect each opinion from team member related to project

### Problem Solving

As a project team, we will:

1. Discuss the difficulty during the project development
2. Provide some suggestion to solve the problem
3. Understand and listen intensely to each other problem
4. Provide a quick action to finish the project on time while facing some obstacle

### Meeting Guidelines

As a project team, we will:

1. Set the ideas of project and divide the task evenly
2. Keep on track and evaluate the outcome of each meeting
3. Be understandable to any problem and find out the solution
4. Use any additional tools to support the result of document or project.
5. Record and evaluate the finished task whether it’s already in accordance with our goal.

### Appendix E: Vision

* 1. **Introduction**

Entering the rapid development of technology, every information can be obtained very easy and accessible from website or social media. But sometimes we could not specify the truth behind information whether it is the valid information from the actual place or just rumor. For example, the current situation of Agung Mountain eruption that happened in Bali. We have found so many issues that happened during the evacuation of refuges, especially the uneven distribution of logistical aid to each post. There are some command post still need reinforcement of aid, otherwise there are some command post which has a pile of unnecessary goods, for example is instant noodle. Therefore, it is needed an information system that could provide the user to know where to give their donation appropriately and evenly. By using our natural information system, that functionality are implemented. Not only to help people find the right point of command post that still need support, but through this system it allows people to contribute themselves as volunteer. Beside of that, stock of logistical aid management is also provided to the admin of command post to do their report more convenient. We hope that our project development could decrease the uneven distribution of logistical aids and this system can be a medium of information for people who want to give their donations in accordance with what is needed by each post.

### Positioning

* + 1. **Problem Statement**

|  |  |
| --- | --- |
| **The problem** | Uneven distribution of donation to each natural disaster command post and many fake news spread through the community especially the needs of logistical aid in each post. |
| **Affects** | Refugees |
| **The impact of which is** | Many people who wants to give their donation feel confuse and do not know the needs of logistical aid needed in each post. |
| **A successful solution would be** | Natural disaster information system that can provide them valid information directly from the admin of command post in each |

|  |  |
| --- | --- |
|  |  |
|  | area they chosen to avoid them from fake news and also know the certain area of post that need reinforcement. |

### Product Position Statement

|  |  |
| --- | --- |
| **For** | Society, Natural disaster recovery agency, Government |
| **Who** | Needs a valid information about natural disaster, command post in the area chosen, and the needs of logistical aids in each post. |
| **Natural Disaster Information System** | System which provides user the valid information directly from each command post admin in chosen area, facilitate user to donate their money, event or contribute themselves as volunteer, and the management of logistical aid stock for command post admin. |
| **That** | Offered in web platform that can be freely accessed in many platform such as smartphone, personal computer or laptop. It is used to manage the stock in and out for admin, and valid information related to natural disaster and command post to user in chosen area. |
| **Unlike** | The current website of natural disaster information that still not update the data of refugees in real time and it does not show the list of needs in each command post. |
| **Our product** | Give an advantages in immediacy which means that the user who are using android, iPhone, Blackberry and PC can access the web application through their web browser easily. This web product also can be accessed everywhere and every time the user wants, the other advantages are increasing the productivity of work because user can do the sorting, searching item of information to be more efficient in time and effort. Meanwhile admin can do their management of logistical stock in and out for per day or per month use. Next is, our product also support the paperless campaign, which is implemented in our registration of volunteer or donator form. Therefore they only need to fill the |

|  |  |
| --- | --- |
|  |  |
|  | form and wait for the confirmation from the admin to participate in helping the victim of natural disaster. |

* 1. **Stakeholder Description**
     1. **Stakeholder Summary**

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Responsibility** |
| **Supervisor** | Person or agency who participate in recovering the natural disaster such as disaster recovery agency, volunteer, and donator | * Gives full information about the workflow conducted in recovering the natural disaster. * Help project team to define the feature or functionality that can decrease the possibility of uneven distribution of logistical aid. |
| **Lecturer** | Person who give us knowledge and suggestion in developing proper documentation and logical programming of our project | * Suggest what functionality that need to be featured in project. * Helping to solve the problem of making documentation and application such as coding, methodology, design artifact and other. |
| **Project leader** | Person who leads the certain project and motivate their team to achieve the project goal | * Initiate the ideas of project to be developed * Decide tasks, roles and responsibility of other team member * Suggest a solution for certain problem that team member face |

|  |  |  |
| --- | --- | --- |
|  |  | * Ensure the team work in accordance with project goals and schedule |
| **Project team** | The member of project | * Gather all information and requirement of project development * Analyzing and designing the project * Produce the project deliverables * Do a testing and implementation of system conducted. |
| **Super admin** | Person who is in charge to manage the admin of province and general information of website. | * Manage admin of province * Approve donation of money and testimonial given |
| **Admin of Province** | Person who is in charge to manage the province | * Manage province * Manage admin of natural disaster * Send testimonials |
| **Admin of Natural Disaster** | Person who is in charge to manage the natural disaster that happened in province chosen | * Manage natural disaster * Manage admin command post * Submit testimonials |
| **Admin of Command Post** | Person who manage command post based on natural disaster and province chosen | * Manage command post * Manage logistical aid stock * Approve event * Approve volunteer * Submit testimonial |

|  |  |  |
| --- | --- | --- |
| **User** | Person who use the natural disaster information system and user can become volunteer or donator after the login of user done. | * Become volunteer * Donate event * Donate money |

### E.4 User Environment

|  |  |
| --- | --- |
| **Hardware Requirement** | |
| Web Platform | A computer which has minimum Intel Core i3 with 2.40GHz, 2GB of RAM, using graphic processor of NVidia, 4GB free space of hard disk. |
| **Software Requirement** | |
| Web Platform | Any computer which implement windows 7 operating system or above with installed web browser of Mozilla Firefox or Chrome. |

**Appendix F : Project Evaluation**



**MYPOSCO.COM : NATURAL DISASTER INFORMATION SYSTEM USING WEB-BASED**

**PROJECT EVALUATION**

**E1400462 TOMMY**

**Submitted to the**

**FACULTY OF COMPUTING AND DIGITAL TECHNOLOGY**

**(SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY)**

**In partial fulfilment of the requirements for the degree of**

**BACHELOR OF INFORMATION TECHNOLOGY (HONS)**

**HELP UNIVERSITY SEPTEMBER 2017**

**Project Evaluation**

1. **Introduction**

In this project we have purpose to develop an information system on web based that would be very useful if there occurred natural disaster attack. As my experience, I was a volunteer for Mount Agung rescue on last September 2017. I had some problems on looking for the evacuation site that needs more help and we got confuse on what kind of logistical aids we should give to them. I had visited some of evacuation site and some of them has full of the logistical aid, then we should to find another evacuation site that need more logistical aid. From these reasons, we want to solve most of the people’s problem about finding appropriate command post or evacuation site.

### Evaluation of Objectives

After finishing this project for BIT304 that means not all the functionalities have been developed, we only developed three use cases from them. We have faced some problem such as:

* + Communication problem because sometimes when we need to discuss our project we could not meet each other then we solved it by discussing through instant messages or voice message.
  + Designing system also made us confuse about the procedures to register admin for each command post and being a volunteer to its command post.

### Further Improvement

During this project we often consult to our supervisor about the problem that we faced in this project and to overcome the lack experience of web programming language we consult to the web developer that has more experience on developing web application.

### Future Work

In our next project, which is continuation from this project we have to emphasize on using web programming language as object oriented and in using google maps API to make navigation to particular location. We have to ensure that all the functionalities will work well.

### Conclusion

After finishing BIT304 project, it made me learn many things especially for project management and software engineering and we have to learn and ensure that we will do better for BIT305 so that this web application will have good quality products.



# MYPOSCO.COM : NATURAL DISASTER INFORMATION SYSTEM USING WEB-BASED

**PROJECT EVALUATION**

## E1400459 RIZQI DANIYA

**Submitted to the**

**FACULTY OF COMPUTING AND DIGITAL TECHNOLOGY**

### (SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY)

**In partial fulfilment of the requirements for the degree of**

**BACHELOR OF INFORMATION TECHNOLOGY (HONS)**

**HELP UNIVERSITY SEPTEMBER 2017**

**Project Evaluation**

1. **Introduction**

In this project we have purposed on building a website application that has function to give information about something that related to natural disaster that occurred in Indonesia, such as the number of refugees, the number of command post in a disaster in particular province, the location of command post or evacuation site. The users can also participate on being volunteer in that evacuation site or hold an event or donate money.

### Evaluation of Objectives

We have successfully developed only few use cases in this project, we only developed login and register menu. The rest of time we used it to make documentation and some other requirements such as designing database, designing use cases, class diagram, designing interface and so forth.

The problem that we faced during developing this project:

* + Communication problem because sometimes we can communicate directly face to face among members that makes us sometimes misunderstand about what the other team meant in his/her words.
  + Skill in designing system is still poor, because we still learn about software engineering and we have only a little experience about how to build software using good software engineering and project management, we still need to learn more and more.

### Further Improvement

In designing this system, we had some problem at first about some procedures that is related in natural disaster countermeasures. We had not known the procedures how to register to be volunteer or how to donate event or how to donate money to evacuation site. Then our supervisor, Mr Yudi Agusta suggested us to go to BPBD Bali, then we decided to go there to have an interview and doing some observation there. BPBD Bali is Natural Disaster Countermeasures Department for Bali. We asked them about these procedures, so we can design the appropriate system.

### Future Work

For future we have to learn about framework in web programming language and google maps API so that the coding will be done easier and more efficient. We have to finished the rest of use cases so that we can develop a good product with better quality.

### Conclusion

After finishing this paper about BIT304, I realized that it made some improvement on my self especially about software engineering and project management. For the next subject I need to improve more and more about these and also in terms of programming language.



# MYPOSCO.COM : NATURAL DISASTER INFORMATION SYSTEM USING WEB-BASED

**PROJECT EVALUATION**

## E1400454 NI MADE SINTHIA PRADNYA WEDANTHI

**Submitted to the**

**FACULTY OF COMPUTING AND DIGITAL TECHNOLOGY**

### (SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY)

**In partial fulfilment of the requirements for the degree of**

**BACHELOR OF INFORMATION TECHNOLOGY (HONS)**

**HELP UNIVERSITY SEPTEMBER 2017**

**Project Evaluation**

1. **Introduction**

We have purpose on developing web application for natural disaster information which is called myPosco.com. This web application will give valid information about something related to natural disaster. This web application is developed for all of areas in Indonesia so it is for national scope.

### Evaluation of Objectives

We have successfully developed the design system along with the prototype but on doing this project we have some problems such as:

* + Time management because we did not only take this subject, we have to manage our time wisely, and sometimes we found that it was difficult.
  + We also had some problem when we want to design the flow of the system, because we do not know the procedure about natural disaster countermeasure.

### Further Improvement

Even though, it is quite difficult to deal with all the project in same times we have completed all the project and assignment in this semester. About the system design, we decided to have some interview to natural disaster countermeasure agency in Bali, to know the procedure of their work, and we got some suggestion and recommendation about the system from them.

### Future Work

In next project we thought that we have to do further interview to natural disaster countermeasure agency in Bali (BPBD Bali) about our prototype and to get the feedback whether it has meet the needs of the users and we also need to complete the rest of use case so that our web application will run maximally.

### Conclusion

This project has given me a lot of experience on doing project and to manage them as in software engineering, and we also learn how to work with the members in team work that has different character to me. We hope that we can do better on next project.