****

Software Requirements Specification

**Group No 07**

**24.07.2020**

**Group No 07**

Information Systems

**UCSC**

Group Members

|  |  |
| --- | --- |
| Index No | Name |
| 18020798 | Asini Pathmila Silva |
| 18020291 | Ruwanthi Hemachandra |
| 18020704 | Hansaka Sandaruwan |
| 18020356 | Sachini Maneesha |

Project Supervisor

Ms. Kulani Sumanasekara

Project Co-Supervisor

Ms. Gayani Rupasinghe

Table of contents

[Table of contents 2](#_Toc46365953)

[Table of figures 7](#_Toc46365954)

[Table of tables 11](#_Toc46365955)

[1. Introduction 12](#_Toc46365956)

[1.1 Domain Description 12](#_Toc46365957)

[1.2 Proposed Solution 12](#_Toc46365958)

[1.3 Current System & It’s Limitation 13](#_Toc46365959)

[1.4 Objectives & Goals 14](#_Toc46365960)

[1.4.1 Project Goal 14](#_Toc46365961)

[1.4.2 Project Objectives 14](#_Toc46365962)

[1.5 Assumptions, Constraints & Limitations. 15](#_Toc46365963)

[1.5.1 Project constraints 15](#_Toc46365964)

[1.5.2 Assumptions 16](#_Toc46365965)

[1.5.3 Limitation 17](#_Toc46365966)

[2. Feasibility Study 18](#_Toc46365967)

[2.1 Technical Feasibility 18](#_Toc46365968)

[2.2 Economic feasibility 19](#_Toc46365969)

[2.3 Legal and Ethical feasibility 19](#_Toc46365970)

[2.4 Schedule feasibility 19](#_Toc46365971)

[2.5 Operational feasibility 20](#_Toc46365972)

[3. Requirements 21](#_Toc46365973)

[3.1 Stakeholders 21](#_Toc46365974)

[3.2 Use case diagram 23](#_Toc46365975)

[3.3.1 Actor wise use cases 24](#_Toc46365976)

[3.3 Use case Descriptions 27](#_Toc46365977)

[3.2.1 Use case descriptions for Admin’s functionalities. 27](#_Toc46365978)

[3.2.2 Use case descriptions for User’s functionalities. 32](#_Toc46365979)

[3.2.3 Use case descriptions for Guide’s functionalities. 36](#_Toc46365980)

[3.2.4 Use case descriptions for the hotel's functionalities. 38](#_Toc46365981)

[3.2.5 Use case descriptions for vehicle owner’s functionalities. 40](#_Toc46365982)

[3.4 Functional Requirement Identification 42](#_Toc46365983)

[3.5 Non-Functional Requirements 44](#_Toc46365984)

[3.5.1 Identification of Quality Attributes 44](#_Toc46365985)

[3.5.2 Suggestion to achieve Quality Attributes 44](#_Toc46365986)

[4. Proposed System’s Architecture 46](#_Toc46365987)

[4.1 System’s Architecture 46](#_Toc46365988)

[4.3 Components & their responsibilities 47](#_Toc46365989)

[4.4 Component diagram 49](#_Toc46365990)

[4.5 Component interactions 50](#_Toc46365991)

[5. System’s Design 51](#_Toc46365992)

[5.1 Class diagram 51](#_Toc46365993)

[5.2 ER diagram 52](#_Toc46365994)

[5.3 Sequence diagrams 53](#_Toc46365995)

[5.3.1 User’s Sequence diagrams 53](#_Toc46365996)

[5.3.2 Admin’s Sequence diagrams 59](#_Toc46365997)

[5.3.3 Hotel’s Sequence diagrams 66](#_Toc46365999)

[5.3.3.2 Hotel changes availability 66](#_Toc46366000)

[5.3.3.3 Hotel changes password 67](#_Toc46366001)

[5.3.3.4 Hotel updates profile 67](#_Toc46366002)

[5.3.4 Vehicle owner’s Sequence diagrams 69](#_Toc46366003)

[5.3.5 Guide’s Sequence diagrams 72](#_Toc46366004)

[5.4 Activity Diagrams 75](#_Toc46366006)

[5.4.1 Activity diagrams for User 75](#_Toc46366007)

[5.4.2 Activity diagrams for Admin 87](#_Toc46366008)

[5.4.3 Activity diagrams for Hotel 102](#_Toc46366009)

[5.4.4 Activity diagrams for Guide 107](#_Toc46366010)

[5.4.5 Activity diagrams for Vehicle 112](#_Toc46366011)

[5.5 State Transition Diagrams 117](#_Toc46366012)

[5.5.1 Login for user, hotel, guide and vehicle owner 117](#_Toc46366013)

[5.5.2 User books a travel package 117](#_Toc46366014)

[5.5.3 User does the payment for booking a travel package 118](#_Toc46366015)

[5.5.4 Delete profile for user, hotel, guide and vehicle owner 118](#_Toc46366016)

[5.5.5 Change availability for guide and vehicle 119](#_Toc46366017)

[5.5.6 Change availability for hotel 119](#_Toc46366018)

[6. Interface Flow Diagrams 120](#_Toc46366019)

[6.1 User’s interface flow diagram 120](#_Toc46366020)

[6.2 Admin’s interface flow diagram 120](#_Toc46366021)

[6.3 Hotel account’s interface flow diagram 121](#_Toc46366022)

[6.4 Guide account’s interface flow diagram 121](#_Toc46366023)

[6.5 Vehicle owner’s interface flow diagram 122](#_Toc46366024)

[7. Wired interfaces 123](#_Toc46366025)

[7.1 User’s interfaces 123](#_Toc46366026)

[7.1.1 Login page 123](#_Toc46366027)

[7.1.2 Account page 123](#_Toc46366028)

[7.1.3 Signup page 124](#_Toc46366029)

[7.1.4 Index page 125](#_Toc46366030)

[7.1.5 Home page 126](#_Toc46366031)

[7.1.6 About page 127](#_Toc46366032)

[7.1.7 Contact page 127](#_Toc46366033)

[7.1.8 Category page 128](#_Toc46366034)

[7.1.9 Package page 128](#_Toc46366035)

[7.1.16 Package more details page 129](#_Toc46366036)

[7.1.10 Book page 129](#_Toc46366037)

[7.1.11 Payhere page 130](#_Toc46366038)

[7.1.12 Thanking page 130](#_Toc46366039)

[7.1.13 destination page 131](#_Toc46366040)

[7.1.14 Search destination page 131](#_Toc46366041)

[7.1.15 Packages for destination page 132](#_Toc46366042)

[7.1.15 View profile page 132](#_Toc46366043)

[7.1.16 View booking page 133](#_Toc46366044)

[7.1.17 View comment page 133](#_Toc46366045)

[7.1.18 Add comment page 134](#_Toc46366046)

[7.1.19 Update profile page 134](#_Toc46366047)

[7.1.20 Delete profile page 135](#_Toc46366048)

[7.1.21 Change the password page 135](#_Toc46366049)

[7.2 Admin’s interfaces 136](#_Toc46366050)

[7.2.1 Home page 136](#_Toc46366051)

[7.2.2 Add category page 136](#_Toc46366052)

[7.2.3 Add destination page 137](#_Toc46366053)

[7.2.4 Add Subcategory page 137](#_Toc46366054)

[7.2.5 Add package page 138](#_Toc46366055)

[7.2.6 Check availability page1 139](#_Toc46366056)

[7.2.7 Check availability page2 139](#_Toc46366057)

[7.2.8 Delete category page 140](#_Toc46366058)

[7.2.9 Delete subcategory page 140](#_Toc46366059)

[7.2.10 Delete package page 141](#_Toc46366060)

[7.2.11 Update Category page1 141](#_Toc46366061)

[7.2.12 Update category page2 142](#_Toc46366062)

[7.2.13 Update subcategory page1 142](#_Toc46366063)

[7.2.14 Update subcategory page2 143](#_Toc46366064)

[7.2.15 Update package page1 143](#_Toc46366065)

[7.2.16 Update package page2 144](#_Toc46366066)

[7.2.17 View account page1 145](#_Toc46366067)

[7.2.18 View account page2 145](#_Toc46366068)

[7.2.19 View account page3 146](#_Toc46366069)

[7.2.20 View booking page 146](#_Toc46366070)

[7.2.21 View report page1 147](#_Toc46366071)

[7.2.22 View report page2 147](#_Toc46366072)

[7.2.23 View report page3 148](#_Toc46366073)

[7.3 Hotel account’s interfaces 149](#_Toc46366074)

[7.3.1 Home page 149](#_Toc46366075)

[7.3.2 View profile page 150](#_Toc46366076)

[7.3.3 Signup page 151](#_Toc46366077)

[7.3.4 Update Profile page 152](#_Toc46366078)

[7.3.5 Change Password page 153](#_Toc46366079)

[7.3.6 Delete Profile page 153](#_Toc46366080)

[7.3.7 Change availability page 154](#_Toc46366081)

[7.4 Guide account’s interfaces 155](#_Toc46366082)

[7.4.1 Home page 155](#_Toc46366083)

[7.4.2 View profile page 156](#_Toc46366084)

[7.4.3 Signup page 157](#_Toc46366085)

[7.4.4 Update profile page 158](#_Toc46366086)

[7.4.5 Change password page 159](#_Toc46366087)

[7.4.6 Delete profile 159](#_Toc46366088)

[7.4.7 Change availability page 160](#_Toc46366089)

[7.5 Vehicle owner’s interfaces 161](#_Toc46366090)

[7.5.1 Home page 161](#_Toc46366091)

[7.5.2 View profile page 162](#_Toc46366092)

[7.5.3 Signup page 163](#_Toc46366093)

[163](#_Toc46366094)

[7.5.4 Update profile page 164](#_Toc46366095)

[7.5.5 Change password page 165](#_Toc46366096)

[7.5.6 Delete profile page 165](#_Toc46366097)

[7.5.7 Change availability page 166](#_Toc46366098)

[8. Declaration 167](#_Toc46366099)

[9. Reference 168](#_Toc46366100)

# Table of figures

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | *Figure 01: Schedule feasibility graph* | *20* | | *Figure 02: Use-case diagram* | *23* | | *Figure 03: Use case Diagram for the user* | *24* | | *Figure 04: Use case Diagram for the hotel* | *25* | | *Figure 05: Use case Diagram for the guide* | *25* | | *Figure 06: Use case Diagram for the vehicle owner* | *26* | | *Figure 07: Use case Diagram for the admin* | *26* | | *Figure 08: System Architecture* | *46* | | *Figure 09: Component Diagram* | *49* | | *Figure 10: Class Diagram* | *51* | | *Figure 11: ER Diagram* | *52* | | *Figure 12: Sequence diagram for user booking travel package* | *53* | | *Figure 13: Sequence diagram for user cancel a booking* | *54* | | *Figure 14: Sequence diagram for user adds comment* | *55* | | *Figure 15: Sequence diagram for user updates password* | *55* | | *Figure 16: Sequence diagram for user deletes profile* | *56* | | *Figure 17: Sequence diagram for user login* | *56* | | *Figure 18: Sequence diagram for user searches destinations* | *57* | | *Figure 19: Sequence diagram for user sends messages* | *57* | | *Figure 20: Sequence diagram for user signup* | *58* | | *Figure 21: Sequence diagram for user updates profile* | *58* | | *Figure 22: Sequence diagram for admin adds a category* | *59* | | *Figure 23: Sequence diagram for admin adds a destination* | *59* | | *Figure 24: Sequence diagram for admin adds a package* | *60* | | *Figure 25: Sequence diagram for admin adds a subcategory* | *60* | | *Figure 26: Sequence diagram for admin checks the availability* | *61* | | *Figure 27: Sequence diagram for admin deletes a category* | *61* | | *Figure 28: Sequence diagram for admin deletes package* | *62* | | *Figure 29: Sequence diagram for admin deletes a subcategory* | *62* | | *Figure 30: Sequence diagram for admin login* | *63* | | *Figure 31: Sequence diagram for admin updates a category* | *63* | | *Figure 32: Sequence diagram for admin updates a package* | *64* | | *Figure 33: Sequence diagram for admin updates a subcategory* | *64* | | *Figure 34: Sequence diagram for admin views accounts* | *65* | | *Figure 35: Sequence diagram for admin views reports* | *65* | | *Figure 36: Sequence diagram for hotel login* | *66* | | *Figure 37: Sequence diagram for hotel changes availability* | *66* | | *Figure 38: Sequence diagram for hotel changes password* | *67* | | *Figure 39: Sequence diagram for hotel updates profile* | *67* | | *Figure 40: Sequence diagram for hotel sign up* | *68* | | *Figure 41: Sequence diagram for hotel deletes profile* | *68* | | *Figure 42: Sequence diagram for vehicle owner login* | *69* | | *Figure 43: Sequence diagram for vehicle owner changes password* | *69* | | *Figure 44: Sequence diagram for vehicle owner signup* | *70* | | *Figure 45: Sequence diagram for vehicle owner changes availability* | *70* | | *Figure 46: Sequence diagram for vehicle owner update profile* | *71* | | *Figure 47: Sequence diagram for vehicle owner deletes profile* | *71* | | *Figure 48: Sequence diagram for guide login* | *72* | | *Figure 49: Sequence diagram for guide changes availability* | *72* | | *Figure 50: Sequence diagram for guide changes password* | *73* | | *Figure 51: Sequence diagram for guide updates the profile* | *73* | | *Figure 52: Sequence diagram for guide signup* | *74* | | *Figure 53: Sequence diagram for guide deletes profile* | *74* | | *Figure 54: Activity diagram for user login to the system* | *75* | | *Figure 55: Activity diagram for user creates an account* | *76* | | *Figure 56: Activity diagram for user adds comments* | *77* | | *Figure 57: Activity diagram for user books travel package* | *78* | | *Figure 58: Activity diagram for user cancels the booking* | *79* | | *Figure 59: Activity diagram for user deletes the account* | *80* | | *Figure 60: Activity diagram for user does the payment* | *81* | | *Figure 61: Activity diagram for user search* | *82* | | *Figure 62: Activity diagram for user updates the account* | *83* | | *Figure 63: Activity diagram for user view [not registered user]* | *84* | | *Figure 64: Activity diagram for user view [registered user]* | *85* | | *Figure 65: Activity diagram for user sends message* | *86* | | *Figure 66: Activity diagram for admin login to the system* | *87* | | *Figure 67: Activity diagram for admin views accounts* | *88* | | *Figure 68: Activity diagram for admin views bookings* | *89* | | *Figure 69: Activity diagram for admin views reports* | *90* | | *Figure 70: Activity diagram for admin adds travel package* | *91* | | *Figure 71: Activity diagram for admin adds category* | *92* | | *Figure 72: Activity diagram for admin adds subcategory* | *93* | | *Figure 73: Activity diagram for admin adds destinations* | *94* | | *Figure 74: Activity diagram for admin checks availability* | *95* | | *Figure 75: Activity diagram for admin deletes category* | *96* | | *Figure 76: Activity diagram for admin deletes subcategory* | *97* | | *Figure 77: Activity diagram for admin deletes travel package* | *98* | | *Figure 78: Activity diagram for admin deletes travel package* | *99* | | *Figure 79: Activity diagram for admin updates subcategory* | *100* | | *Figure 80: Activity diagram for admin updates travel package* | *101* | | *Figure 81: Activity diagram for hotel login to the system* | *102* | | *Figure 82: Activity diagram for hotel creates an account* | *103* | | *Figure 83: Activity diagram for hotel updates the account* | *104* | | *Figure 84: Activity diagram for hotel deletes the account* | *105* | | *Figure 85: Activity diagram for hotel changes the availability* | *106* | | *Figure 86: Activity diagram for guide login to the system* | *107* | | *Figure 87: Activity diagram for guide creates an account* | *108* | | *Figure 88: Activity diagram for guide updates the account* | *109* | | *Figure 89: Activity diagram for guide deletes the account* | *110* | | *Figure 90: Activity diagram for guide changes the availability* | *111* | | *Figure 91: Activity diagram for vehicle login to the system* | *112* | | *Figure 92: Activity diagram for vehicle creates an account* | *113* | | *Figure 93: Activity diagram for vehicle updates the account* | *114* | | *Figure 94: Activity diagram for vehicle deletes the account* | *115* | | *Figure 95: Activity diagram for vehicle changes availability* | *116* | | *Figure 96: State transition diagram for accounts’ login* | *117* | | *Figure 97: State transition diagram for user books a travel package* | *117* | | *Figure 98: State transition diagram for user dose the payment* | *118* | | *Figure 99: State transition diagram for delete profile* | *118* | | *Figure 100: State transition diagram for changing the availability for guide and vehicle* | *119* | | *Figure 101: State transition diagram for changing the availability for hotel* | *119* | | *Figure 102: Interface flow diagram for the user* | *120* | | *Figure 103: Interface flow diagram for the admin* | *120* | | *Figure 104: Interface flow diagram for the hotel* | *121* | | *Figure 105: Interface flow diagram for the guide* | *121* | | *Figure 106: Interface flow diagram for the vehicle owner* | *122* | | *Figure 107: Interface for user login page* | *123* | | *Figure 108: Interface for user Account page* | *123* | | *Figure 109: Interface for user signup page* | *124* | | *Figure 110: Interface for Index page* | *125* | | *Figure 111: Interface for user home page* | *126* | | *Figure 112: Interface for About page* | *127* | | *Figure 113: Interface for Contact page* | *127* | | *Figure 114: Interface for Category page* | *128* | | *Figure 115: Interface for Package page* | *128* | | *Figure 116: Interface for package more details page* | *129* | | *Figure 117: Interface for Book page* | *129* | | *Figure 118: Interface for Payhere page* | *130* | | *Figure 119: Interface for Thanking page* | *130* | | *Figure 120: Interface for Destination page* | *131* | | *Figure 121: Interface for Search Destination page* | *131* | | *Figure 122: Interface for Packages for destination page* | *132* | | *Figure 123: Interface for user View profile page* | *132* | | *Figure 124: Interface for user View booking* | *133* | | *Figure 125: Interface for View comment page* | *133* | | *Figure 126: Interface for Add comment page* | *134* | | *Figure 127: Interface for user Update profile page* | *134* | | *Figure 128: Interface for user Delete profile* | *135* | | *Figure 129: Interface for user Change password* | *135* | | *Figure 130: Interface for admin Home page* | *136* | | *Figure 131: Interface for admin add category page* | *136* | | *Figure 132: Interface for admin add destination page* | *137* | | *Figure 133: Interface for admin add subcategory page* | *137* | | *Figure 134: Interface for admin add package page* | *138* | | *Figure 135: Interface for admin checks availability page1* | *139* | | *Figure 136: Interface for admin checks availability page2* | *139* | | *Figure 137: Interface for admin deletes category page* | *140* | | *Figure 138: Interface for admin deletes subcategory page* | *140* | | *Figure 139: Interface for admin deletes package page* | *141* | | *Figure 140: Interface for admin update category page1* | *141* | | *Figure 141: Interface for admin update category page2* | *142* | | *Figure 142: Interface for admin update subcategory page1* | *142* | | *Figure 143: Interface for admin update subcategory page2* | *143* | | *Figure 144: Interface for admin update package page1* | *143* | | *Figure 145: Interface for admin update package page2* | *144* | | *Figure 146: Interface for admin view account page1* | *145* | | *Figure 147: Interface for admin view account page2* | *145* | | *Figure 148: Interface for admin view account page3* | *146* | | *Figure 149: Interface for admin view booking page* | *146* | | *Figure 150: Interface for admin view report page1* | *147* | | *Figure 151: Interface for admin view report page2* | *147* | | *Figure 152: Interface for admin view report page3* | *148* | | *Figure 153: Interface for hotel home page* | *149* | | *Figure 154: Interface for hotel view profile page* | *150* | | *Figure 155: Interface for hotel signup* | *151* | | *Figure 156: Interface for hotel update profile* | *152* | | *Figure 157: Interface for hotel change password* | *153* | | *Figure 158: Interface for hotel delete profile page* | *153* | | *Figure 159: Interface for hotel change availability page* | *154* | | *Figure 160: Interface for guide home page* | *155* | | *Figure 161: Interface for guide view profile page* | *156* | | *Figure 162: Interface for guide signup page* | *157* | | *Figure 163: Interface for guide update profile page* | *158* | | *Figure 164: Interface for guide change password page* | *159* | | *Figure 165: Interface for guide delete profile page* | *159* | | *Figure 166: Interface for guide change availability page* | *160* | | *Figure 167: Interface for vehicle owner’s home page* | *161* | | *Figure 168: Interface for vehicle owner’s view profile page* | *162* | | *Figure 169: Interface for vehicle owner’s signup page* | *163* | | *Figure 170: Interface for vehicle owner’s update profile page* | *164* | | *Figure 171: Interface for vehicle owner’s change password page* | *165* | | *Figure 172: Interface for vehicle owner’s delete profile page* | *165* | | *Figure 173: Interface for vehicle owner’s change availability page* | *166* | |  |
| Table of tables  |  |  | | --- | --- | | *Table 01: Current systems and its limitations* | *14* | | *Table 02: Use case description for admin logins into the system* | *28* | | *Table 03: Use case description for admin views accounts’ profiles.* | *28* | | *Table 04: Use case description for admin views booking list.* | *28* | | *Table 05: Use case description for admin views the reports.* | *29* | | *Table 06: Use case description for admin adds a travel package.* | *29* | | *Table 07: Use case description for admin adds a category* | *29* | | *Table 08: Use case description for admin adds a subcategory* | *30* | | *Table 09: Use case description for admin adds a destination* | *30* | | *Table 10: Use case description for admin checks availability* | *30* | | *Table 11: Use case description for admin deletes a category* | *31* | | *Table 12: Use case description for admin deletes a subcategory* | *31* | | *Table 13: Use case description for admin deletes a travel package* | *31* | | *Table 14: Use case description for admin updates a category* | *32* | | *Table 15: Use case description for admin updates a subcategory* | *32* | | *Table 16: Use case description for admin updates a travel package* | *32* | | *Table 17: Use case description for user logins into the system* | *33* | | *Table 18: Use case description for user creates an account* | *33* | | *Table 19: Use case description for user adds comments* | *33* | | *Table 20: Use case description for user books travel package* | *34* | | *Table 21: Use case description for user cancels the booking* | *34* | | *Table 22: Use case description for user deletes the account* | *34* | | *Table 23: Use case description for user does the payment* | *35* | | *Table 24: Use case description for user searches* | *35* | | *Table 25: Use case description for user updates the account* | *35* | | *Table 26: Use case description for user [not registered user] views* | *36* | | *Table 27: Use case description for user [registered user] views* | *36* | | *Table 28: Use case description for user sends a message* | *36* | | *Table 29: Use case description for guide login into the system* | *37* | | *Table 30: Use case description for guide creates and account* | *37* | | *Table 31: Use case description for guide updates the profile* | *37* | | *Table 32: Use case description for guide deletes the account* | *38* | | *Table 33: Use case description for guide changes availability* | *38* | | *Table 34: Use case description for hotel logins into the system* | *39* | | *Table 35: Use case description for hotel creates an account* | *39* | | *Table 36: Use case description for hotel updates the account* | *39* | | *Table 37: Use case description for hotel deletes the account* | *40* | | *Table 38: Use case description for hotel changes availability* | *40* | | *Table 39: Use case description for vehicle owner logins into the system* | *41* | | *Table 40: Use case description for vehicle owner creates an account* | *41* | | *Table 41: Use case description for vehicle owner updates the account* | *41* | | *Table 42: Use case description for vehicle owner deletes the account* | *42* | | *Table 43: Use case description for vehicle owner changes availability* | *42* | | *Table 44: Components and their responsibilities* | *48* | |  |
|  |  |

# 1. Introduction

## 1.1 Domain Description

Nowadays people have different methods to organize their journeys. Most Sri Lankan people still use traditional methods to plan their journey. Considering that method, they decide the location. Then they will find the place for accommodation. After that, they will select the transportation mode for the journey. Finally, they will give telephone calls for booking the hotels or vehicle. Some people will physically visit and book the vehicle or the hotel.

Because of this method, they waste time and money. Many people have to face the following issues when they are organizing their journeys.

1. The budget of the travel plan.
2. Finding hotel booking options that are relevant to their plan and budget.
3. Finding a vehicle suitable for the number of travelers.
4. Not having enough knowledge about the places to visit.

But there is another group of people, who use the internet to plan their journeys. They will look into the given travel packages and select the most preferred one. But sometimes they have to book a hotel or a vehicle separately.

## 1.2 Proposed Solution

From our proposed website, we are providing solutions to these problems. In our website, the user has to choose the travel package selecting the category and the subcategory and just give the personal details and the number of travelers and the number of hotel rooms.

After that, the administrator receives the inquiry and he will find a hotel, vehicle, and guide for the journey. Therefore, it is very easy to go on a journey because the user does not have to worry about the bookings. The administrator of the system will arrange everything.

Another functionality that we provide is the registration of the hotels, guides, and vehicles on our website. From those registered accounts, the administrator will select the hotel, vehicle, and guide according to the location and package price.

Hotels have to change their availability by entering the available rooms regularly. The guide has to change his availability whether he is free or not and the vehicle owner has to update the availability of the vehicle. If the vehicle is available, the administrator will select the vehicle for journeys.

Users can give ratings to the hotel, guide, and vehicle used in the journey. Therefore, the quality of the service of the hotel, guide, and the vehicle will increase. From the feedbacks, the proposed system will provide easy and quality service to users.

## 1.3 Current System & It’s Limitation

Considering the current systems, they provide travel packages, but those travel packages are not categorized. Most of the current systems targeted to foreign people. If the local people want to travel around Sri Lanka, they can’t think about such packages due to the cost.

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Travellanka.com | olankatravels.com | easytravels.com |
| Categorized travel packages. |  |  |  |
| Rated travel packages based on prices.  (gold-high prices, silver-middle prices, bronze-low prices) |  |  |  |
| Registration of hotels, guides and vehicle owners to the system. |  |  |  |
| Admin assigning the hotel, guide, and vehicle according to the user requirements |  |  |  |
| Send messages and create own travel packages with the help of the administrator. |  |  |  |
| Rate the hotel, guide and vehicle by the user |  |  |  |
| User can add comments |  |  |  |
| Offer a complete travel package |  |  |  |

*Table 01: Current systems and its limitations*

## 1.4 Objectives & Goals

### 1.4.1 Project Goal

* Design and implement a comprehensive and functioning travel management system within 52 weeks.

### 1.4.2 Project Objectives

* Provide high-quality service to a user who wants to plan a journey. This will be covered by the professional team and well-designed travel packages.
* Provide an easy way to plan a journey. The user has to select a travel package and book it using the number of travelers and the types of rooms for accommodation. The administrator will find the hotel, vehicle, and guide for the journey.
* Provide facilities like accommodation, transportation, and also a guide according to the user’s needs.
* Provide facilities to register hotels, vehicles, and guides to the website. From that, the registered vendors will get the opportunity to find customers to their business.
* Establish short-term and long-term profitability by increasing users and other accounts (hotels, guides, and vehicles).

## 1.5 Assumptions, Constraints & Limitations.

### 1.5.1 Project constraints

#### 1.5.1.1 Cost

According to our proposed system, the development cost will be zero. But if we want to host the website we have to pay for the domain. In addition, we are using a test model of a payment gateway but if we want to host the website we have to pay for the payment gateway.

#### Scope

Considering the boundary activities the first point is our travel packages are predefined. Users can select the travel packages according to his preference. So the user can’t select the hotel, guide, or the vehicle according to his wish. Instead, the admin will select the hotel, guide, and vehicle according to the user requirements and the price of the travel package. And also users will be able to put comments, send messages, search travel packages, and view destinations. The administrator has the authority to add, update, and delete the travel packages, categories, subcategories, and destinations. The administrator can view the messages which are sent by users. But he uses the manual process to react to those massages. User, Hotel, Vehicle, and Guide accounts have the authority to change passwords, update and delete their accounts. The hotel can update availability by changing the number of rooms based on the room’s type daily. The Vehicle and the Guide accounts can update whether they are available or not. We are going to use a sandbox for the payments.

Considering the out of boundary activities, we are not implementing the payment gateway for the payments. And the administrator doesn’t use the system to organize the journeys. He manually contacts the available hotels, guides, and vehicles according to the user requirements. And also the administrator doesn’t use the system to react to the user messages.

#### 1.5.1.3 Quality

From our proposed website users can give ratings for the various vendors. Through that, the administrator can get an idea about the quality of the service which is given by the website.

#### Customer Satisfaction

With the development of the project, we have to implement a user-friendly environment. When considering the working website, the administrator has a responsibility to provide good service to the user.

#### Risk

When the development of the project we have to submit the project in the given period of the time. When considering the working website, the administrator will face some risks of the conditions of the hotel and the vehicle. The weather condition will be an issue sometimes as well.

#### Resources

For the development of the proposed system, we are using our laptops, freely available tools, and open-source software. There will be four members to develop the proposed system.

#### 1.5.1.7 Time

We have only one year to do the proposed system. In that period we have to identify the problem, gather requirements, system designing, system implementation, system testing, and the maintenance of the system.

### Assumptions

During the development of the proposed system, we have all the resources that we need to develop the proposed website. For example, internet connection, laptops, the knowledge of the software that we are going to use. We assume that all our equipment is in good condition. And we assume that our team members have all the required skills.

Considering the working website, the administrator will assume that all the vehicles are in good condition. As well he will assume that all the hotels have good service and facilities. And also the administrator will assume that the weather condition will not be an issue for the journey.

Considering the technology, we assume that we have all the required technology and all the members learned about those technologies. If someone has a problem with some technologies, he will learn those within this period.

Considering the finance of the proposed system, we assume that the development cost will be zero for the project.

We assume that once we complete the proposed system, we are going to host the website.

### Limitation

The proposed system will require an internet connection to work. So it is unstable to have access to it in an area where there is no internet network.

The system will be developed by using the English language. It does not work in local languages.

There will be very high data interactions during the tourist season. Because of that the system might be down. Moderate security needed for this system. It is a step by step procedure as an example all steps come after one another. So every step is needed to be completed, then only you can move the other steps. You need to wait as well.

Considering the proposed system’s limitations, the user has to register before booking a travel package. The hotel, guide, and vehicle accounts should update their availability regularly. Considering the hotel account, the hotel has to update the availability by entering the number of available rooms based on its type, single rooms, double rooms, and family rooms. And also the hotel account has to update the availability daily. The vehicle and the guide accounts have to update whether they are available or not as well. If they are not changing their availability, Easy Travel Company can’t guarantee to select that account for the journeys.

At the registration process, the user has to agree to the terms and conditions of the Easy Travels Company. The hotel, guide, and vehicle accounts will also have to agree on the terms and conditions. If they are not working according to the agreement, Easy Travel Company will neglect that account.

Considering the travel packages we will only provide three types of travel packages called gold, silver, and bronze. That means the price of those packages will be high, moderate, and low. So based on the selected travel package, the administrator will assign the hotel, the guide, and the vehicle for the journey.

# 2. Feasibility Study

From this feasibility study, we are analyzing whether our software meets the requirements. We used this feasibility study to determine whether our software can be implemented using current technology and within a specified budget and schedule as well. And also we are doing this feasibility study to find whether our software can be integrated with other existing software.

We make this feasibility study covering five areas, technical feasibility, economic feasibility legal and ethical feasibility, operational feasibility, and schedule feasibility.

## Technical Feasibility

Considering our project it is possible with current technologies. As programming languages, we are going to use HTML, CSS, JavaScript as front-end programming languages. For back-end programming languages we are going to use PHP and MySQL Database. For the PHP software requirements, we are going to use the WAMP server [Microsoft Windows o/s, Apache MySQL PHP]. As the MySQL administrator tool, we are going to use phpMyAdmin. As a diagram drawing tool, we are going to use drow.io. For documentation tools, we are going to use Microsoft Word and Microsoft Excel. For Project management software we are going to use Microsoft Project (To create Gantt chart and timeline). As text editing tools we are going to use Visual Studio Code and Notepad++. For the contribution platform, we are going to use GitHub.

Each of these technologies are freely available and the technical skills required are manageable. Front-end tools like HTML, CSS, JavaScript are easily compatible with our current hardware configuration. Back-end language, PHP is stable as compared to other programming languages and platform-independent. PHP is easily connected with the database and makes a secure connection with the database. And also the PHP unit is used to perform the unit testing quickly and easily. Considering the webserver technology, Apache is a proven, and reliable tool. Considering the database, MySQL is easily accessible. According to web technology, Apache we can use the WAMP web development stack. If we want to expand our system with more functionalities and features. The front-end and back-end technologies provide the accuracy, reliability, and ease of access and data security.

## Economic feasibility

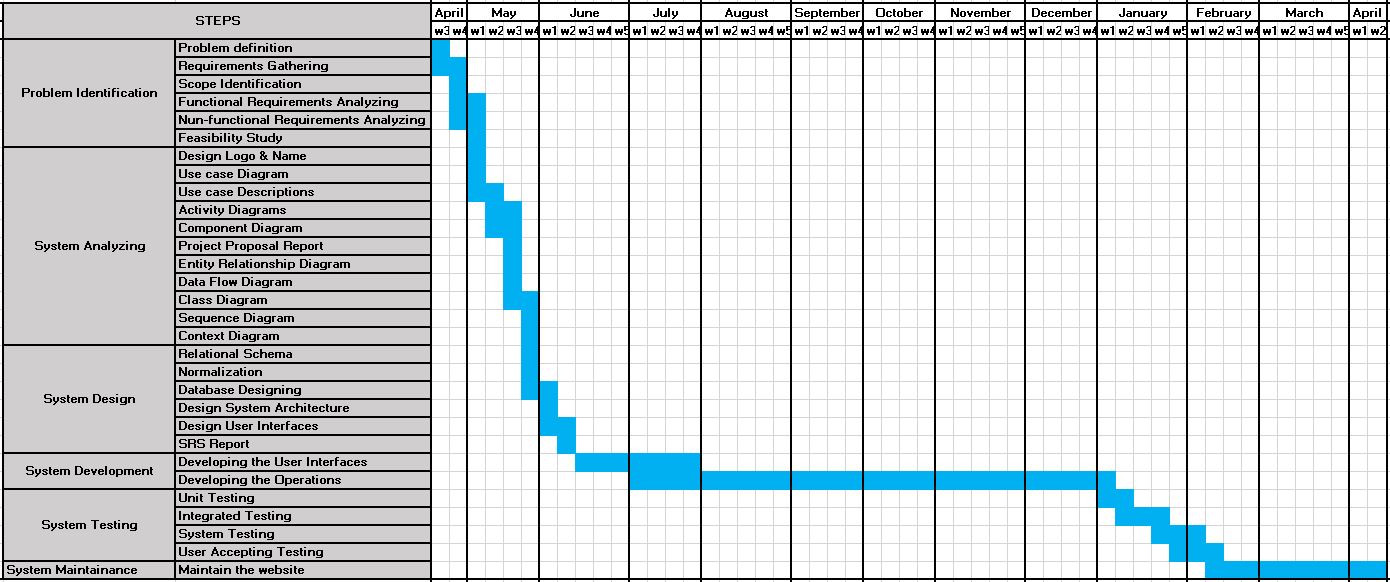
Our development cost for the project will be zero because we use open source software and freely available tools. Now we will use the test mode of a payment gateway but when the website hosts we have to pay for the payment gateway. The administrator should know to maintain the system. If he is not, then the training cost will be added as well. The proposed website provides benefits to the administrator. He will save his time and money because when he arranges the journey he will find everything in the same place such as finding a hotel, guide, and vehicle. And also considering the value of the working project, it will provide economic benefits to the administrator because he will receive a commission from the hotel, guide, and the vehicle owner who is engaging with travel packages. When comparing the existing websites they are not providing facilities like ours. From the many users that will engage with our website, the administrator will get money.

## Legal and Ethical feasibility

When considering the project like ours' legal aspects have a minor effect on the feasibility study. Considering we are using freely available tools and open-source software our project doesn’t include legal aspects including software licenses. Our system will not save the users’ sensitive data like credit card information. Ethically we have to consider how our group members think about the project. And also those ethical points should include how we work together and how we interact with other members of the group.

## Schedule feasibility

The project started on the 20th of April 2020. We have to fulfill the requirements and implement our project within one year. We have four members in the group, so we can separate our works and implement our project on time. Everyone will be a developer as well as a tester. So that will reduce the time that we are taking to implement the project. As we can include alternative timelines like holidays to do those tasks. We are planning our works according to the following steps.



*Figure 01: Schedule feasibility graph*

## Operational feasibility

When we consider our project it is a really good solution for current requirements. We surveyed to find the problems and how people plan their journey. So it is clear that our solution for travel management will fulfill the requirements of the user. The user has to book a hotel and a vehicle for their journey. According to the survey, it is not easy to find those needs. So from our system, they have to only spend a few minutes to find a travel package and enter their details. The administrator will find all the other needs of them. When considering the proposed system, it will provide more benefits. Usually the user has to pay the cost for accommodation, vehicle or the guide separately. But in our solution, the cost will be based on the travel package. So it is economically beneficial for the user.

Not only the user, the other accounts hotel, guide, and the vehicle will also get benefits. If they are hiring themselves that will be a big cost more than they registered on this website. Because they need to have a method like a website to get hired. So the only thing they have to do is change their availability on this website. So they will get advantages from our website.

# 3. Requirements

## 3.1 Stakeholders

In our system there are five stakeholders.



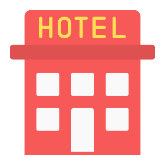
1. User

The user’s main task is to book travel packages. A person who is not registered to the system, will not be able to book travel packages. So if he wants to book a travel package he has to register to the website. He can put comments regarding the travel packages. He can rate the hotel, the vehicle, and the guide based on his journey as well.



1. Admin

The admin’s main role is to maintain the system. He is responsible for creating packages and organizing the journeys according to the user requirements. He has to use a manual way to contact the hotel, the guide, and the vehicle using the available list which is provided by the system. Before he selects those, he has to consider the price of the package and the user requirements.

1. Hotel

The hotel’s main task is maintaining the profile. He has to update the availability based on the room’s type. So he has to change this daily because the availability might change day by day. So it is better to update availability daily. He has to update the profile as well.



1. Guide

Guide’s main task is maintaining his profile. He has to update whether he is available or not. He has to update the profile as well. Based on the price and the description he will find the opportunities through this website.



1. Vehicle owner

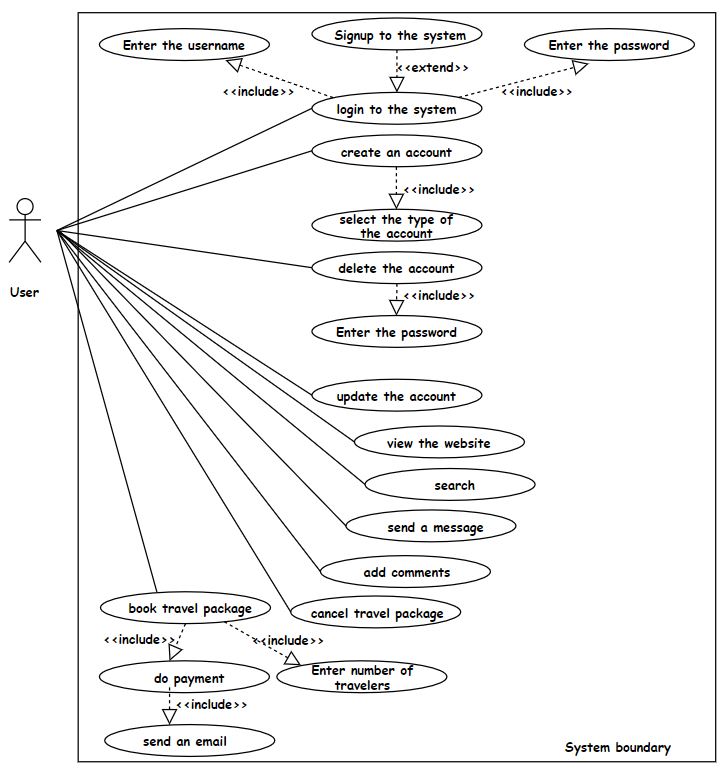
The vehicle owner’s main task is maintaining his profile. He has to update whether the vehicle is available or not. He has to update the profile as well. Based on the price of the vehicle he will find the opportunities through this website.

## 3.2 Use case diagram

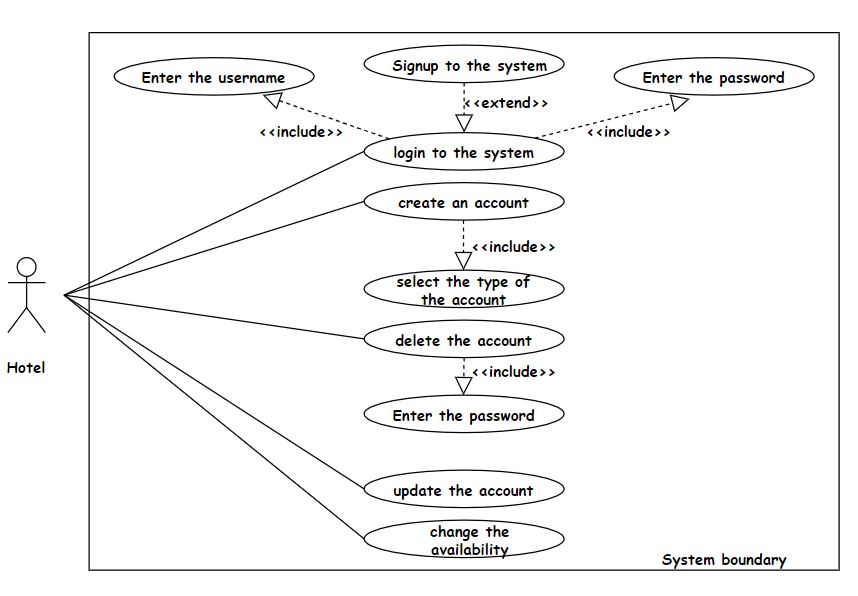
*Figure 02: Use-case diagram*

### 

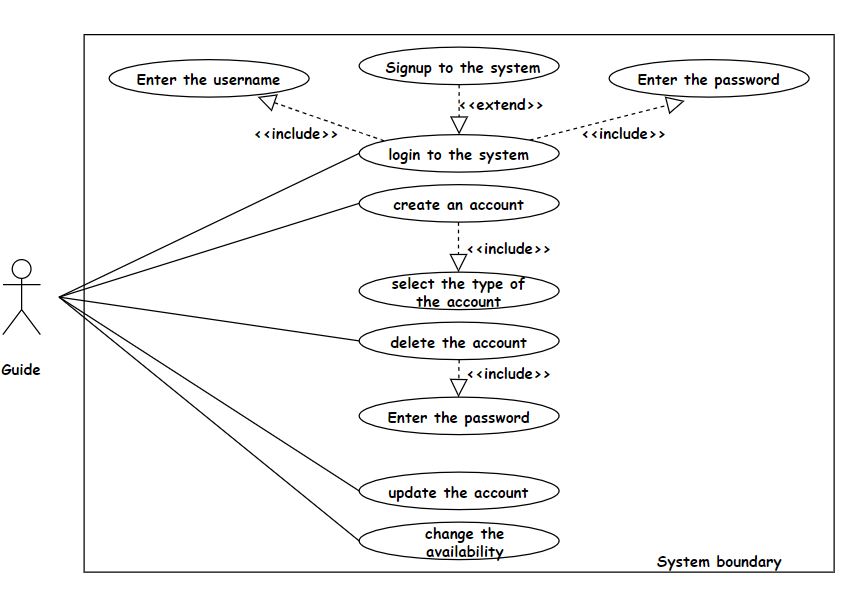
### 3.3.1 Actor wise use cases

1. User

*Figure 03: Use case Diagram for the user*

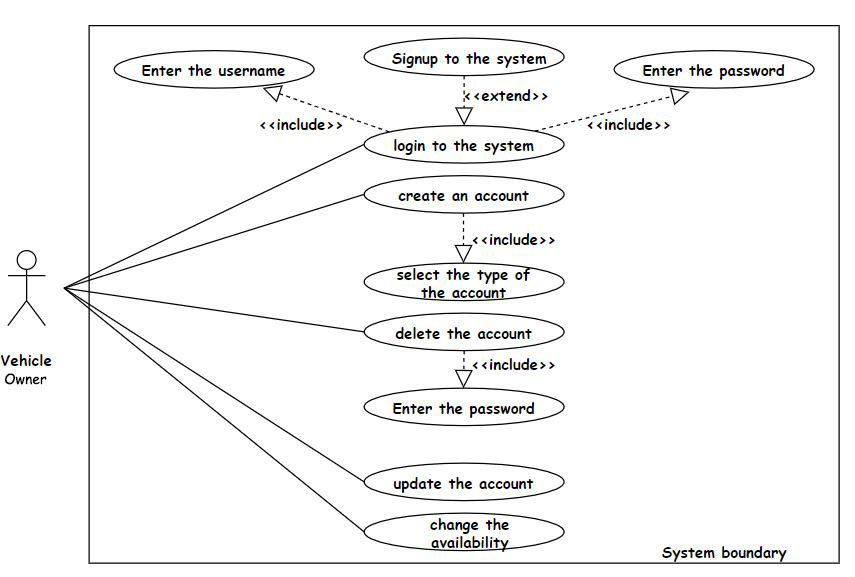
1. Hotel

*Figure 04: Use case Diagram for the hotel*

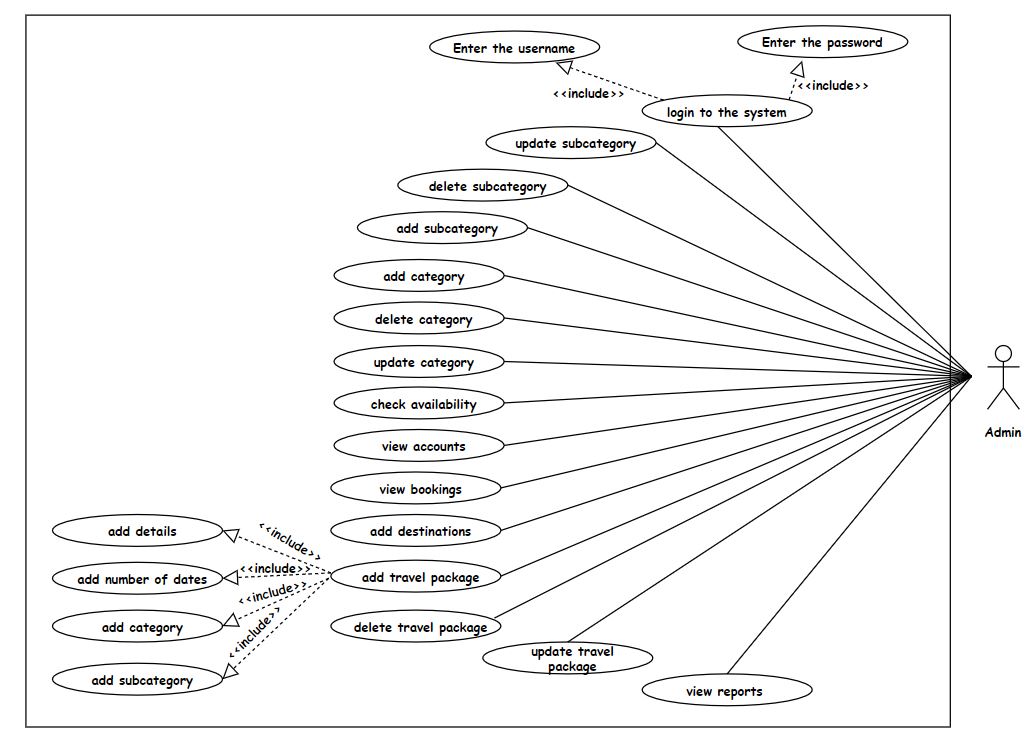
1.  Guide

*Figure 05: Use case Diagram for the guide*

1. Vehicle owner



*Figure 06: Use case Diagram for the vehicle owner*

1. Admin

*Figure 07: Use case Diagram for the admin*

## 3.3 Use case Descriptions

### 3.2.1 Use case descriptions for Admin’s functionalities.

|  |  |
| --- | --- |
| Use case | Login to the system |
| Summary | Admin logins into the system. |
| Actors | Admin |
| Preconditions | Admin should not be logged in to the system. |
| Description | Admin should enter the username and the password to login to the system. |
| Exceptions | Admin’s username and the password not in the system.  Admin already logs in to the system. |
| Post conditions | Admin logged in to the system.  Admin views the home page |

*Table 02: Use case description for admin logins into the system*

|  |  |
| --- | --- |
| Use case | View accounts |
| Summary | Admin views accounts’ profiles. |
| Actors | Admin |
| Preconditions | Admin should be logged in to the system by using the username and the password.  Admin should select the view profile and select the type of the accounts. |
| Description | Admin will see a list of accounts. Then admin can view the accounts’ profiles using their ids or names. From selecting the name of the account, the admin will be able to see more details of the corresponding account. |
| Exceptions | The selected profile does not exist in the system. |
| Post conditions | Admin will view the selected profile. |

*Table 03: Use case description for admin views accounts’ profiles.*

|  |  |
| --- | --- |
| Use case | View bookings |
| Summary | Admin views booking list. |
| Actors | Admin |
| Preconditions | Admin should be logged in to the system by using the username and the password.  Admin should select the view bookings. |
| Description | Admin can view the booking list. Then admin can enter booking id or date to find the booking. From that admin will get more details about the user and he will get the number of days and the number of people who will join the journey. |
| Exceptions | There is no booking available. |
| Post conditions | Admin will view the booking list and the more details of the selected booking. |

*Table 04: Use case description for admin views booking list.*

|  |  |
| --- | --- |
| Use case | View reports |
| Summary | Admin views the reports. |
| Actors | Admin |
| Preconditions | Admin should be logged in to the system by using the username and the password.  Admin should select the view reports. |
| Description | Admin can view the reports like the number of travelers for a travel package, best hotels, best guides, the best vehicle, best travel package, etc. First admin has to select the report. After that he will see the report’s details. |
| Exceptions | There are no details to generate reports. |
| Post conditions | Admin will view the selected report’s details. |

*Table 05: Use case description for admin views the reports.*

|  |  |
| --- | --- |
| Use case | Add travel package |
| Summary | Admin adds a new travel package to the system |
| Actors | Admin |
| Preconditions | Admin should be logged in to the system by using the username and the password.  Admin should select the add travel package. |
| Description | Admin can add a new travel package to the system by choosing category and subcategory. Admin has to enter details of the travel plan and the number of days. As well the admin has to enter the price of the travel package for a single person. |
| Exceptions | Admin is not logged in to the system. |
| Post conditions | A message of “successfully added” will appear. |

*Table 06: Use case description for admin adds a travel package.*

|  |  |
| --- | --- |
| Use case | Add category |
| Summary | Admin adds the categories of the travel package to the system |
| Actors | Admin |
| Preconditions | Admin should be logged in to the system by using the username and the password.  Admin should select the add category. |
| Description | Admin can add a new category for the travel package by entering its name. Then the system will automatically give an id number to that category. |
| Exceptions | The added category exists in the system. |
| Post conditions | A message of “successfully added” will appear. |

*Table 07: Use case description for admin adds a category*

|  |  |
| --- | --- |
| Use case | Add subcategory |
| Summary | Admin adds the subcategories of the travel package to the system |
| Actors | Admin |
| Preconditions | Admin should be logged in to the system by using the username and the password.  Admin should select the add subcategory. |
| Description | First admin should have selected the category of the new subcategory. Then admin can add a new subcategory for the travel package by using its name. Then the system will automatically give an id number to that subcategory. |
| Exceptions | The added subcategory exists in the system. |
| Post conditions | A message of “successfully added” will appear. |

*Table 08: Use case description for admin adds a subcategory*

|  |  |
| --- | --- |
| Use case | Add destinations |
| Summary | Admin adds the destination into the system. |
| Actors | Admin |
| Preconditions | Admin should be logged in to the system by using the username and the password.  Admin should select the add destinations. |
| Description | Admin can add a new destination including the details and photos.  Then the system will automatically give an id number to that destination. |
| Exceptions | The added destination name exists in the system. |
| Post conditions | A message of “successfully added” will appear. |

*Table 09: Use case description for admin adds a destination*

|  |  |
| --- | --- |
| Use case | Check availability |
| Summary | Admin checks the availability of hotels, guides, and vehicles |
| Actors | Admin |
| Preconditions | Admin should be logged in to the system by using the username and the password.  Admin should select view availability of the wanted account type. |
| Description | Admin can view a list of registered accounts according to the selected account type. In this list, the admin can view the available accounts. Then admin can get more details by selecting one of those accounts. |
| Exceptions | There are no registered accounts of the selected account types. |
| Post conditions | The list of selected and available accounts will be displayed. |

*Table 10: Use case description for admin checks availability*

|  |  |
| --- | --- |
| Use case | Delete a category |
| Summary | Admin deletes a category. |
| Actors | Admin |
| Preconditions | Admin should be logged in to the system by using the username and the password.  Admin should select the delete category. |
| Description | Admin will delete a category selecting its id number. Then he will see the details of the category. Then there will be a confirmation of the deletion again. After the admin will confirm the deletion the category will be deleted. |
| Exceptions | The category id is not in the system. |
| Post conditions | A message of “successfully deleted” will appear. |

*Table 11: Use case description for admin deletes a category*

|  |  |
| --- | --- |
| Use case | Delete a subcategory |
| Summary | Admin deletes a subcategory. |
| Actors | Admin |
| Preconditions | Admin should be logged in to the system by using the username and the password.  Admin should select the delete subcategory. |
| Description | Admin will delete a subcategory selecting its id number. Then he will see the details of the subcategory. Then there will be a confirmation of the deletion again. After admin will confirm the deletion the subcategory will be deleted. |
| Exceptions | The subcategory id is not in the system. |
| Post conditions | A message of “successfully deleted” will appear. |

*Table 12: Use case description for admin deletes a subcategory*

|  |  |
| --- | --- |
| Use case | Delete a travel package |
| Summary | Admin deletes the travel package. |
| Actors | Admin |
| Preconditions | Admin should be logged in to the system by using the username and the password.  Admin should select the delete travel package. |
| Description | Admin will delete a travel package selecting its id number. Then he will see the details of the travel package. Then there will be a confirmation of the deletion again. After the admin will confirm the deletion the travel package will be deleted. |
| Exceptions | The travel package id is not in the system. |
| Post conditions | A message of “successfully deleted” will appear. |

*Table 13: Use case description for admin deletes a travel package*

|  |  |
| --- | --- |
| Use case | Update category |
| Summary | Admin updates the category. |
| Actors | Admin |
| Preconditions | Admin should be logged in to the system by using the username and the password.  Admin should select the update category. |
| Description | Admin updates the category name by selecting its id. |
| Exceptions | The updated category does not exist in the system. |
| Post conditions | A message of “successfully updated” will appear. |

*Table 14: Use case description for admin updates a category*

|  |  |
| --- | --- |
| Use case | Update subcategory |
| Summary | Admin updates the subcategory. |
| Actors | Admin |
| Preconditions | Admin should be logged in to the system by using the username and the password.  Admin should select the update subcategory. |
| Description | Admin updates the subcategory name by selecting its id. |
| Exceptions | The update subcategory does not exist in the system. |
| Post conditions | A message of “successfully updated” will appear. |

*Table 15: Use case description for admin updates a subcategory*

|  |  |
| --- | --- |
| Use case | Update travel package |
| Summary | Admin updates the travel package. |
| Actors | Admin |
| Preconditions | Admin should be logged in to the system by using the username and the password.  Admin should be selected the update travel package. |
| Description | Admin updates the travel package's details selecting its id. Then he can change details about the travel package. |
| Exceptions | The updated travel package does not exist in the system. |
| Post conditions | A message of “successfully updated” will appear. |

*Table 16: Use case description for admin updates a travel package*

### 

### 3.2.2 Use case descriptions for User’s functionalities.

|  |  |
| --- | --- |
| Use case | Log in to the system |
| Summary | User logins into the system |
| Actors | User |
| Preconditions | The user should not be logged in to the system by using the username and the password.  The user should select the login. |
| Description | The user will have to enter the username and the password to login into the system. |
| Exceptions | The user is not registered in the system.  The user already logs into the system. |
| Post-conditions | User logged in to the system |

*Table 17: Use case description for user logins into the system*

|  |  |
| --- | --- |
| Use case | Create an account |
| Summary | User registers to the system using the account type of user. |
| Actors | User |
| Preconditions | The user should not have an account in the system by using the username and the password.  The user has to select the account type as the “user”. |
| Description | User will get a form to fill his information for registration to the website. After he submitted the details he will be a registered user in the system. |
| Exceptions | The user has an account with the same details like email. |
| Post-conditions | User will get the home page. |

*Table 18: Use case description for user creates an account*

|  |  |
| --- | --- |
| Use case | Add comments |
| Summary | The user adds comments about the travel package. |
| Actors | User |
| Preconditions | The user should be logged in to the system by using the username and the password.  User should select the view profile and add comments. |
| Description | The user will comment on the travel packages and add photos. There is an option to give ratings for the hotel, vehicle, and the guide as well. |
| Exceptions | The user is not logged in to the system. |
| Post-conditions | The user can input the comments to the system. |

*Table 19: Use case description for user adds comments*

|  |  |
| --- | --- |
| Use case | Book travel package |
| Summary | User books a travel package. |
| Actors | User |
| Preconditions | The user should be logged in to the system by using the username and the password. |
| Description | First user has to select the category and the subcategory of the package. Then he will find travel packages under the subcategory. Pressing the “Book Now” button, the user can view a form to fill. After entering the details he will be able to book a travel package. |
| Exceptions | The user is not logged in to the system. |
| Post-conditions | The user will see the payment options for the booked travel package. |

*Table 20: Use case description for user books travel package*

|  |  |
| --- | --- |
| Use case | Cancel the booking |
| Summary | The user cancels the booking which he made before. |
| Actors | User |
| Preconditions | The user should be logged in to the system by using the username and the password.  The date of cancelation should be within two dates starting the booking date.  The user should select the view profile and view bookings. |
| Description | User can view the booking which had done by him. Then he will click the name and he cancel the booking by clicking the “cancel” button. After confirming the cancelling the booking will be canceled. |
| Exceptions | The user is not logged in to the system.  User not book a travel package.  The cancelation date is after two dates of the booking date. |
| Post-conditions | The user gets a message for canceling the booking. |

*Table 21: Use case description for user cancels the booking*

|  |  |
| --- | --- |
| Use case | Delete the account |
| Summary | User can delete his account which includes his details from the system. |
| Actors | User |
| Preconditions | The user should be logged in to the system by using the username and the password.  The user should select the view profile. |
| Description | The user can delete his account by clicking the “delete profile” button. After that he has to enter the password of his account. Then there will be a confirmation. After confirming that he will be able to delete the account from the system. |
| Exceptions | The user is not logged in to the system. |
| Post-conditions | The user gets a message for deleting his account. |

*Table 22: Use case description for user deletes the account*

|  |  |
| --- | --- |
| Use case | Do payment |
| Summary | User pay for the booking travel package. |
| Actors | User |
| Preconditions | The user should be logged in to the system by using the username and the password.  The user should book a travel package before the payment. |
| Description | User can pay for the booking travel package using his bank details. |
| Exceptions | The user is not logged in to the system.  The user is not booked a travel package. |
| Post-conditions | The user gets a message for payment. |

*Table 23: Use case description for user does the payment*

|  |  |
| --- | --- |
| Use case | Search |
| Summary | User search travel packages. |
| Actors | User |
| Preconditions | The user should go to the website it doesn't matter whether he logs in to the system or not. |
| Description | The user search travel packages by entering the number of days, places, or the price of the travel package. |
| Exceptions | User is not on the home page. |
| Post-conditions | User will see the filtered packages |

*Table 24: Use case description for user searches*

|  |  |
| --- | --- |
| Use case | Update the account |
| Summary | The user updates the profile. |
| Actors | User |
| Preconditions | The user should be logged in to the system by using the username and the password.  The user should select the view profile. |
| Description | The user updates his profile by clicking the “update profile” button. After that user will see the details of the profile and he will be able to update details. |
| Exceptions | The user is not logged in to the system. |
| Post-conditions | User will get a message for updating. |

*Table 25: Use case description for user updates the account*

|  |  |
| --- | --- |
| Use case | View [not registered user] |
| Summary | The user views the website. |
| Actors | User |
| Preconditions | The user should not be logged in to the system by using the username and the password. |
| Description | The user will see the options to view the packages by selecting category and the subcategory, view comments, view destinations, and view about us. |
| Exceptions | The user is logged in to the system. |
| Post-conditions | User will see the viewed item. |

*Table 26: Use case description for user [not registered user] views*

|  |  |
| --- | --- |
| Use case | View [registered user] |
| Summary | The user views the website. |
| Actors | User |
| Preconditions | The user should be logged in to the system. |
| Description | The user will see the options to view the packages by selecting category and the subcategory, view comments, view destinations, view profile, and view about us. |
| Exceptions | The user is not logged in to the system. |
| Post-conditions | User will see the viewed item. |

*Table 27: Use case description for user [registered user] views*

|  |  |
| --- | --- |
| Use case | Send message |
| Summary | The user sends messages to the administrator regarding something. |
| Actors | User |
| Preconditions | The user should be on the home page. It doesn’t matter whether the user is registered or not.  The user should select the contact page. |
| Description | User will see a form and he will enter his contact details and the message. After that he has to click on the “send” button. |
| Exceptions | The user is not on the website. |
| Post-conditions | User will get a message for sending. |

*Table 28: Use case description for user sends a message*

### 3.2.3 Use case descriptions for Guide’s functionalities.

*Table 29: Use case description for guide login into the system*

*Table 30: Use case description for guide creates and account*

|  |  |
| --- | --- |
| Use case | Log in to the system |
| Summary | Guide logins into the system |
| Actors | Guide |
| Preconditions | The guide should not be logged in to the system.  The guide should select the login. |
| Description | The guide will have to enter the username and the password to login to the system. |
| Exceptions | Guide’s username and the password not in the system.  The guide already logs in to the system. |
| Post-conditions | Guide logged in to the system. |

|  |  |
| --- | --- |
| Use case | Create an account |
| Summary | Guide registers to the system using the account type of “guide”. |
| Actors | Guide |
| Preconditions | The guide should not have an account in the system.  The guide should be selected the account type as “guide” |
| Description | The guide will get a form to fill his information for registration to the website. |
| Exceptions | The guide has an account with the same details like email. |
| Post-conditions | Guide views the home page. |

|  |  |
| --- | --- |
| Use case | Update the account |
| Summary | Guide update the profile. |
| Actors | Guide |
| Preconditions | The guide should be logged in to the system by using the username and the password.  The guide should select the view profile. |
| Description | The guide updates his profile by clicking the “update profile” button. After that the guide will see the details of the profile and he will be able to update details. |
| Exceptions | The guide is not logged in to the system. |
| Post-conditions | The guide will get a message for updating. |

*Table 31: Use case description for guide updates the profile*

|  |  |
| --- | --- |
| Use case | Delete the account |
| Summary | The guide can delete his account which includes his details from the system. |
| Actors | Guide |
| Preconditions | The guide should be logged in to the system by using the username and the password.  The guide should select the view profile. |
| Description | The guide can delete his account by clicking the “delete profile” button. After that he has to enter the password of his account. Then there will be a confirmation. After confirming that he will be able to delete the account from the system. |
| Exceptions | The guide is not logged in to the system. |
| Post-conditions | Guide gets a message for deleting his account. |

*Table 32: Use case description for guide deletes the account*

|  |  |
| --- | --- |
| Use case | Change availability |
| Summary | The guide can input whether he is available or not available. |
| Actors | Guide |
| Preconditions | The guide should be logged in to the system by using the username and the password.  The guide should be selected the view profile. |
| Description | The guide will see an option as “change availability”. After clicking that the guide will be able to change the availability. |
| Exceptions | The guide is not logged in to the system. |
| Post-conditions | The guide will see a message of changing the availability. |

*Table 33: Use case description for guide changes availability*

### 3.2.4 Use case descriptions for the hotel's functionalities.

|  |  |
| --- | --- |
| Use case | Log in to the system |
| Summary | Hotel logins into the system |
| Actors | Hotel |
| Preconditions | The hotel should not be logged in to the system.  The hotel should select the login. |
| Description | The hotel will have to enter the username and the password to login to the system. |
| Exceptions | Hotel’s username and the password not in the system.  The hotel already logs in to the system. |
| Post-conditions | Hotel logged in to the system. |

*Table 34: Use case description for hotel logins into the system*

|  |  |
| --- | --- |
| Use case | Create an account |
| Summary | Hotel registers to the system using the account type of “hotel”. |
| Actors | Hotel |
| Preconditions | The hotel should not have an account in the system.  The hotel should be selected the account type as “hotel ” |
| Description | The hotel will get a form to fill his information for register to the website. |
| Exceptions | The hotel has an account with the same details like email. |
| Post-conditions | The hotel views the home page. |

*Table 35: Use case description for hotel creates an account*

|  |  |
| --- | --- |
| Use case | Update the account |
| Summary | Hotel updates the profile. |
| Actors | Hotel |
| Preconditions | The hotel should be logged in to the system by using the username and the password.  The hotel should select the view profile. |
| Description | The hotel updates his profile by clicking the “update profile” button. After that the hotel will see the details of the profile and he will be able to update details. |
| Exceptions | The hotel is not logged in to the system. |
| Post-conditions | The hotel will get a message for updating. |

*Table 36: Use case description for hotel updates the account*

|  |  |
| --- | --- |
| Use case | Delete the account |
| Summary | The hotel can delete his account which includes his details from the system. |
| Actors | Hotel |
| Preconditions | The hotel should be logged in to the system by using the username and the password.  The hotel should select the view profile. |
| Description | The hotel can delete his account by clicking the “delete profile” button. After that he has to enter the password of his account. Then there will be a confirmation. After confirming that he will be able to delete the account from the system. |
| Exceptions | The hotel is not logged in to the system. |
| Post-conditions | The hotel gets a message for deleting his account. |

*Table 37: Use case description for hotel deletes the account*

|  |  |
| --- | --- |
| Use case | Change availability |
| Summary | The hotel can input whether he is available or not available. |
| Actors | Hotel |
| Preconditions | The hotel should be logged in to the system by using the username and the password.  The hotel should select the view profile. |
| Description | The hotel will see an option as “change availability”. After clicking that the hotel will be able to change the availability. Hotel has to enter the available number of rooms according to its type. |
| Exceptions | The hotel is not logged in to the system. |
| Post-conditions | The hotel will see a message of changing the availability. |

*Table 38: Use case description for hotel changes availability*

### 3.2.5 Use case descriptions for vehicle owner’s functionalities.

*Table 39: Use case description for vehicle* owner *logins into the system*

|  |  |
| --- | --- |
| Use case | Create an account |
| Summary | Vehicle owner registers to the system using the account type of “vehicle”. |
| Actors | Vehicle owner |
| Preconditions | The vehicle owner should not have an account in the system.  The vehicle owner should select the account type as “vehicle ” |
| Description | The vehicle owner will get a form to fill his information for registration to the website. |
| Exceptions | The vehicle owner has an account with the same details like email. |
| Post-conditions | The vehicle owner views the home page. |

*Table 40: Use case description for vehicle* owner *creates an account*

|  |  |
| --- | --- |
| Use case | Update the account |
| Summary | The vehicle owner updates the profile. |
| Actors | Vehicle owner |
| Preconditions | The vehicle should be logged in to the system by using the username and the password.  The vehicle owner should select the view profile. |
| Description | The vehicle owner updates his profile by clicking the “update profile” button. After that vehicle owner will see the details of the profile and he will be able to update details. |
| Exceptions | The vehicle owner is not logged in to the system. |
| Post-conditions | The vehicle owner will get a message for updating. |

|  |  |
| --- | --- |
| Use case | Log in to the system |
| Summary | Vehicle owner logins into the system |
| Actors | Vehicle owner |
| Preconditions | The vehicle owner should not be logged in to the system.  The vehicle owner should select the login. |
| Description | The vehicle owner will have to enter the username and the password to login to the system. |
| Exceptions | Vehicle owner’s username and the password not in the system.  The vehicle owner already logs in to the system. |
| Post-conditions | The vehicle owner logged in to the system. |

*Table 41: Use case description for vehicle* owner *updates the account*

|  |  |
| --- | --- |
| Use case | Delete the account |
| Summary | The vehicle owner can delete his account which includes his details from the system. |
| Actors | Vehicle owner |
| Preconditions | The vehicle owner should be logged in to the system by using the username and the password.  The vehicle should select the view profile. |
| Description | The vehicle owner can delete his account by clicking the “delete profile” button. After that he has to enter the password of his account. Then there will be a confirmation. After confirming that he will be able to delete the account from the system. |
| Exceptions | The vehicle owner is not logged in to the system. |
| Post-conditions | The vehicle owner gets a message for deleting his account. |

*Table 42: Use case description for vehicle* owner *deletes the account*

|  |  |
| --- | --- |
| Use case | Change availability |
| Summary | The vehicle owner can input whether he is available or not available. |
| Actors | Vehicle owner |
| Preconditions | The vehicle owner should be logged in to the system by using the username and the password.  The vehicle owner should select the view profile. |
| Description | The vehicle owner will see an option as “change availability”. After clicking that the vehicle owner will be able to change the availability. |
| Exceptions | The vehicle owner is not logged in to the system. |
| Post-conditions | The vehicle owner will see a message of changing the availability. |

*Table 43: Use case description for vehicle* owner *changes availability*

## 3.4 Functional Requirement Identification

According to our proposed system, we can find the following requirements as functional requirements.

1. User Requirements

Consider the user, users must register to the system by creating an account before he booking a travel package. He has to enter the username and password to login to the system as well. And also users can search for information about travel packages by entering the number of days, price of the package, or location name. Users can view travel packages with separating categories and subcategories. And also users can view the destinations. There will be an option to select the destination that the user wants to select. From that user will get the details of that destination and the travel packages which include that destination. Users can view his profile. He can update his profile and delete his profile as well. Users can add comments. From the comment section user will be able to give ratings for hotel, guide, and vehicle. Users can book travel packages with entering details and also he can pay for the booked travel package. Users can cancel the bookings within two days. Users can send messages to admin, regarding issues.

Considering the hotel, the hotel must register to the system by creating an account. And also he will log in to the website by using his username and password. Hotels can view his profile and update his profile. If the hotel wants to delete their accounts he can delete his profiles. Hotels can change the availability by entering the number of available rooms based on room type as well.

Considering the guide, he must register to the system by creating an account. He will log in to the website by using his username and password as well. The guide can view their profile and update the profile. If the guide wants to delete the profile he can delete his profile. The guide can change his availability.

Considering the vehicle Owner he must register to the system by creating an account. And also will log in to the website by using his username and password. The vehicle owner can view his profile and update his profile. If the vehicle owner wants to delete the profile he can delete his profile. The vehicle owner can change the availability of the vehicle.

1. System Requirements

The system can be able to store the details of bookings. And also the system should have different home pages for different user account types. The system will be able to generate reports. The system should be able to add, update, and delete travel packages as well. The system should be able to add, update, and delete categories and subcategories. The system should be able to add a destination. And also the system should be able to search for accounts and view accounts.

## 3.5 Non-Functional Requirements

### 3.5.1 Identification of Quality Attributes

1. Security
2. Performance
3. Usability
4. User-friendliness
5. Portability
6. Maintainability

### 3.5.2 Suggestion to achieve Quality Attributes

1. Security

Considering the security, the password is really important. So the required passwords should be longer than 12 characters. There will be a password recovery option. The password will be saved on the database in an encrypted manner. The forms are validated with the required fields as well. Considering the data backup option, we are going to use phpMyAdmin as the database. So it has an option to export data from the database.

1. Performance

We can improve the performance of the website by speeding the process, decrease the response time, etc. We use the PHP programming language for back-end development. So because of that applications can be easily loaded over the slow internet and data speed. And also we can use minify and combine files to reduce the size of each file as well as the total number of files. From that system's performance will be improved.

1. Usability

At the testing phase, we are planning to give our website to selected users to check whether it is usable or not. From this testing, we can identify what the user thinks about our website.

1. User-friendliness

The user will find travel packages very easily because all the packages are divided into categories and subcategories. There will be travel packages based on prices so it will really help the user as well. He can select the travel package according to his budget. Considering the clean and clear user interfaces, it will help the user to access the website. As mentioned before, we are planning to have user testing. So we can get an idea about how they feel about the user-friendliness of our website.

1. Portability

We are planning to implement our website responsively. So it will compatible with operating systems and also the mobiles. So the user can use any device to see a good output.

1. Maintainability

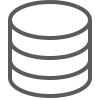
We will use readable codes that can be easily understood. And also we are going to use comments to identify the source codes. So code can be easily changed. We are going to use automated validation parts as well.

# 4. Proposed System’s Architecture

## 4.1 System’s Architecture

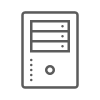


**Payment Gateway**



**PHP scripting engine**

**Persistence**



**Merchant Bank**



**User**

**Hotel**

**Vehicle owner**

**Guide**

**Admin**

**Customer Bank**



**Credit card network**

**Merchant Account**



**Subcategory**

**Reports**

**Packages**

**Security**

**Category**

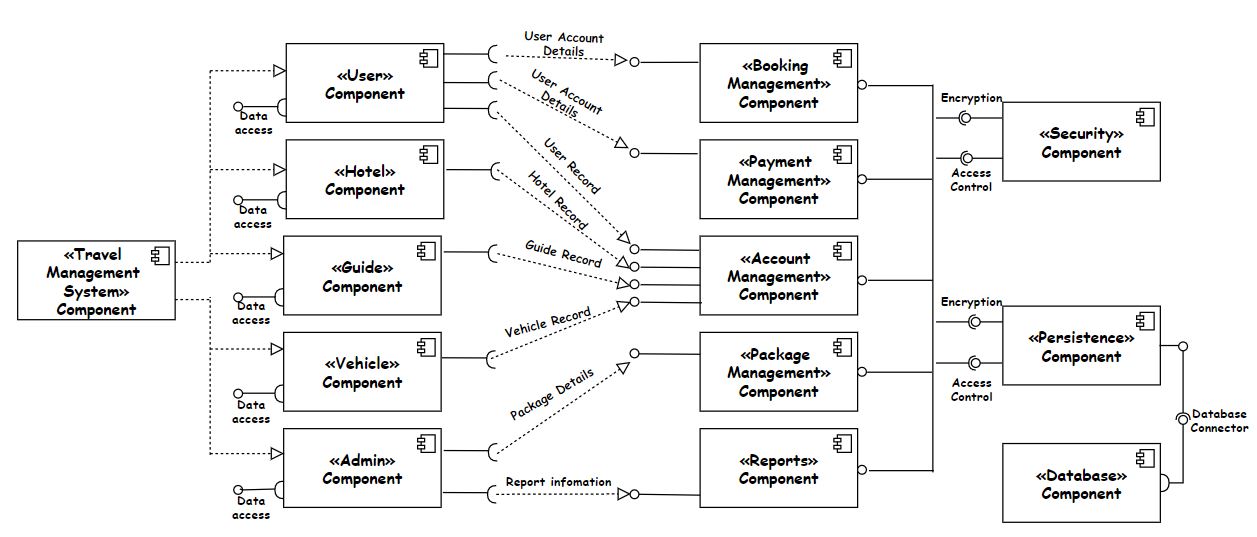
*Figure 08: System Architecture*

## 4.3 Components & their responsibilities

|  |  |
| --- | --- |
| Component | Responsibility |
| User | * Register to the system. * Book the travel packages * Add comments and ratings. * Send messages to the administrator. * Do the payment for booked travel packages. * Maintain his profile. |
| Hotel | * Register to the system. * Change the availability of the hotel rooms daily. * Maintain his profile. |
| Guide | * Register to the system. * Change the availability of whether he is available or not by daily. * Maintain his profile. |
| Vehicle owner | * Register to the system. * Change the availability of whether the vehicle is available or not by daily. * Maintain his profile. |
| Admin | * Maintain travel packages including categories, subcategories, and destinations. * Check availability and assign the hotel, vehicle, and guide for the booking. * Check reports. |
| Security | * Enables safe operations. * Protects data the organization collects and uses |
| MySQL database. | * Stores and retrieves data. |
| PHP scripting engine | * PHP script code embedded in the web page is invoked to request data from the MySQL database. * Data returned by the MySQL database is proceeded by the PHP script to customize the web page. |
| Persistence | * Realizes store and retrieve functionality for supported services. * Generates a scheduler for each supported service to partition data to suitable database back-ends. |
| Apache web server | * Establishes a connection between a server and the browsers of website visitors. * Delivers files back forth between client and server. |
| Payment management component | * The payment gateway used to make the payments that are done by the user. * It is authorized transactions between Easy Travel Company and users. |
| Booking management component | * Stores bookings which are done by the user * Update the bookings after user cancel a booking |
| Report management component | * Preview the reports according to the data in the database. |

*Table 44: Components and their responsibilities*

## 4.4 Component diagram



*Figure 09: Component Diagram*

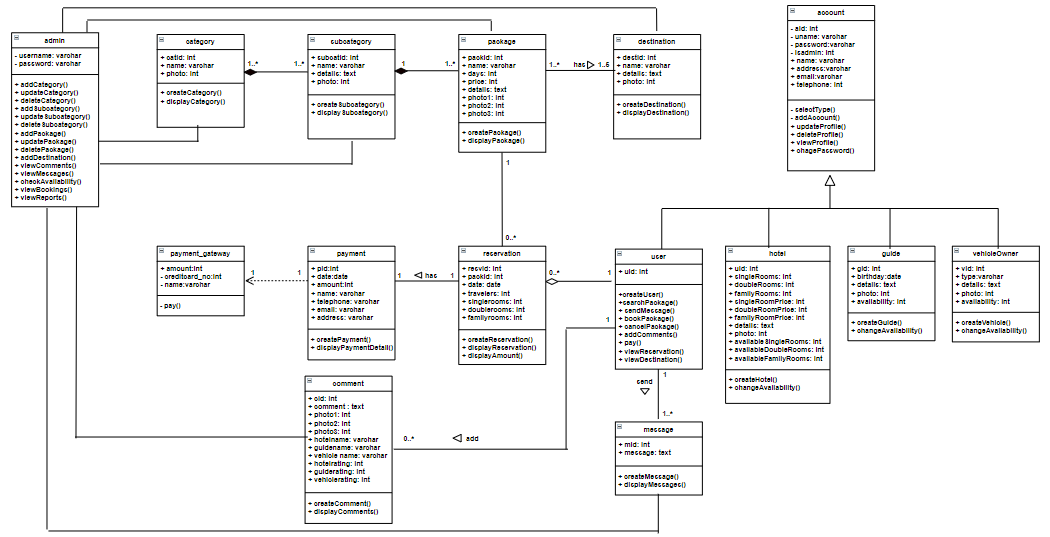
## 4.5 Component interactions

Our system based on the following basic components.

* User component
  + Provides interfaces for data access.
  + Requires services from the booking management component, payment management component, and account management component in the system.
* Hotel component
  + Provides interfaces for data access.
  + Requires services from the account management component.
* Vehicle component
  + Provides interfaces for data access.
  + Requires services from the account management component.
* Guide component
  + Provides interfaces for data access.
  + Requires services from the account management component.
* Admin component
  + Requires services from the package management component and the report component.
* Booking Management component
  + Provides services for the user component.
* Payment Management component
  + Provide services for the user component.
* Account Management component
  + Provides services for user, admin, hotel, guide, and vehicle components.
* Package Management component
  + Provides services for the admin component.
* Report Management component
  + Provides services for the admin component.
* Security component
  + Provides services for booking management, account management, package management, and report management component.
* Persistence component
  + Provides services for booking management, account management, package management, and report management component.
  + The persistence component requires services from the database component.
* Database component
  + The database component provides services for the persistence component through the database connector.

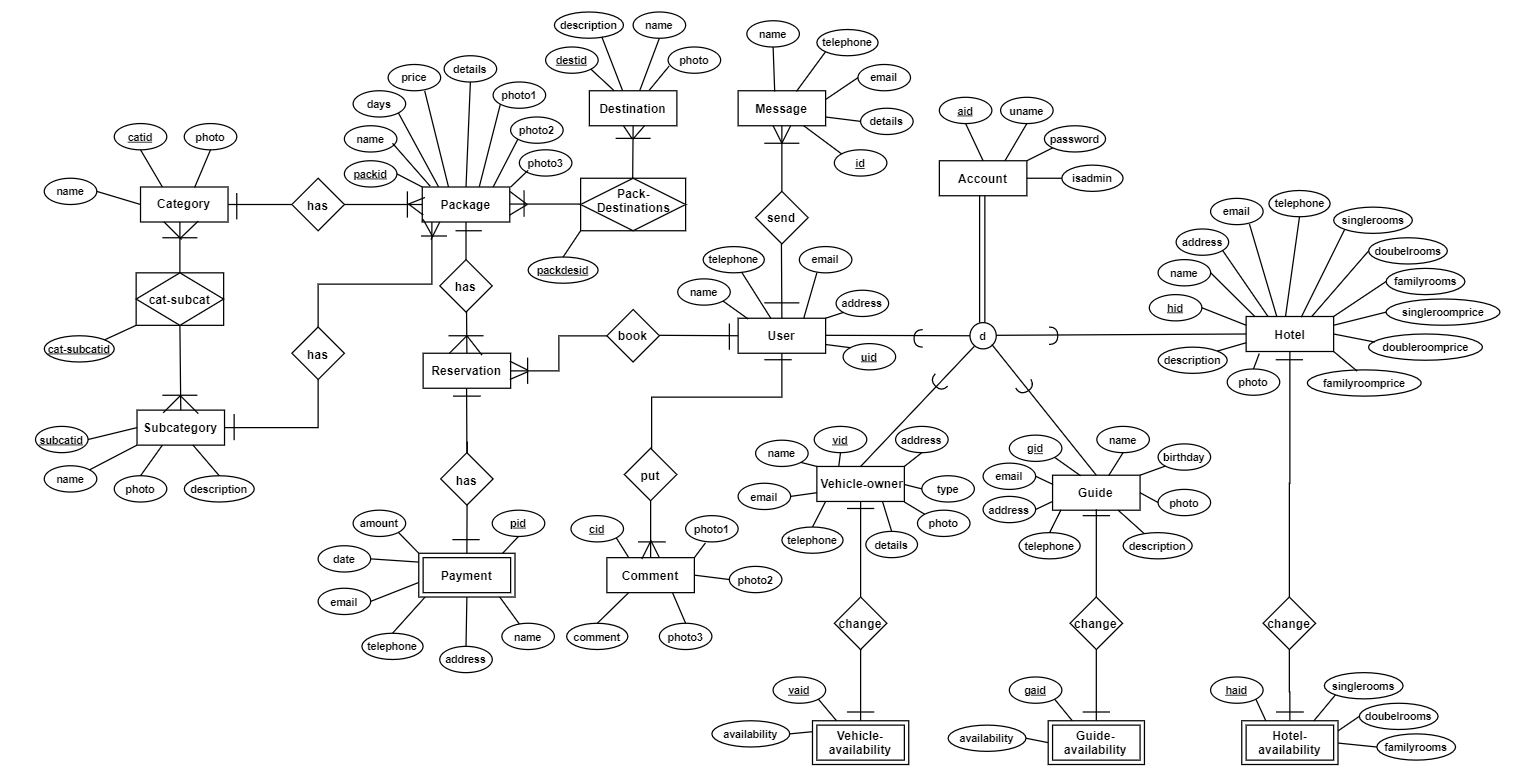
# 5. System’s Design

## 5.1 Class diagram



*Figure 10: Class Diagram*

## 5.2 ER diagram

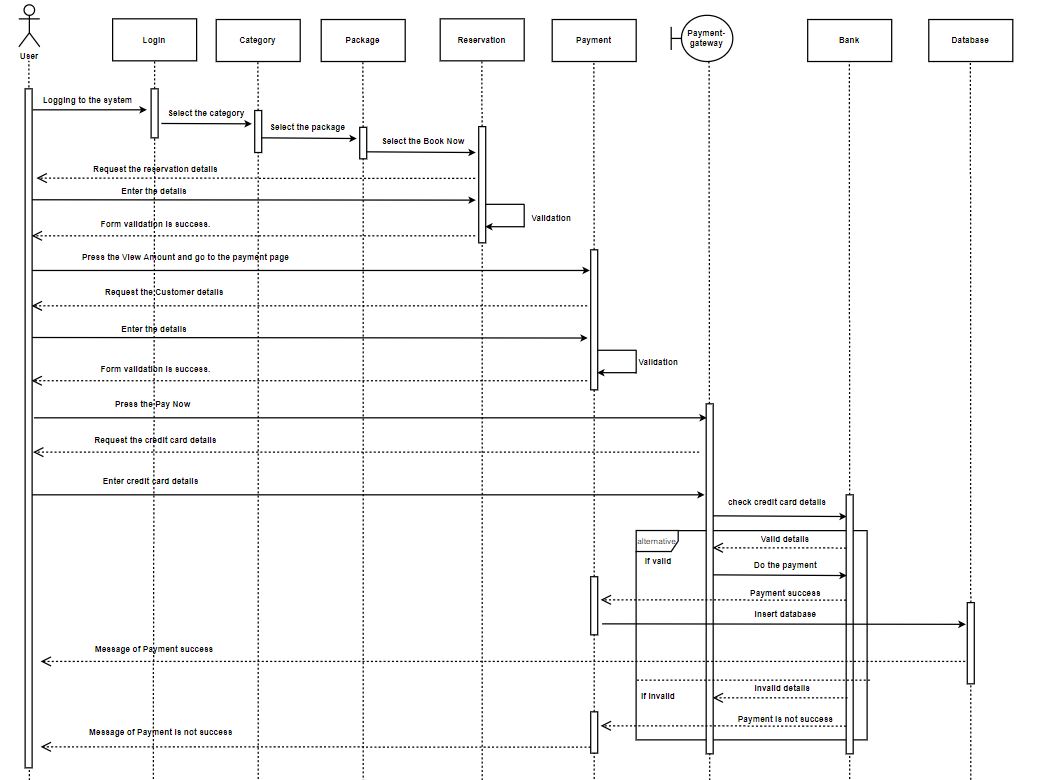


*Figure 11: ER Diagram*

## 5.3 Sequence diagrams

### 5.3.1 User’s Sequence diagrams

#### 5.3.1.1 User books travel package



*Figure 12: Sequence diagram for user booking travel package*

#### 5.3.1.2 User cancels the booking



*Figure 13: Sequence diagram for user cancel a booking*

#### 5.3.1.3 User adds comment

*Figure 14: Sequence diagram for user adds comment*

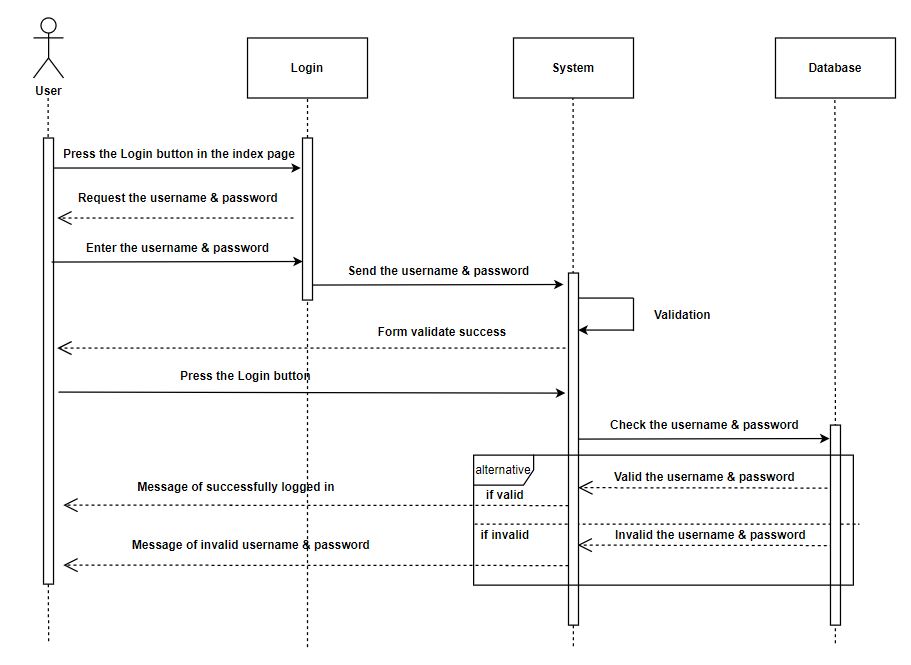
#### 5.3.1.4 User updates password

*Figure 15: Sequence diagram for user updates password*

#### 5.3.1.5 User deletes profile

*Figure 16: Sequence diagram for user deletes profile*

#### 5.3.1.6 User login



*Figure 17: Sequence diagram for user login*

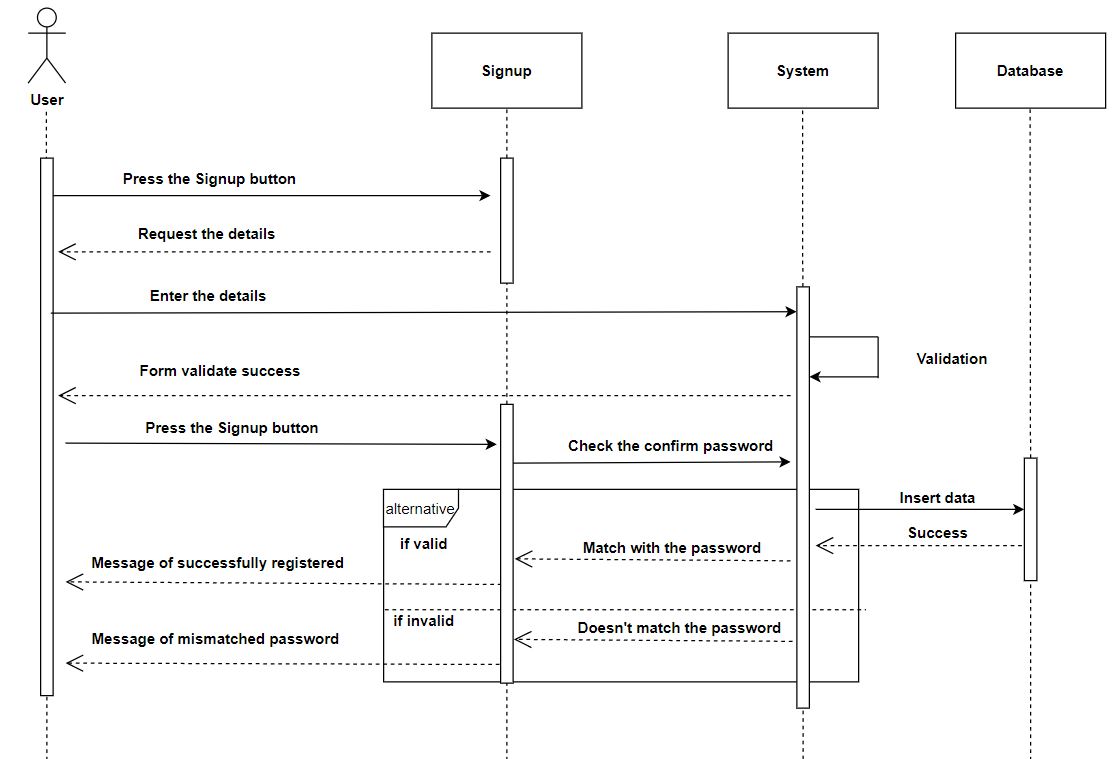
#### 5.3.1.7 User searches destinations

*Figure 18: Sequence diagram for user searches destinations*

#### 5.3.1.8 User sends messages

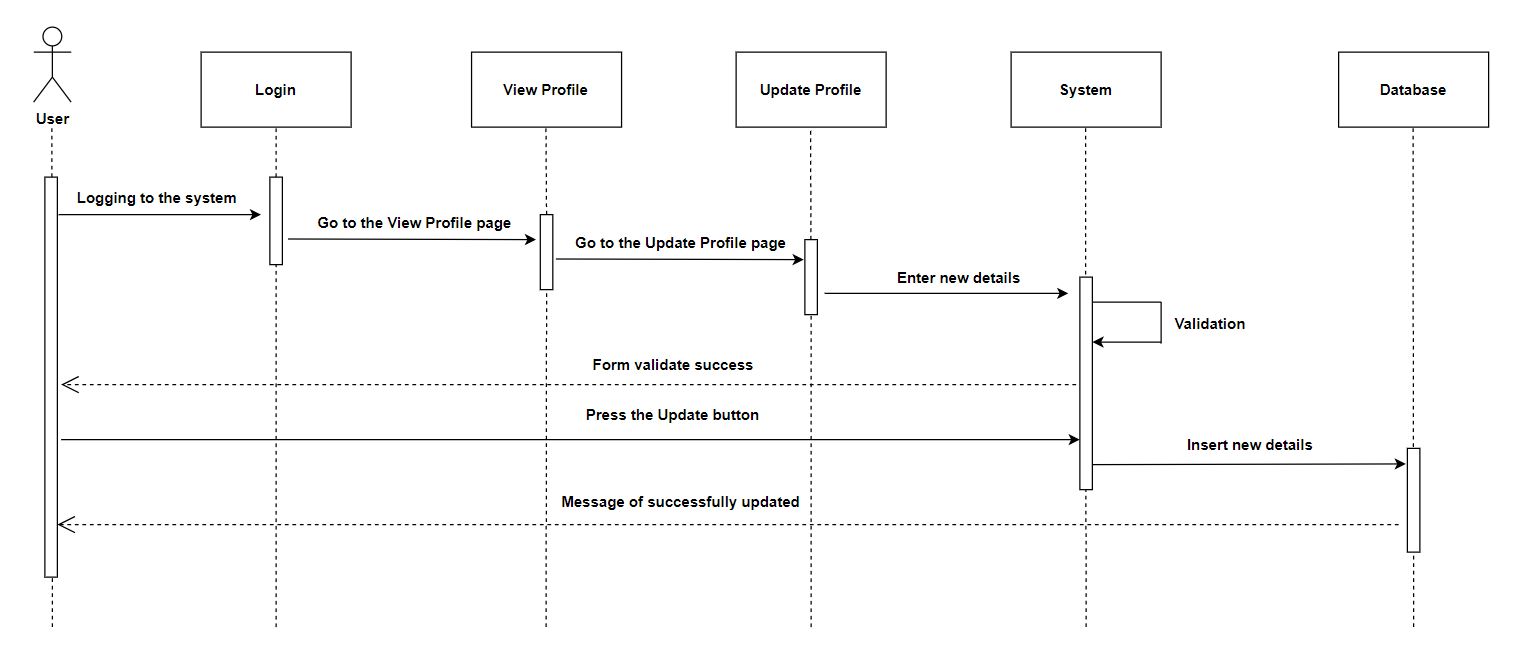
*Figure 19: Sequence diagram for user sends messages*

#### 5.3.1.9 User signup



*Figure 20: Sequence diagram for user signup*

#### 5.3.1.10 User updates profile



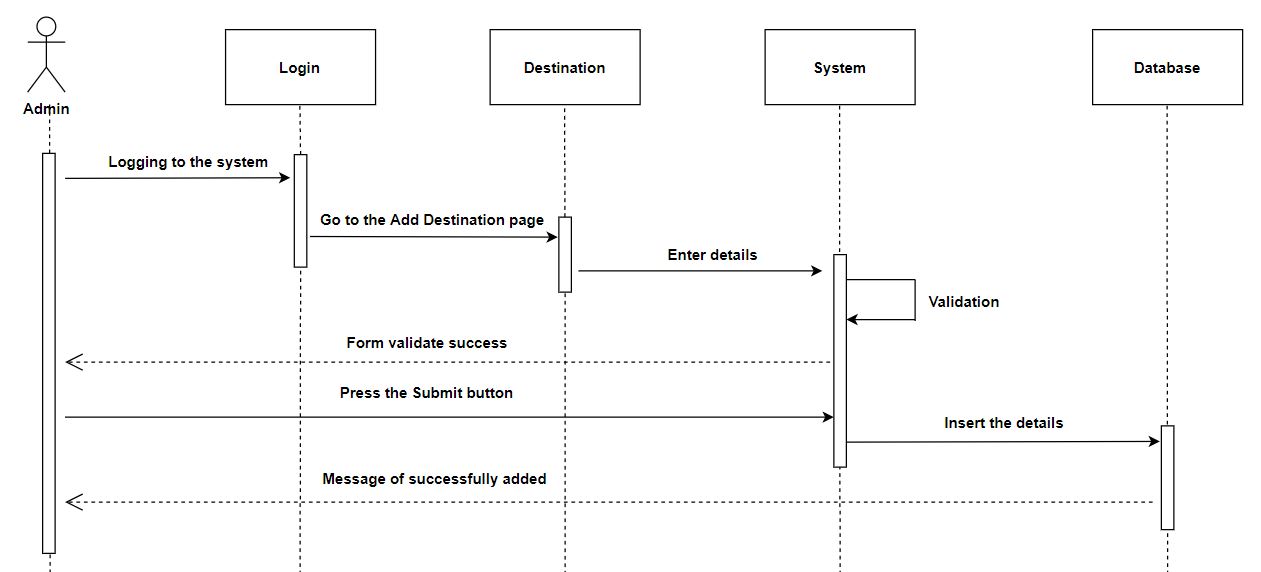
*Figure 21: Sequence diagram for user updates profile*

### 5.3.2 Admin’s Sequence diagrams

#### 5.3.2.1 Admin adds a category

*Figure 22: Sequence diagram for admin adds a category*

#### 5.3.2.2 Admin adds a destination



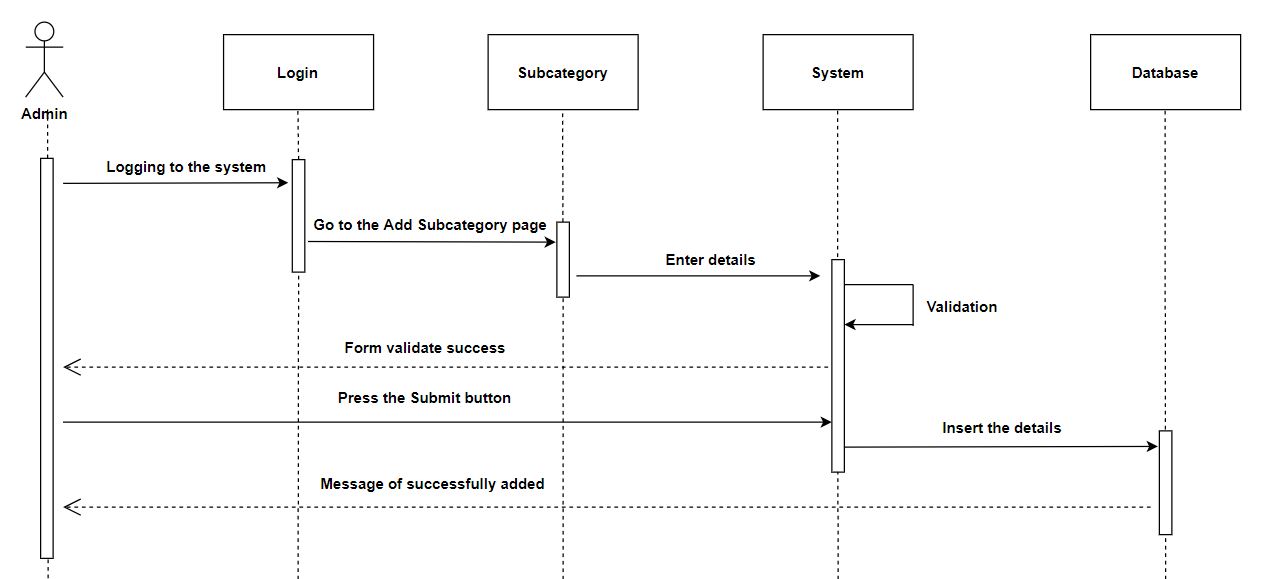
*Figure 23: Sequence diagram for admin adds a destination*

#### 5.3.2.3 Admin adds a package

## 

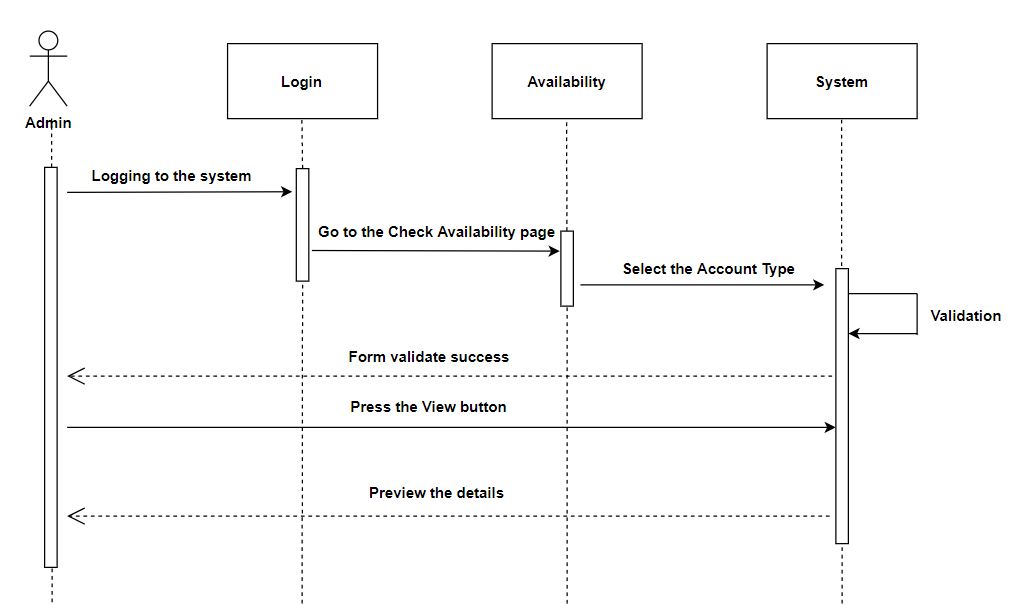
*Figure 24: Sequence diagram for admin adds a package*

#### 5.3.2.4 Admin adds a subcategory



*Figure 25: Sequence diagram for admin adds a subcategory*

#### 5.3.2.5 Admin checks availability

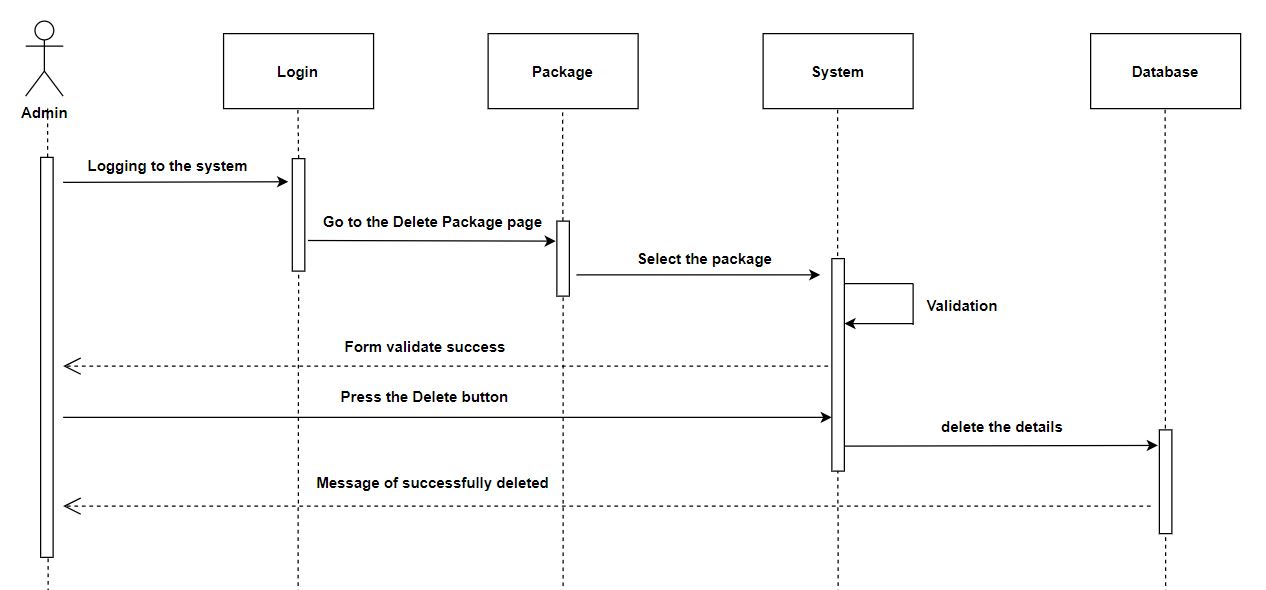


*Figure 26: Sequence diagram for admin checks the availability*

#### 5.3.2.6 Admin deletes a category

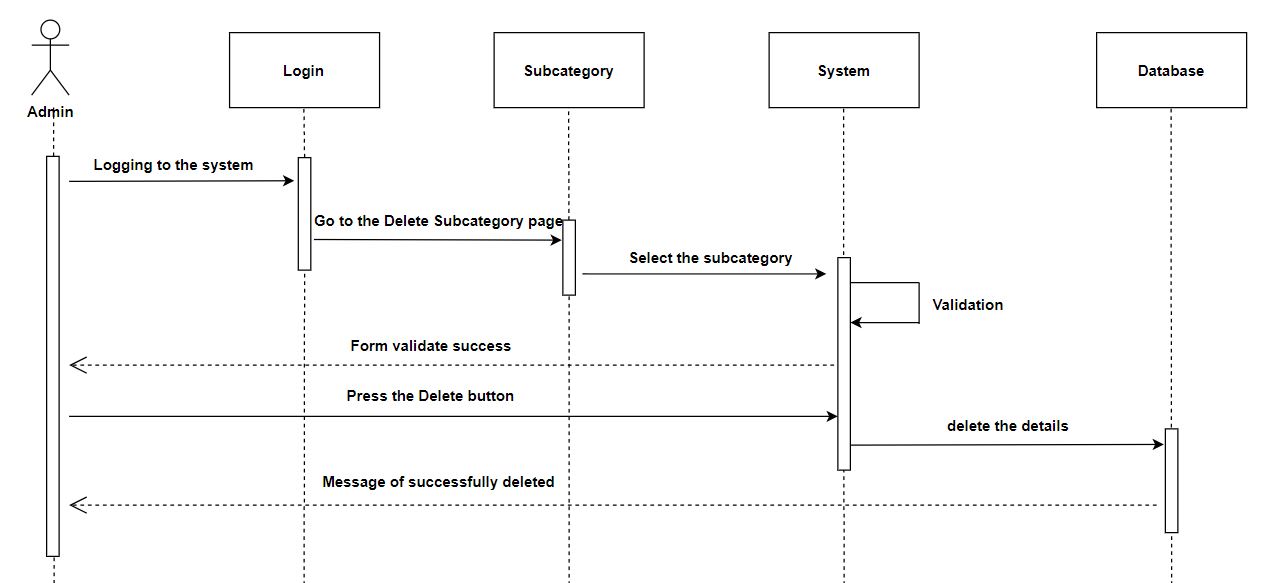
*Figure 27: Sequence diagram for admin deletes a category*

#### 5.3.2.7 Admin deletes a package



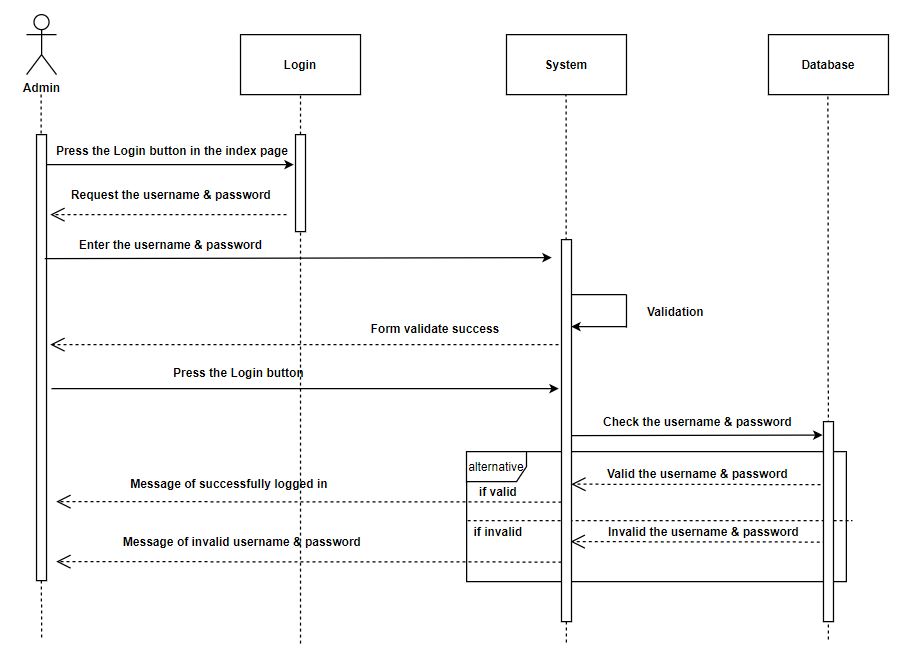
*Figure 28: Sequence diagram for admin deletes package*

#### 5.3.2.8 Admin deletes a subcategory



*Figure 29: Sequence diagram for admin deletes a subcategory*

#### 5.3.2.9 Admin login

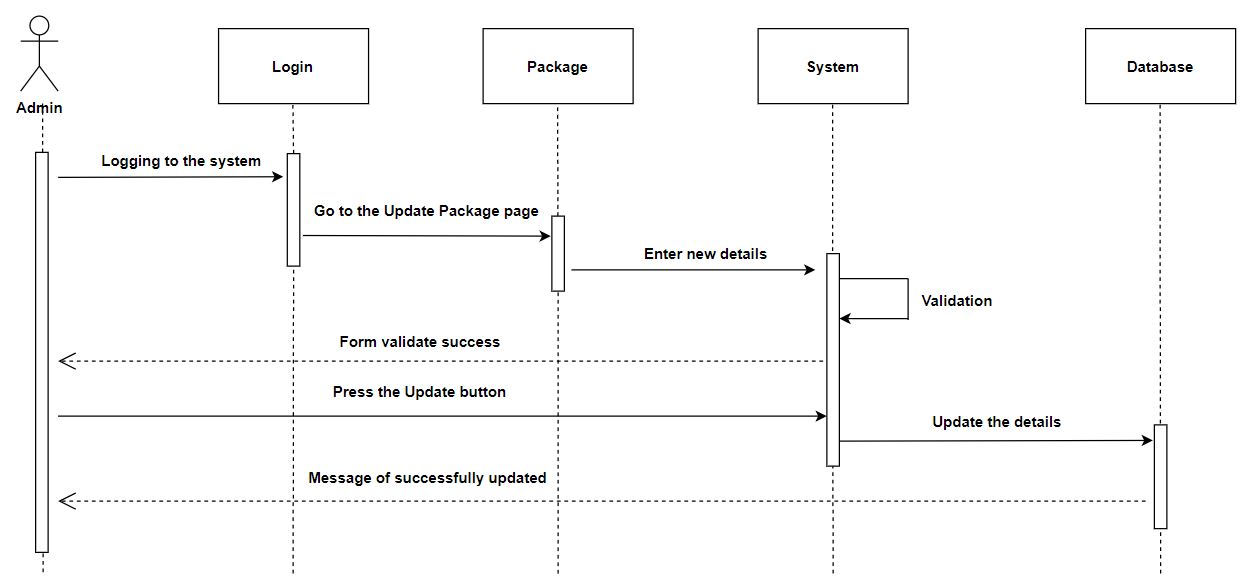


*Figure 30: Sequence diagram for admin login*

#### 5.3.2.10 Admin updates a category

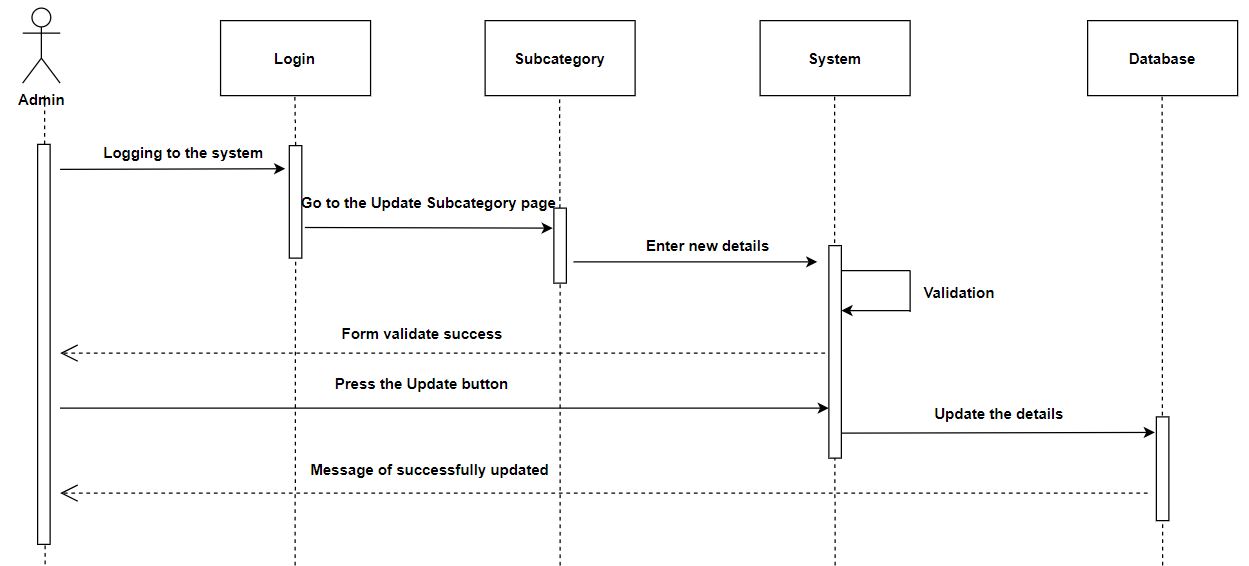
*Figure 31: Sequence diagram for admin updates a category*

#### 5.3.2.11 Admin updates a package



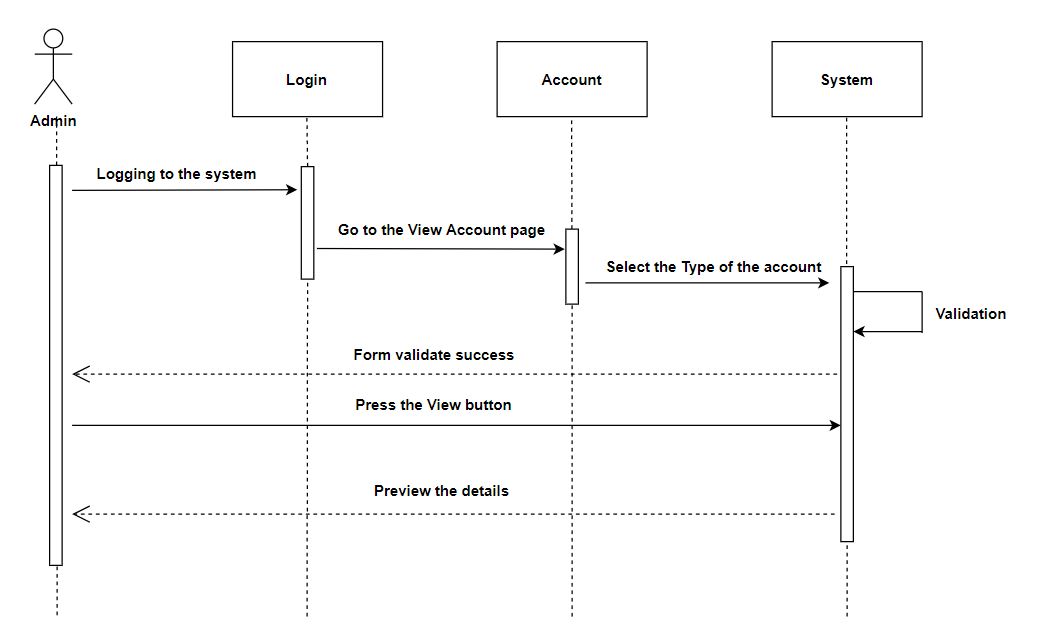
*Figure 32: Sequence diagram for admin updates a package*

#### 5.3.2.12 Admin updates a subcategory



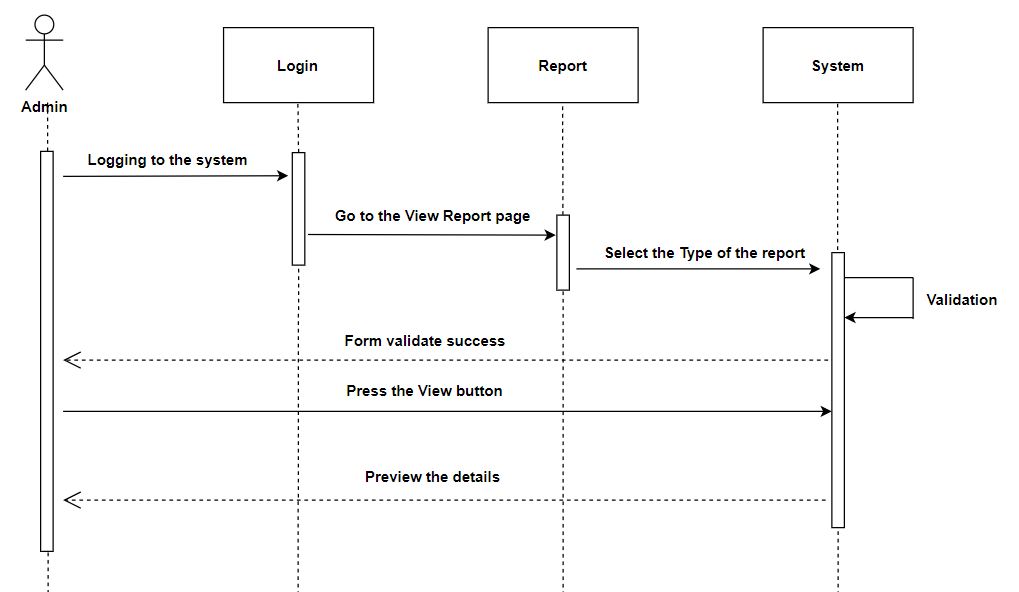
*Figure 33: Sequence diagram for admin updates a subcategory*

#### 5.3.2.13 Admin views accounts



*Figure 34: Sequence diagram for admin views accounts*

#### 5.3.2.14 Admin views reports



*Figure 35: Sequence diagram for admin views reports*

### 5.3.3 Hotel’s Sequence diagrams

#### 5.3.3.1 Hotel login

*Figure 36: Sequence diagram for hotel login*

### 5.3.3.2 Hotel changes availability

*Figure 37: Sequence diagram for hotel changes availability*

### 5.3.3.3 Hotel changes password

*Figure 38: Sequence diagram for hotel changes password*

### 5.3.3.4 Hotel updates profile

*Figure 39: Sequence diagram for hotel updates profile*

#### 5.3.3.5 Hotel signup

#### 

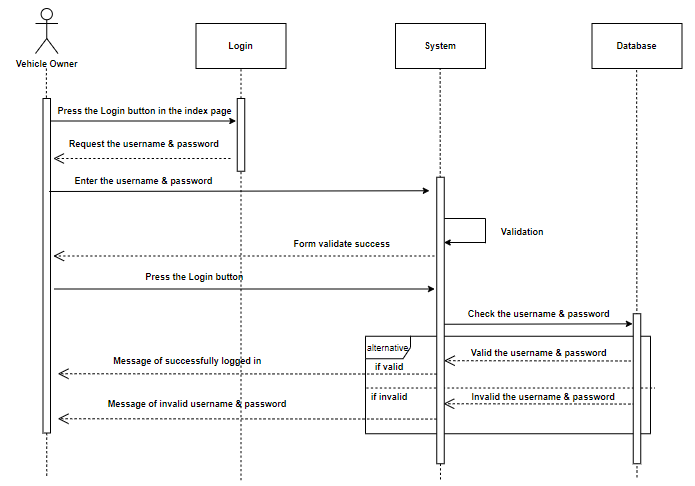
*Figure 40: Sequence diagram for hotel sign up*

#### 5.3.3.6 Hotel deletes profile

*Figure 41: Sequence diagram for hotel deletes profile*

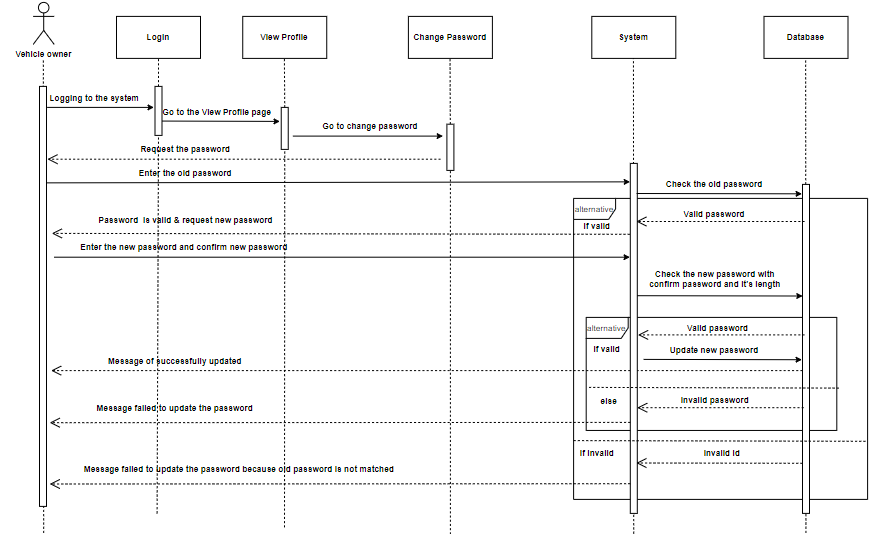
### 5.3.4 Vehicle owner’s Sequence diagrams

#### 5.3.4.1 Vehicle owner login



*Figure 42: Sequence diagram for vehicle owner login*

#### 5.3.4.2 Vehicle owner changes password

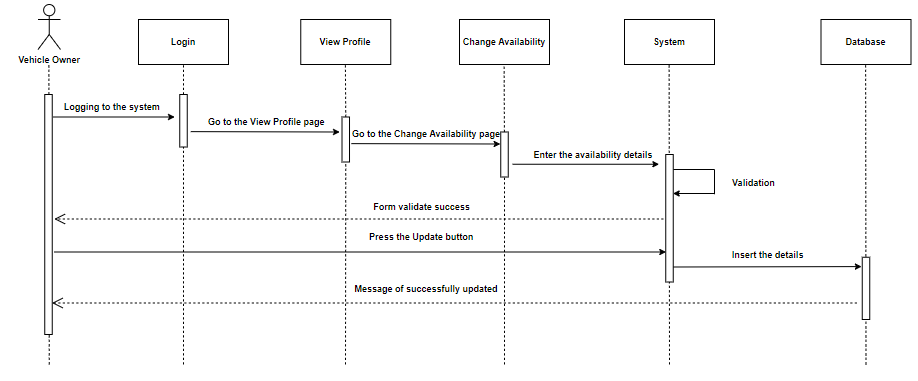


*Figure 43: Sequence diagram for vehicle owner changes password*

#### 5.3.4.3 Vehicle owner signup

*Figure 44: Sequence diagram for vehicle owner signup*

#### 5.3.4.4 Vehicle owner changes availability



*Figure 45: Sequence diagram for vehicle owner changes availability*

#### 5.3.4.5 Vehicle owner updates profile

*Figure 46: Sequence diagram for vehicle owner update profile*

#### 5.3.4.5 Vehicle owner deletes profile

*Figure 47: Sequence diagram for vehicle owner deletes profile*

### 5.3.5 Guide’s Sequence diagrams

#### 5.3.5.1 Guide login

*Figure 48: Sequence diagram for guide login*

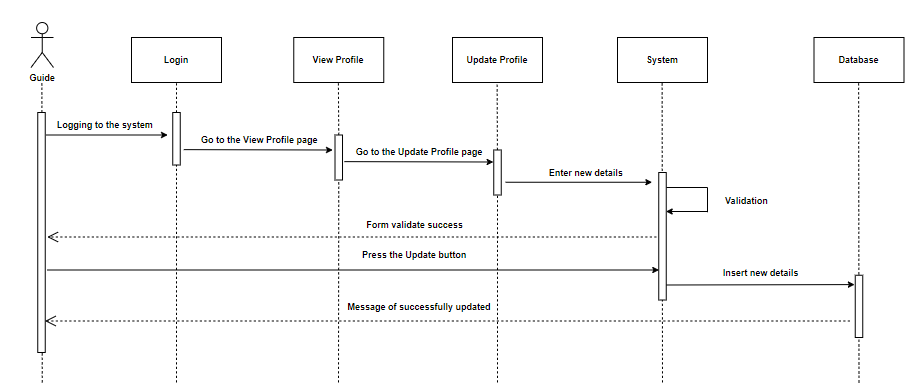
#### 5.3.5.2 Guide changes availability

*Figure 49: Sequence diagram for guide changes availability*

#### 5.3.5.3 Guide changes password

*Figure 50: Sequence diagram for guide changes password*

#### 5.3.5.4 Guide updates profile



*Figure 51: Sequence diagram for guide updates the profile*

#### 5.3.5.5 Guide signup

## 

*Figure 52: Sequence diagram for guide signup*

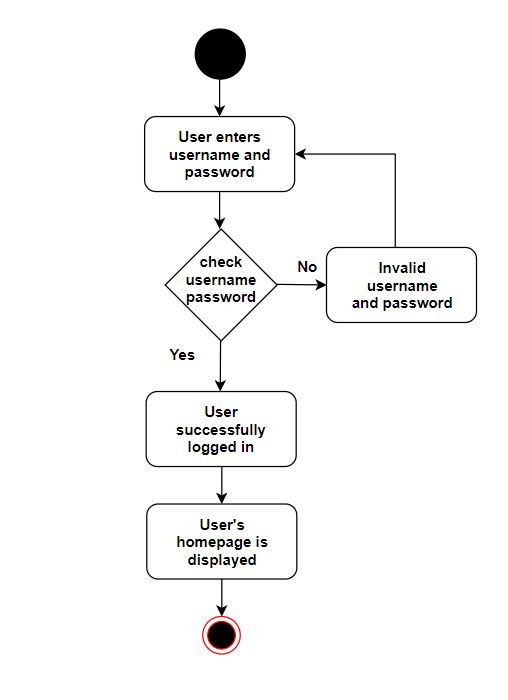
#### 5.3.5.6 Guide deletes profile

*Figure 53: Sequence diagram for guide deletes profile*

## 5.4 Activity Diagrams

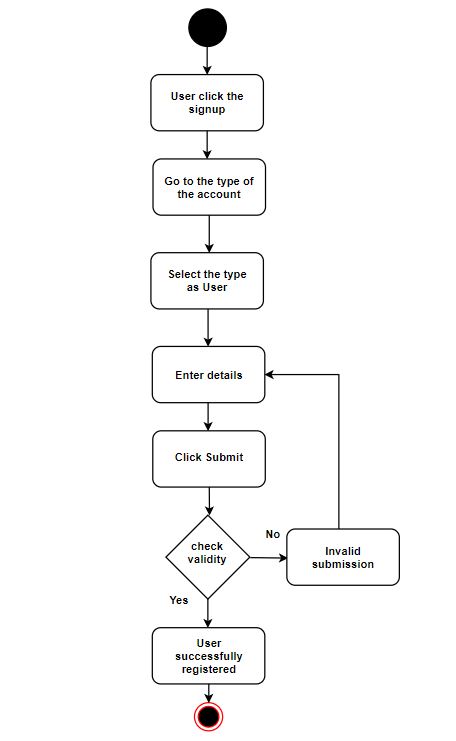
### 5.4.1 Activity diagrams for User

#### 5.4.1.1 Login to the system



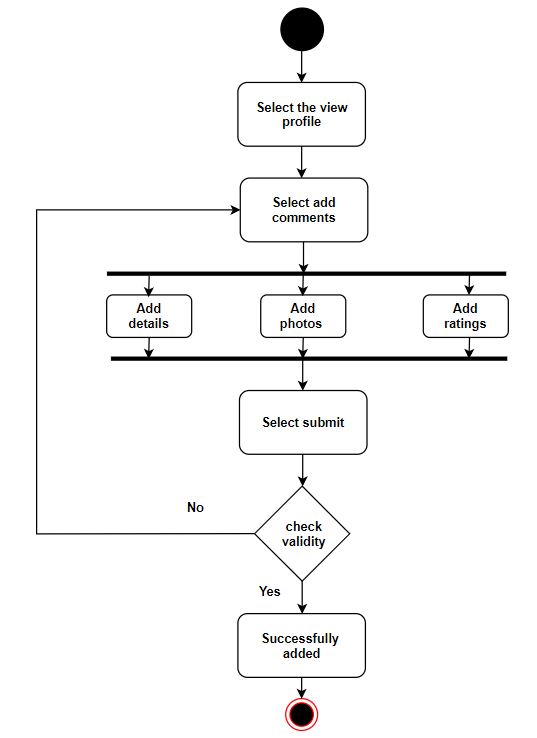
*Figure 54: User login to the system*

#### 5.4.1.2 Create an account



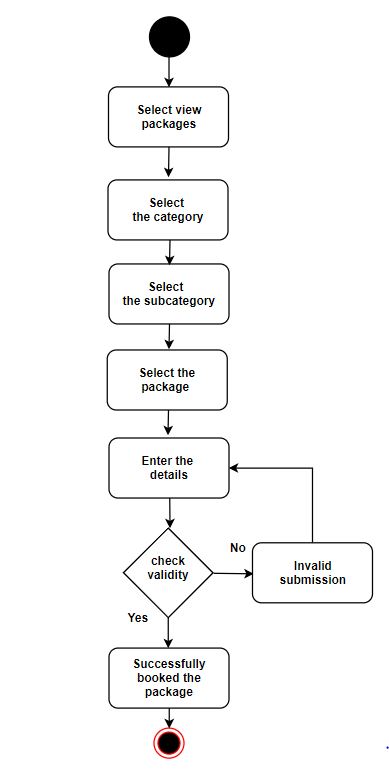
*Figure 55: User creates an account*

#### 5.4.1.3 Add comments



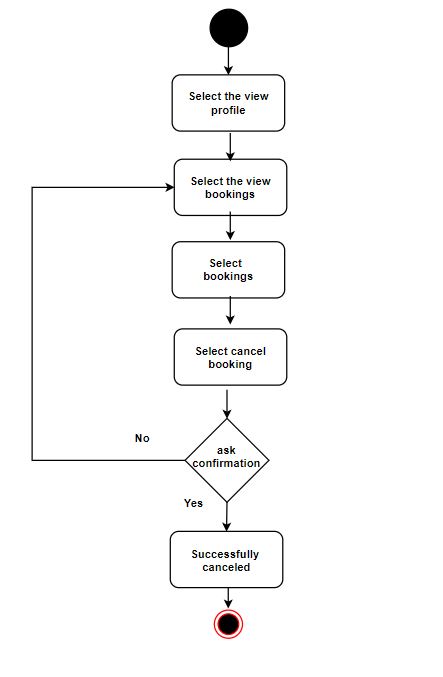
*Figure 56: User adds comments*

#### 5.4.1.4 Book travel package



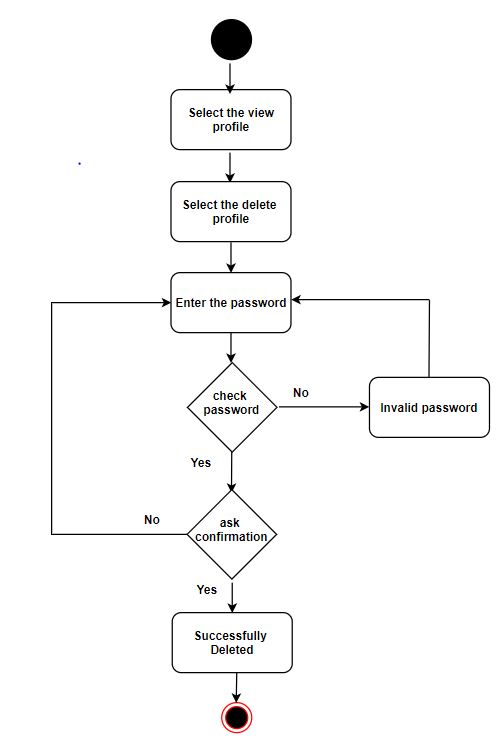
*Figure 57: User books travel package*

#### 5.4.1.5 Cancel the booking



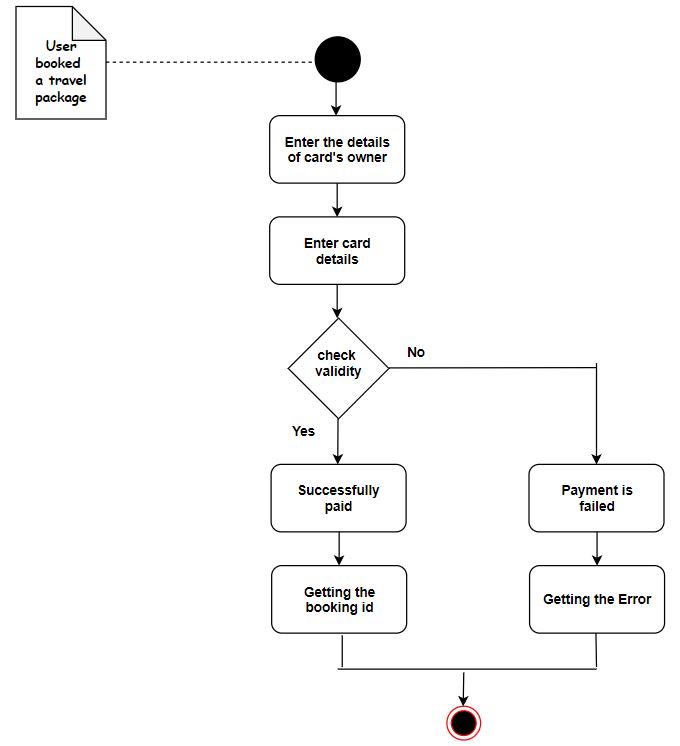
*Figure 58: User cancels the booking*

#### 5.4.1.6 Delete the account



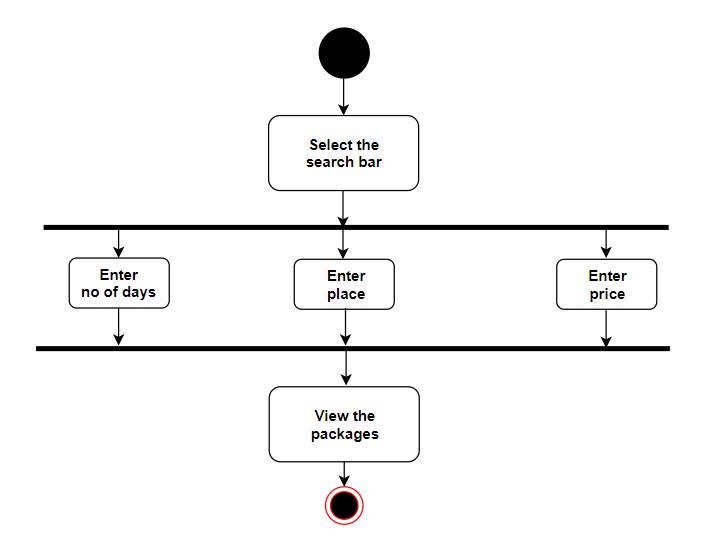
*Figure 59: User deletes the account*

#### 5.4.1.7 Do payment



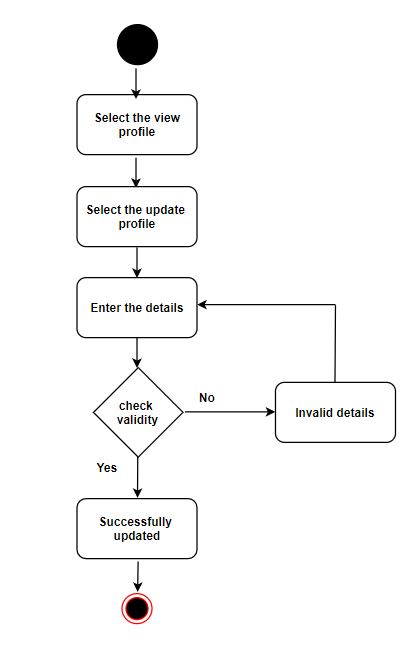
*Figure 60: User does the payment*

#### 5.4.1.8 Search



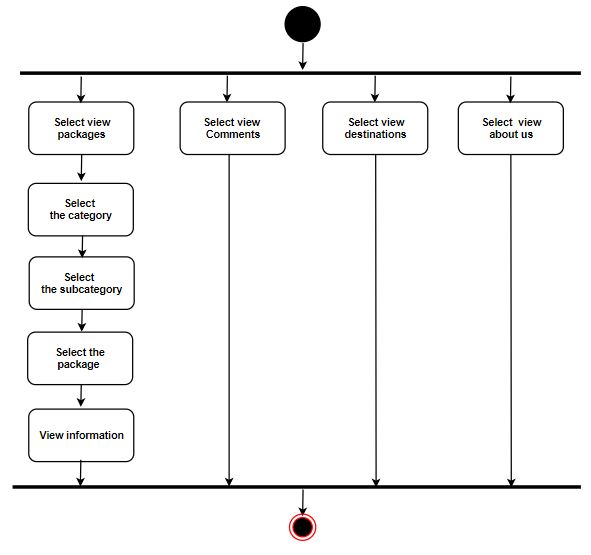
*Figure 61: User search*

#### 5.4.1.9 Update the account



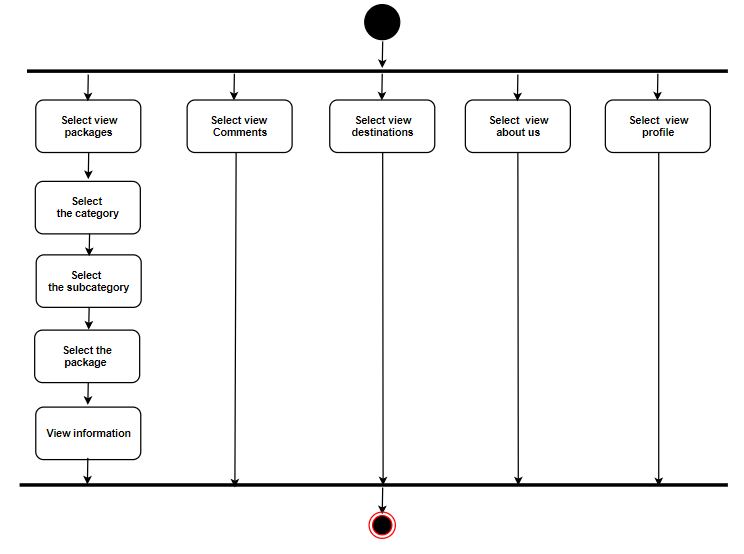
*Figure 62: User updates the account*

#### 5.4.1.10 View [not registered user]



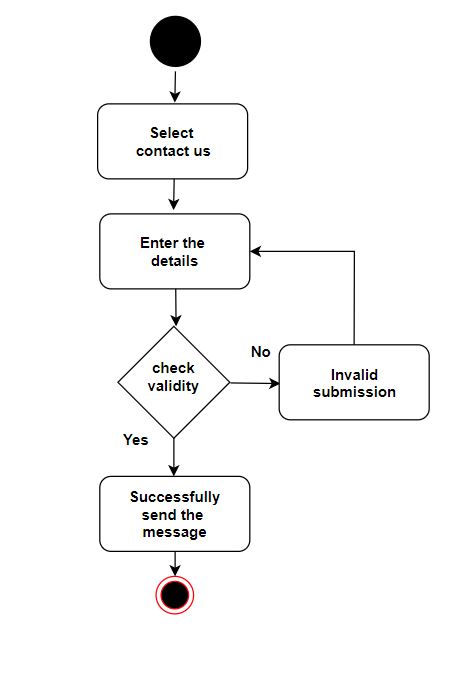
*Figure 63: User view [not registered user]*

#### 5.4.1.11 View [registered user]



*Figure 64: User view [registered user]*

#### 5.4.1.12 User sends message

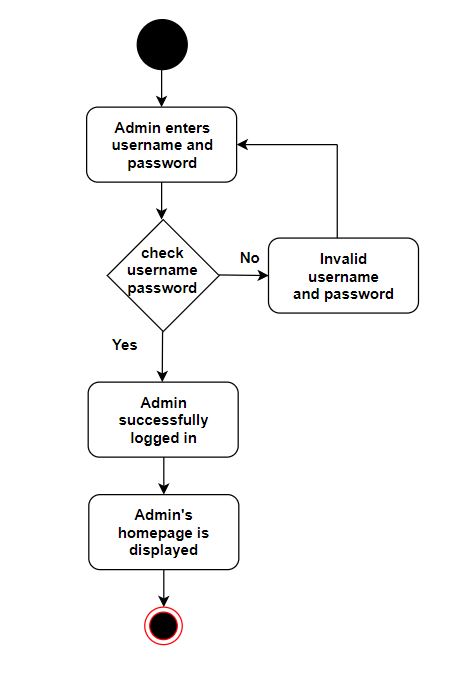


*Figure 65: User sends message*

### 

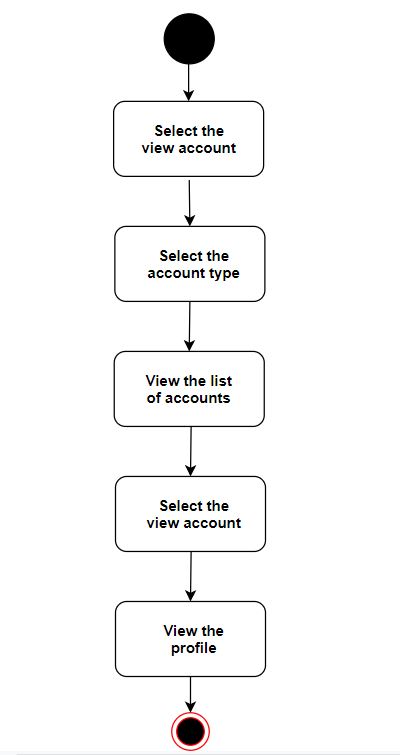
### 5.4.2 Activity diagrams for Admin

#### 5.4.2.1 Login to the system



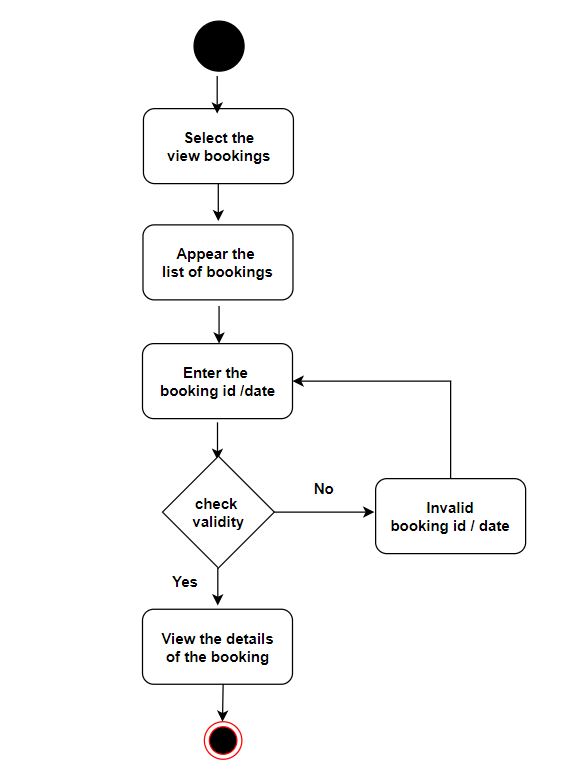
*Figure 66: Admin login to the system*

#### 5.4.2.2 View accounts



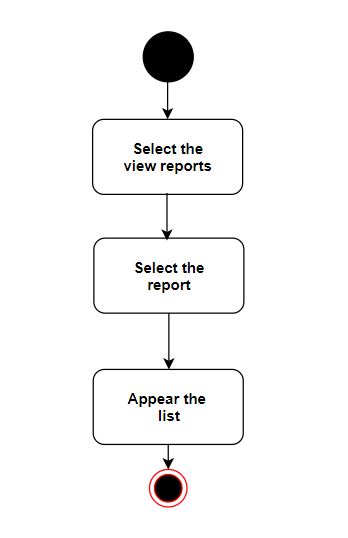
*Figure 67: Admin views accounts*

#### 5.4.2.3 View bookings



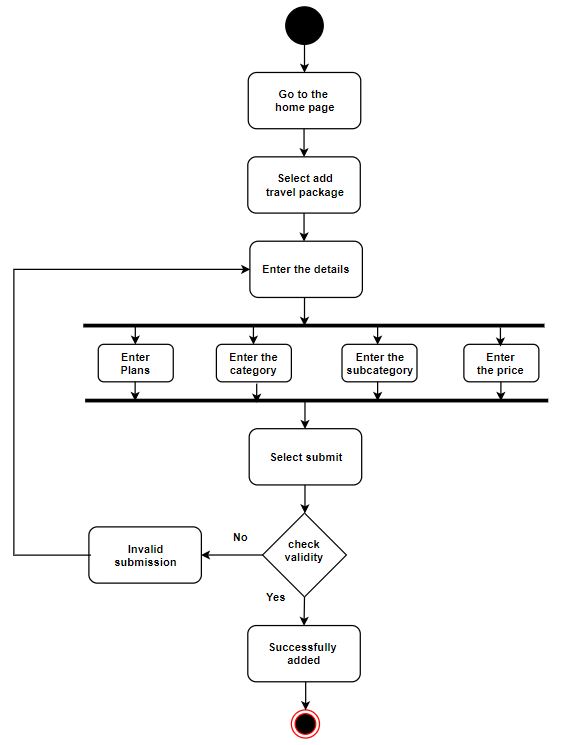
*Figure 68: Admin views bookings*

#### 5.4.2.4 View reports



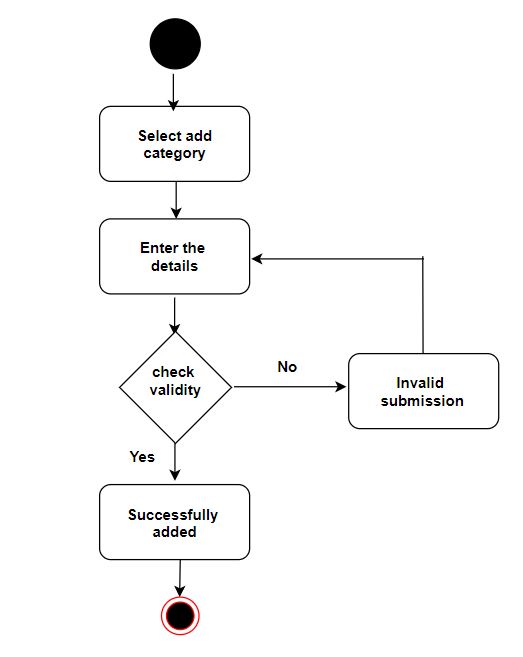
*Figure 69: Admin views reports*

#### 5.4.2.5 Add travel package



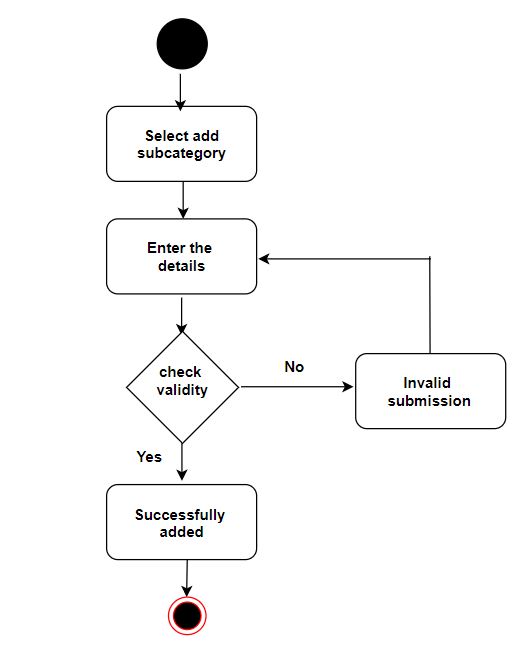
*Figure 70: Admin adds travel package*

#### 5.4.2.6 Add category



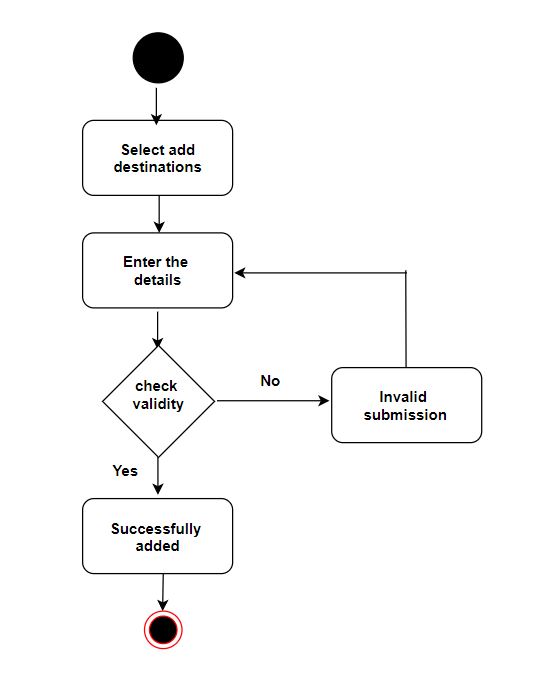
*Figure 71: Admin adds category*

#### 5.4.2.7 Add subcategory



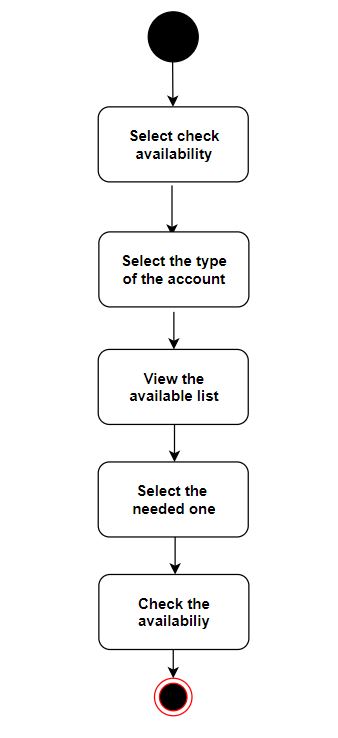
*Figure 72: Admin adds subcategory*

#### 5.4.2.8 Add destinations



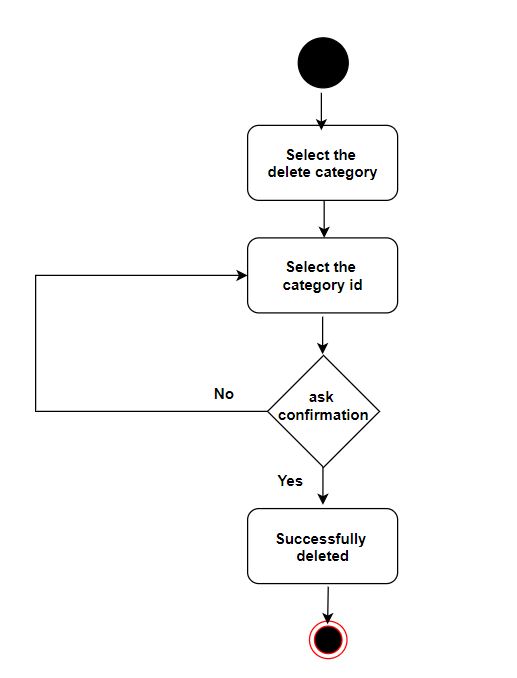
*Figure 73: Admin adds destinations*

#### 5.4.2.9 Check availability



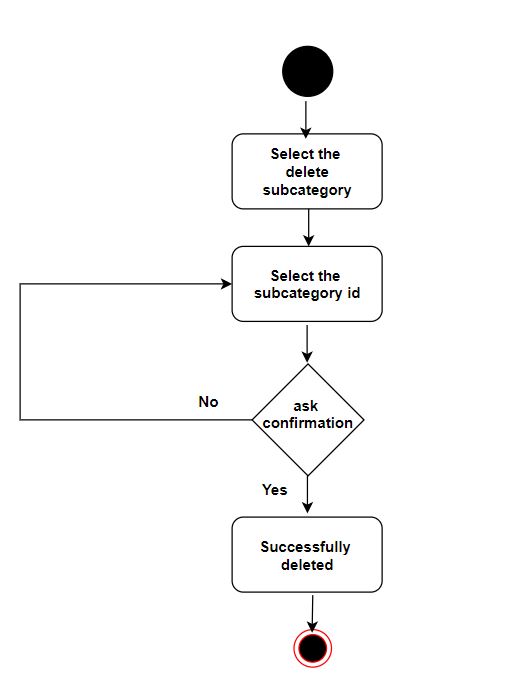
*Figure 74: Admin checks availability*

#### 5.4.2.10 Delete a category



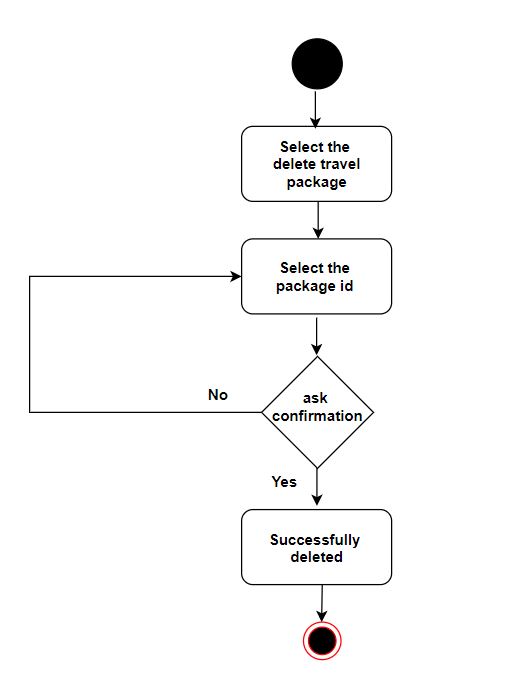
*Figure 75: Admin deletes category*

#### 5.4.2.11 Delete a subcategory



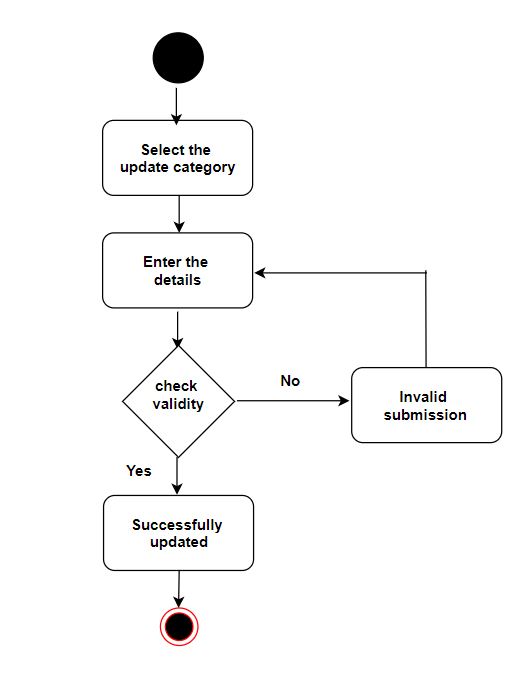
*Figure 76: Admin deletes subcategory*

#### 5.4.2.12 Delete a travel package



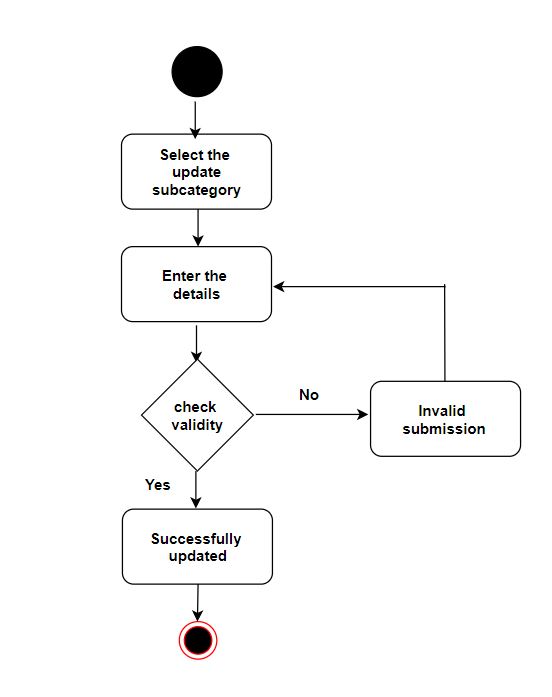
*Figure 77: Admin deletes travel package*

#### 5.4.2.13 Update category



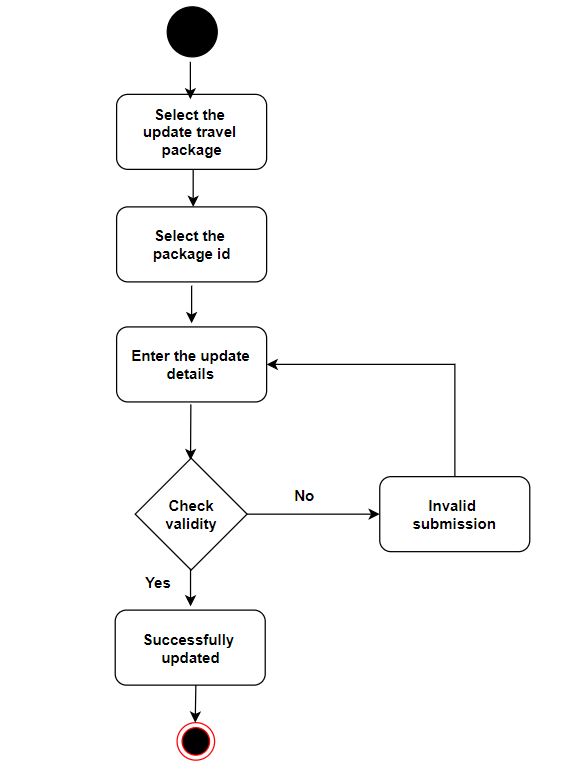
*Figure 78: Admin deletes travel package*

#### 5.4.2.14 Update subcategory



*Figure 79: Admin updates subcategory*

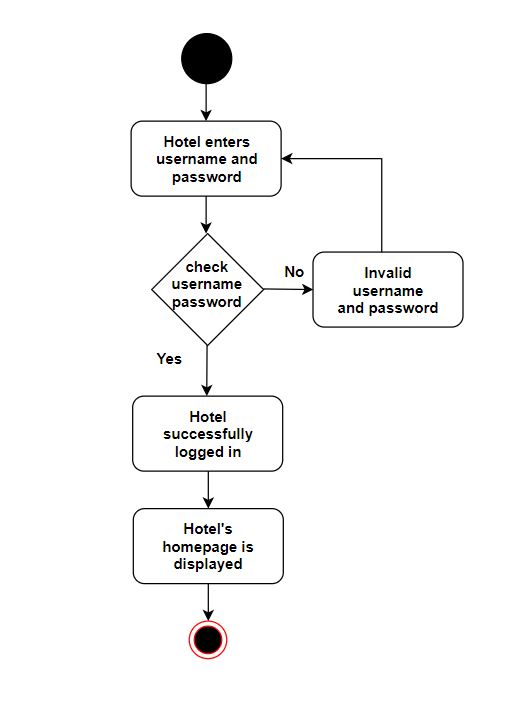
#### 5.4.2.15 Update travel package



*Figure 80: Admin updates travel package*

### 5.4.3 Activity diagrams for Hotel

#### 5.4.3.1 Login to the system



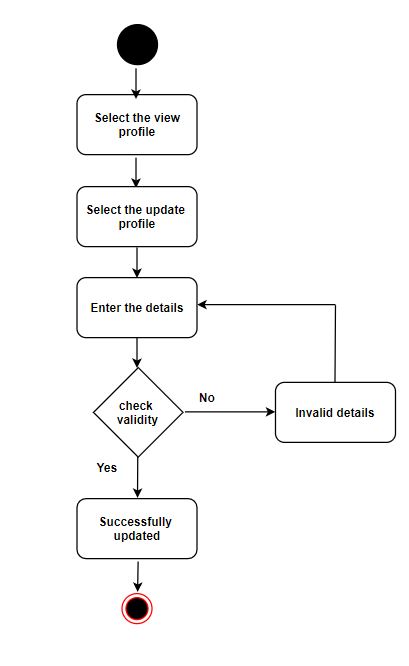
*Figure 81: Hotel login to the system*

#### 5.4.3.2 Create an account

#### 

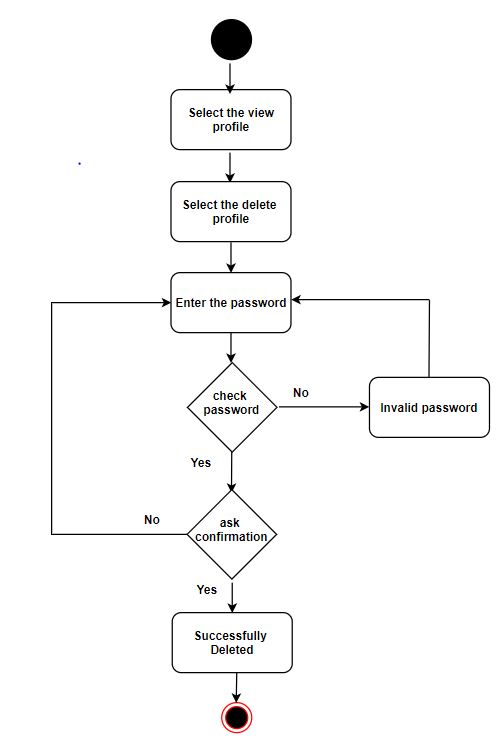
*Figure 82: Hotel creates an account*

#### 5.4.3.3 Update the account



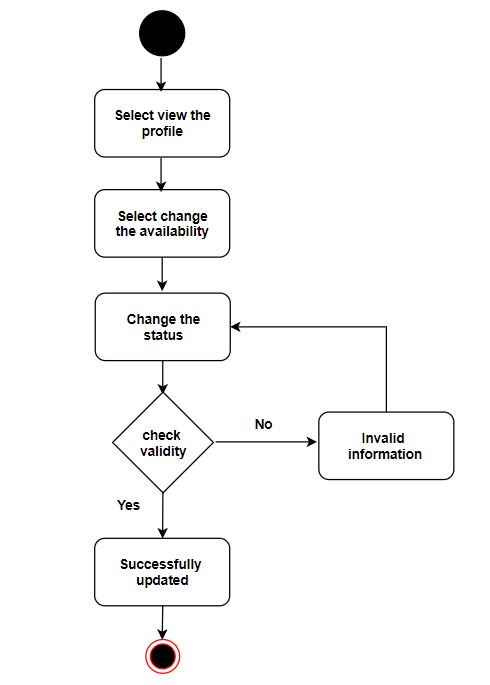
*Figure 83: Hotel updates the account*

#### 5.4.3.4 Delete the account



*Figure 84: Hotel deletes the account*

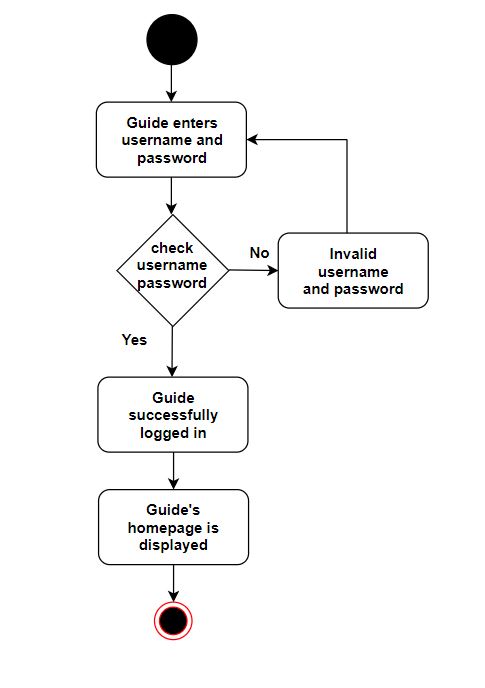
#### 5.4.3.5 Change availability



*Figure 85: Hotel changes the availability*

### 5.4.4 Activity diagrams for Guide

#### 5.4.4.1 Login to the system



*Figure 86: Guide login to the system*

#### 5.4.4.2 Create an account

#### 

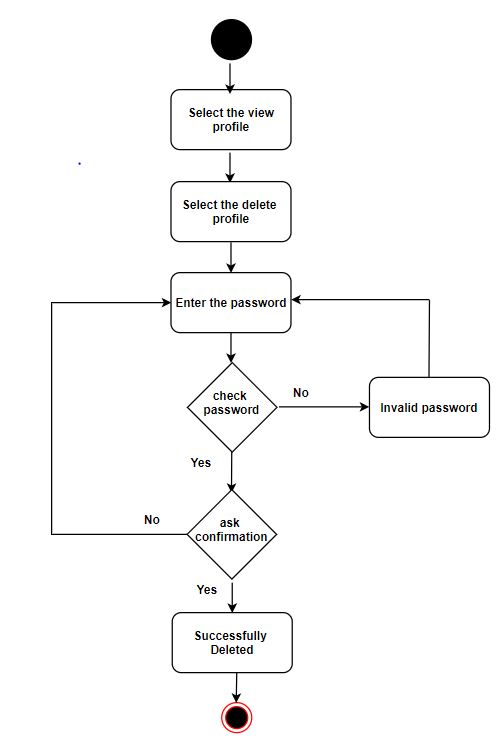
*Figure 87: Guide creates an account*

#### 5.4.4.3 Update the account



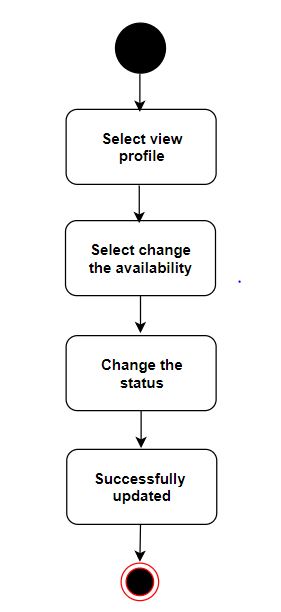
*Figure 88: Guide updates the account*

#### 5.4.4.4 Delete the account



*Figure 89: Guide deletes the account*

#### 5.4.4.5 Change availability



*Figure 90: Guide changes the availability*

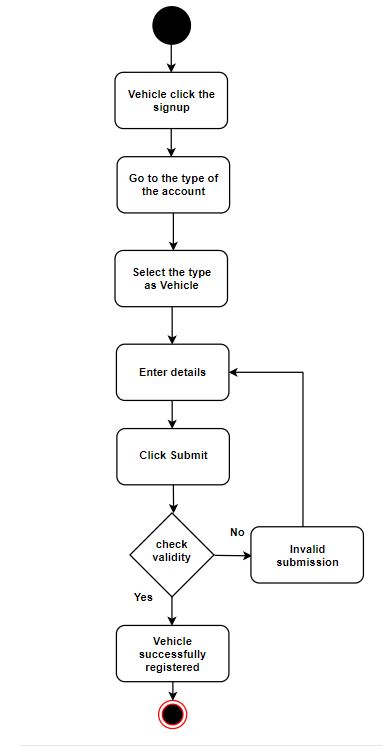
### 5.4.5 Activity diagrams for Vehicle

#### 5.4.5.1 Login to the system



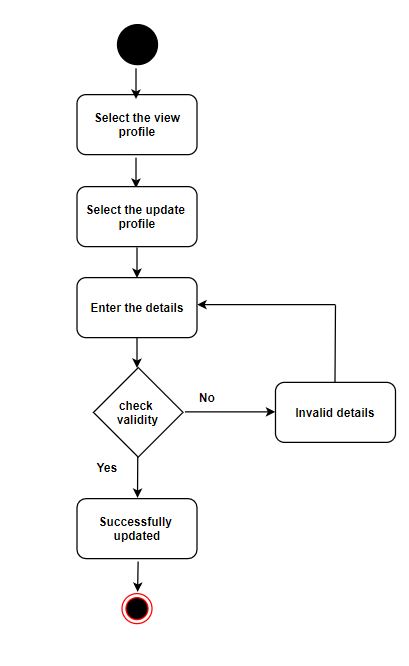
*Figure 91: Vehicle login to the system*

#### 5.4.5.2 Create an account



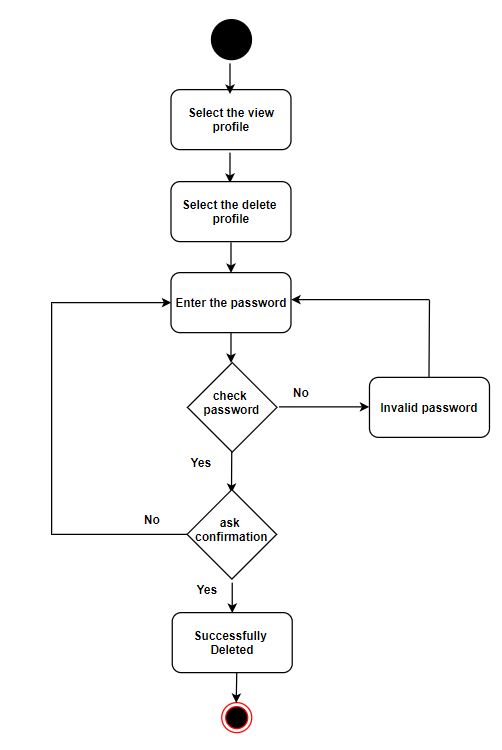
*Figure 92: Vehicle creates an account*

#### 5.4.5.3 Update the account



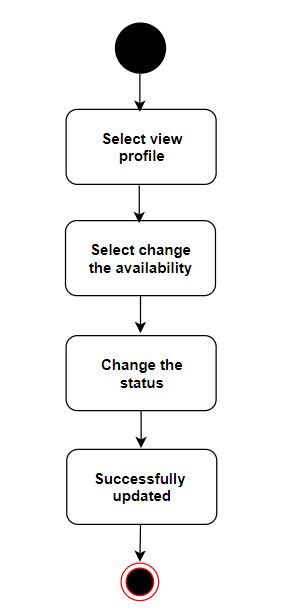
*Figure 93: Vehicle updates the account*

#### 5.4.5.4 Delete the account



*Figure 94: Vehicle deletes the account*

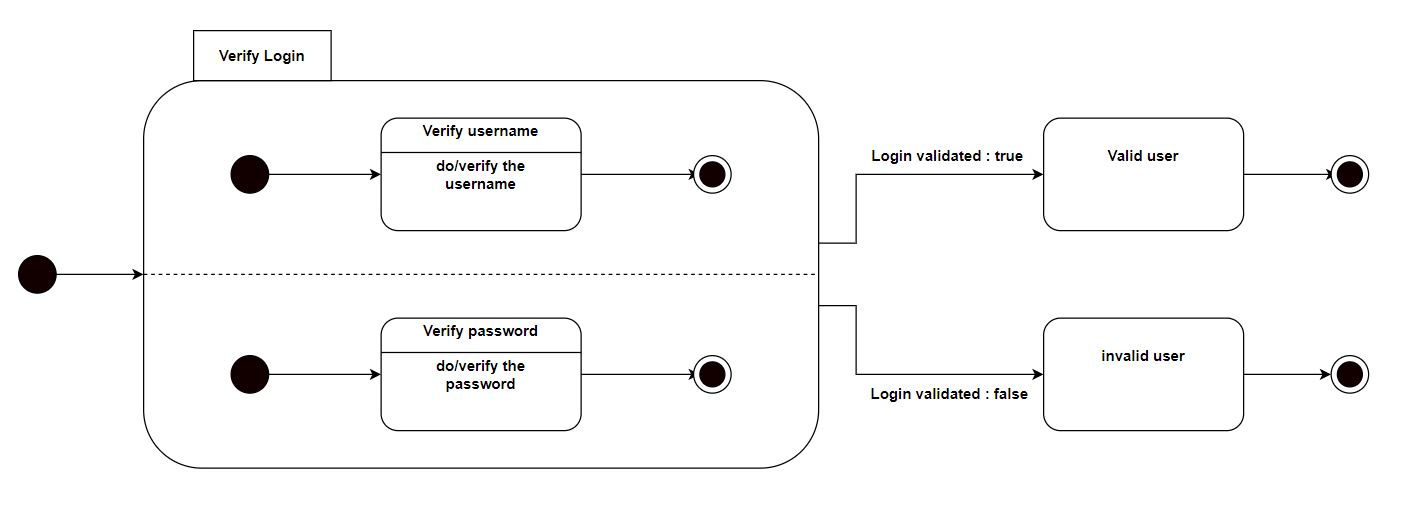
#### 5.4.5.5 Change availability



*Figure 95: Vehicle changes availability*

## 5.5 State Transition Diagrams

### 5.5.1 Login for user, hotel, guide and vehicle owner



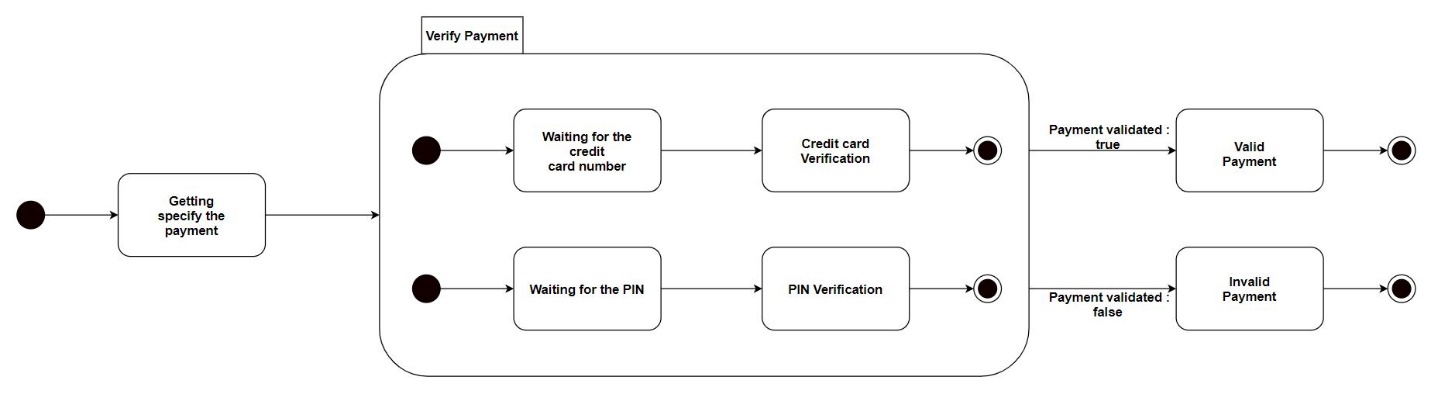
*Figure 96: State transition diagram for accounts’ login*

### 5.5.2 User books a travel package



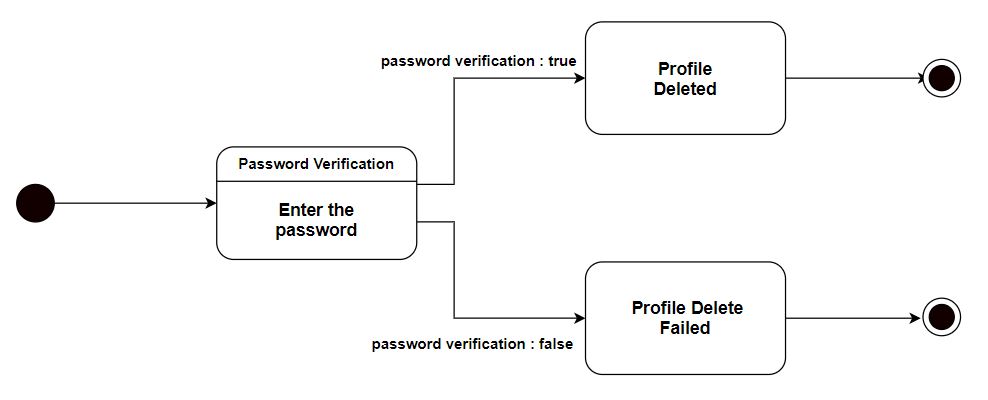
*Figure 97: State transition diagram for user books a travel package*

### 5.5.3 User does the payment for booking a travel package



*Figure 98: State transition diagram for user dose the payment*

### 5.5.4 Delete profile for user, hotel, guide and vehicle owner

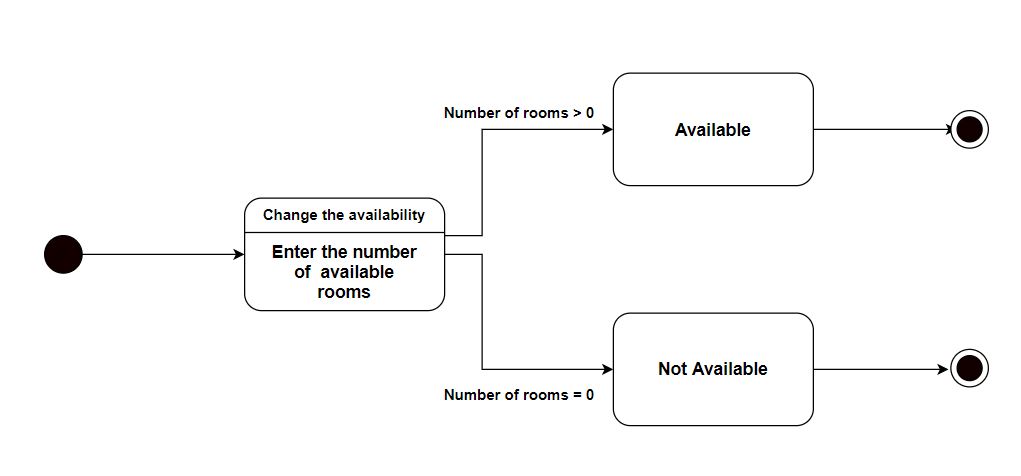


*Figure 99: State transition diagram for delete profile*

### 5.5.5 Change availability for guide and vehicle

*Figure 100: State transition diagram for changing the availability for guide and vehicle*

### 5.5.6 Change availability for hotel



*Figure 101: State transition diagram for changing the availability for hotel*

# 6. Interface Flow Diagrams

## 6.1 User’s interface flow diagram

*Figure 102: Interface flow diagram for the user*

## Admin’s interface flow diagram

*Figure 103: Interface flow diagram for the admin*

## Hotel account’s interface flow diagram

*Figure 104: Interface flow diagram for the hotel*

## Guide account’s interface flow diagram

*Figure 105: Interface flow diagram for the guide*

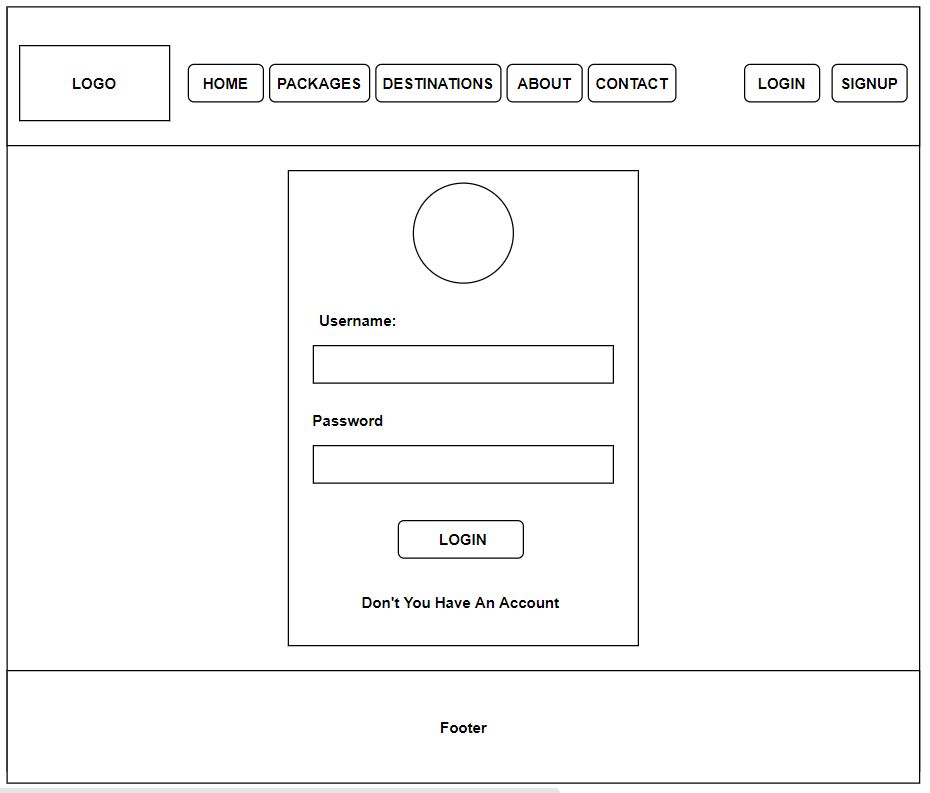
## Vehicle owner’s interface flow diagram

*Figure 106: Interface flow diagram for the vehicle owner*

# 7. Wired interfaces

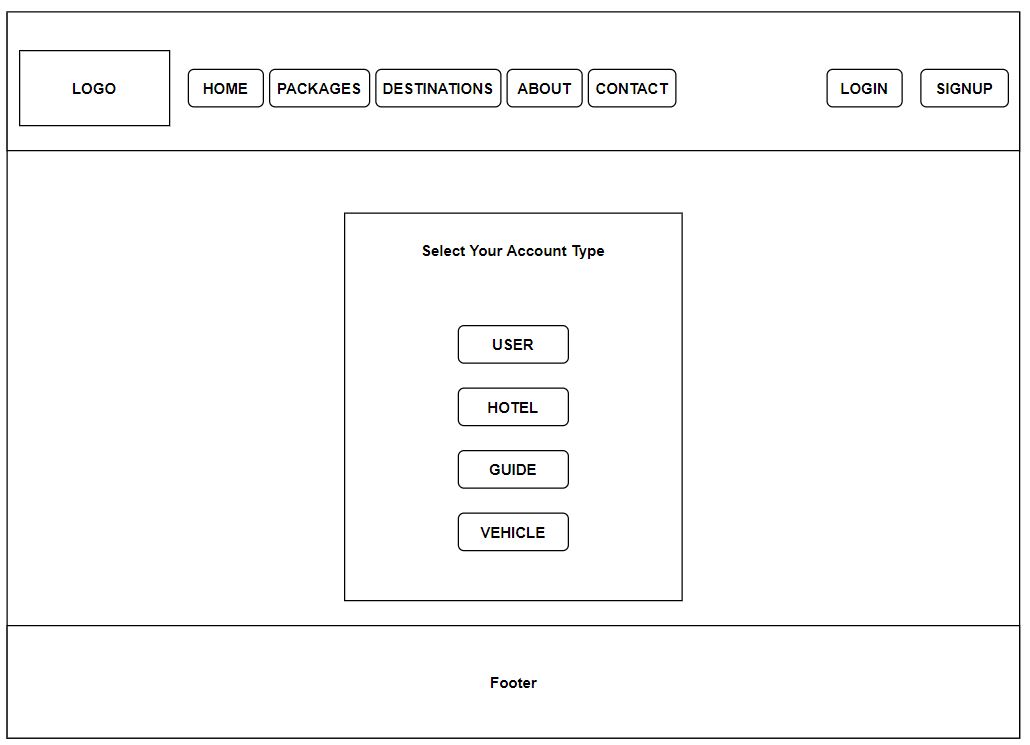
## 7.1 User’s interfaces

### 7.1.1 Login page



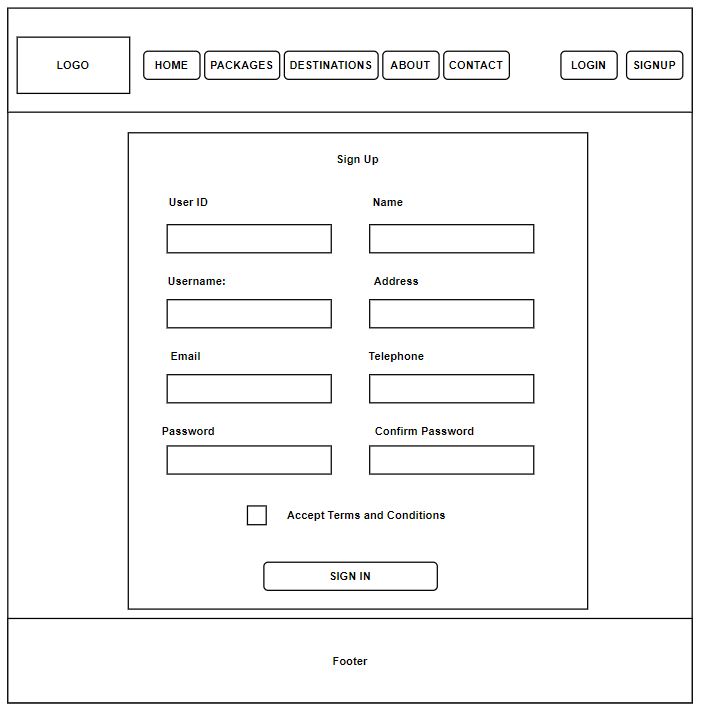
*Figure 107: Interface for user Login page*

### 7.1.2 Account page



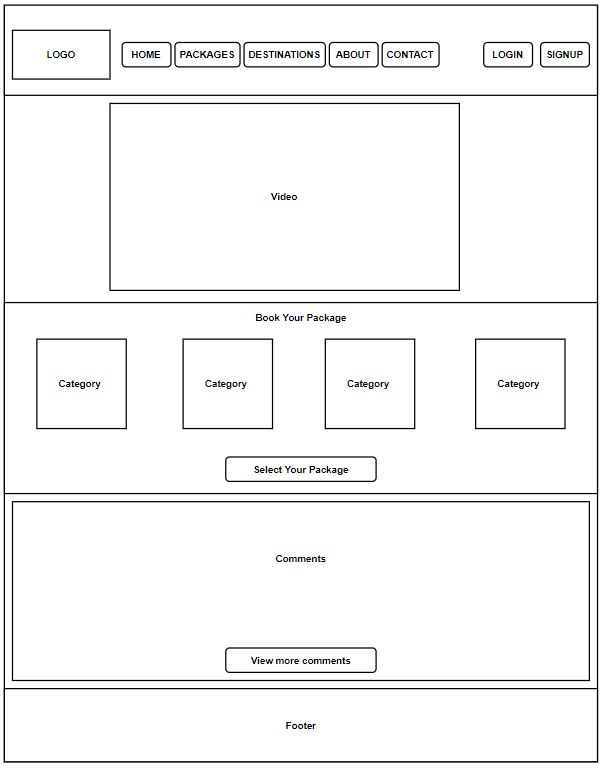
*Figure 108: Interface for user Account page*

### 7.1.3 Signup page



*Figure 109: Interface for user signup page*

### 7.1.4 Index page

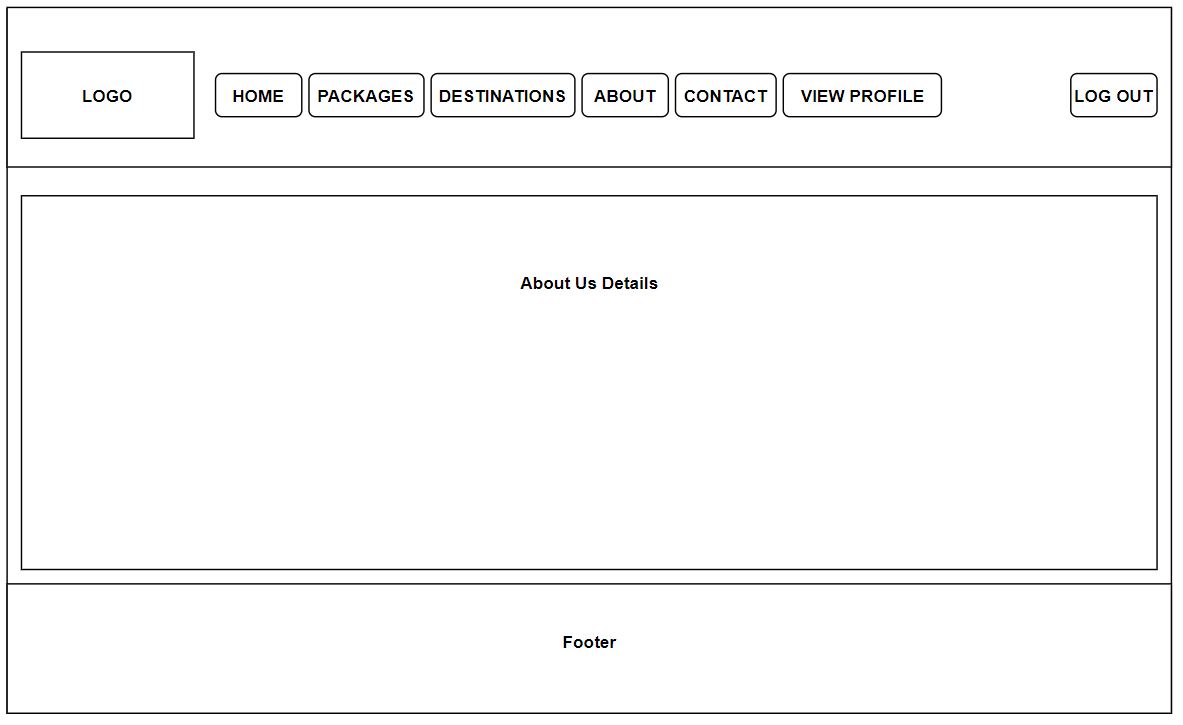


*Figure 110: Interface for Index page*

### 7.1.5 Home page

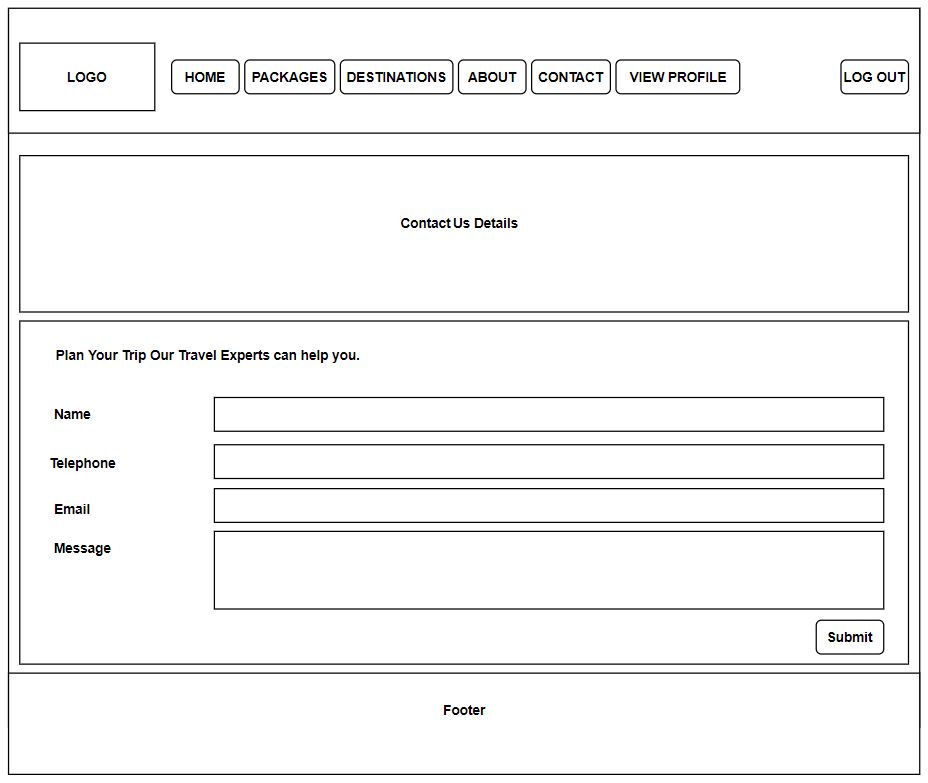
*Figure 111: Interface for user home page*

### 7.1.6 About page



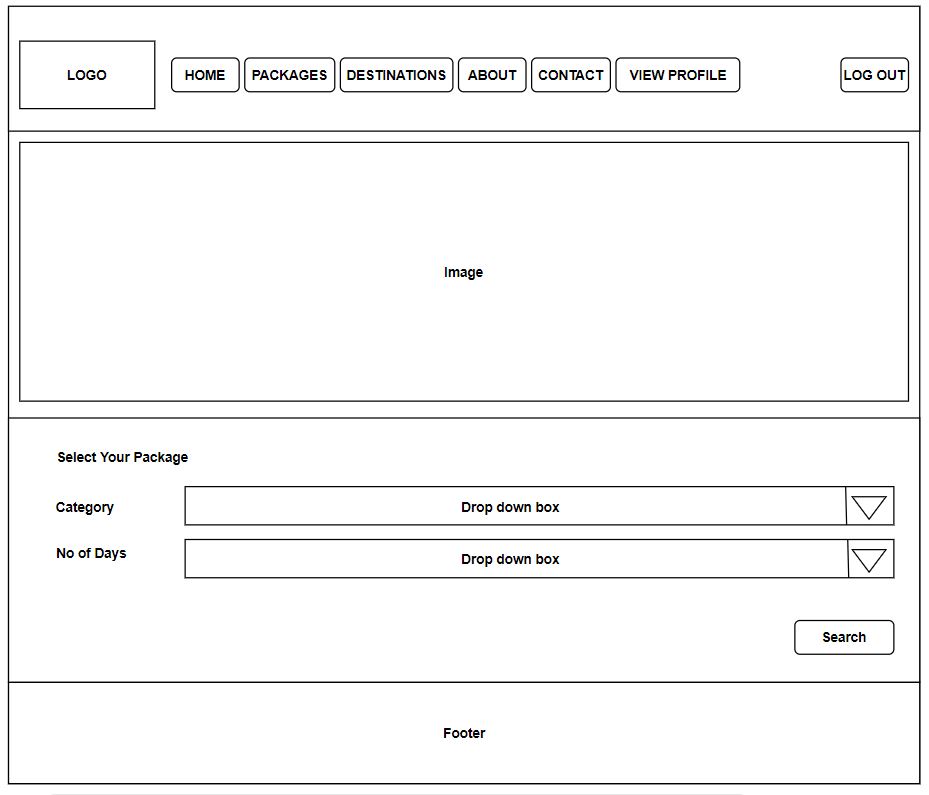
*Figure 112: Interface for About page*

### 7.1.7 Contact page



*Figure 113: Interface for Contact page*

### 7.1.8 Category page



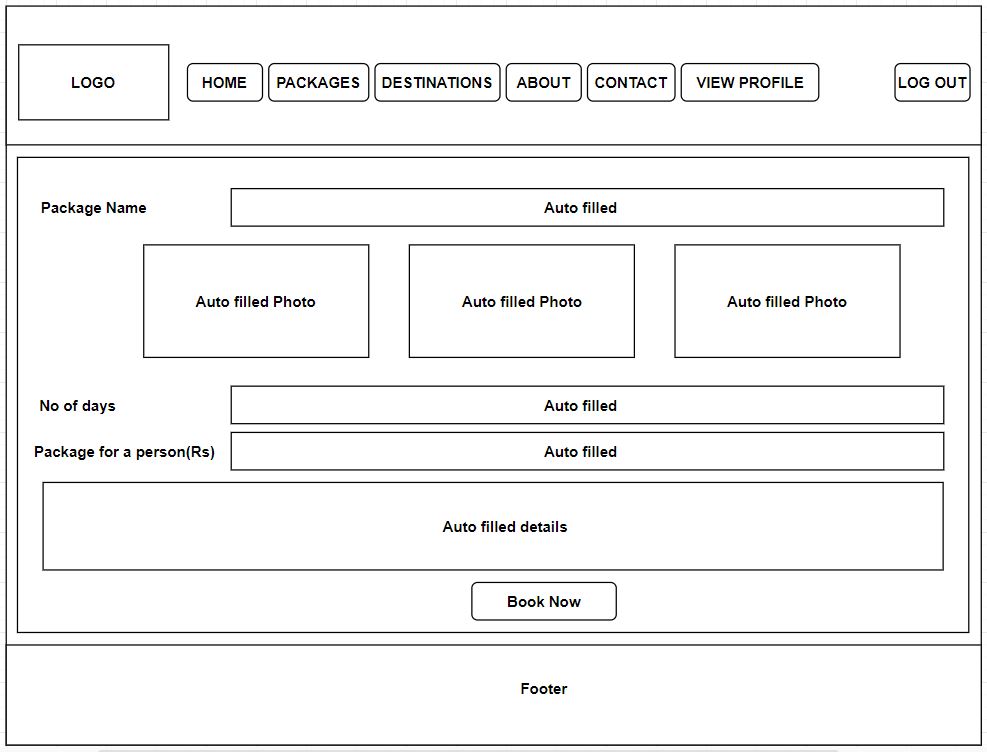
*Figure 114: Interface for Category page*

### 7.1.9 Package page



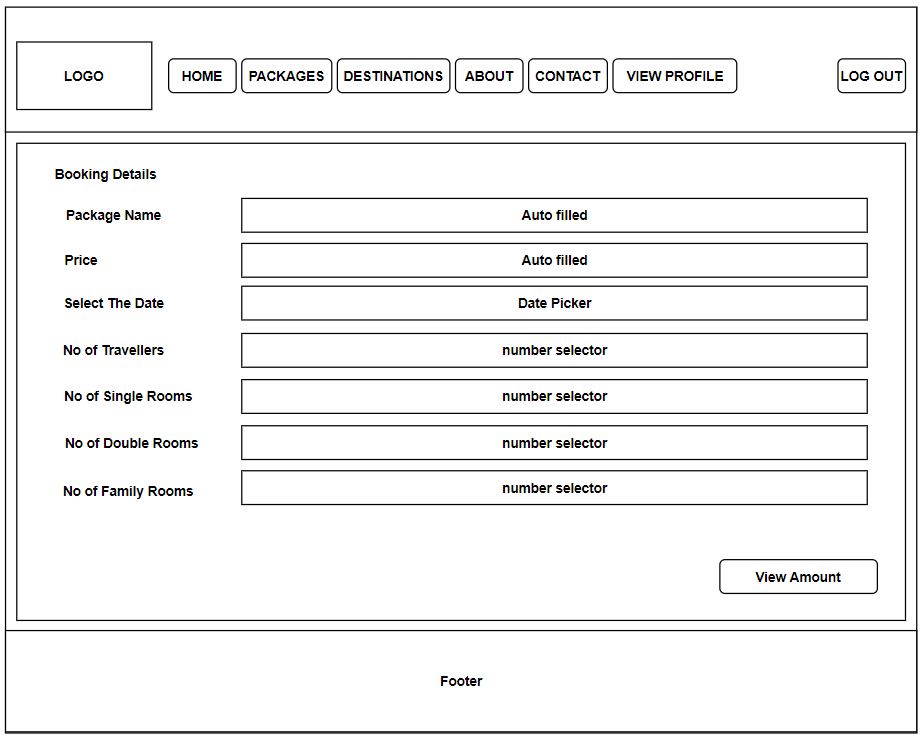
*Figure 115: Interface for Package page*

### 7.1.16 Package more details page



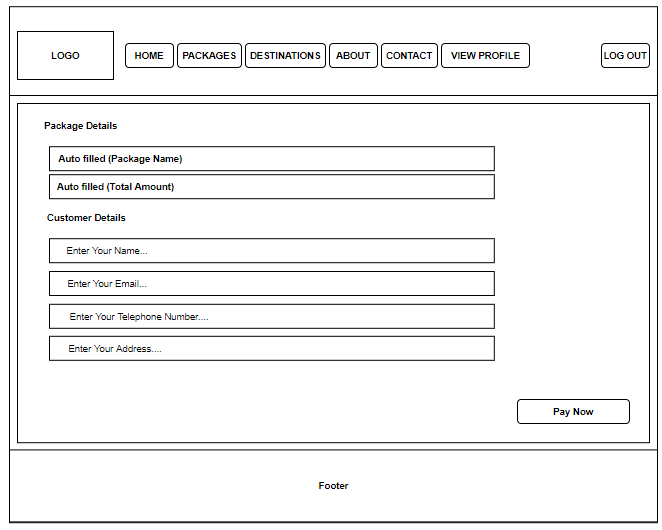
*Figure 116: Interface for package more details page*

### 7.1.10 Book page



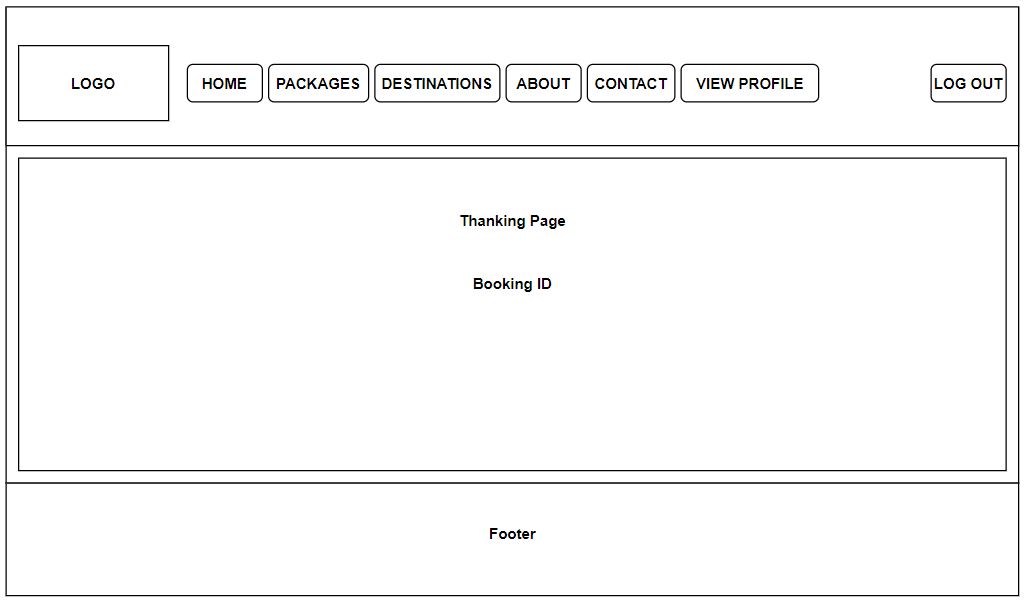
*Figure 117: Interface for Book page*

### 7.1.11 Payhere page



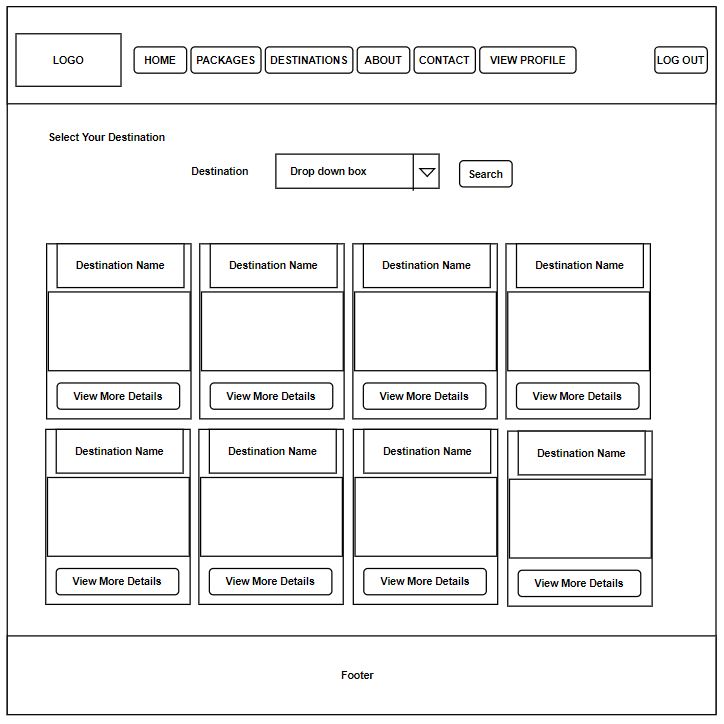
*Figure 118: Interface for Payhere page*

### 7.1.12 Thanking page



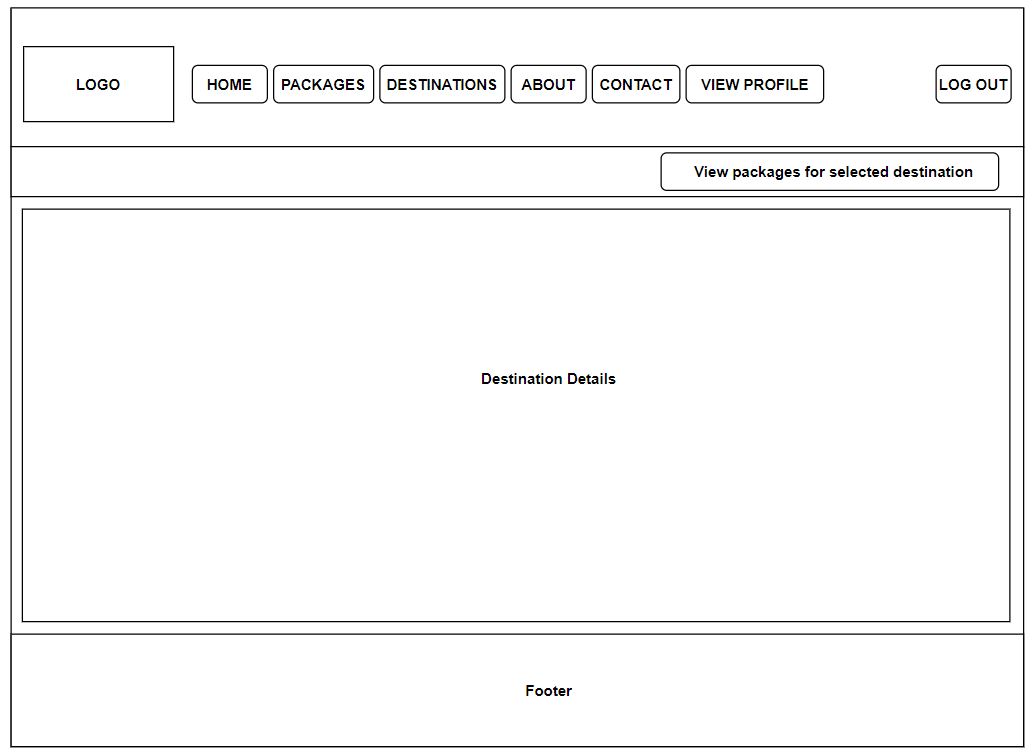
*Figure 119: Interface for Thanking page*

### 7.1.13 destination page



*Figure 120: Interface for Destination page*

### 7.1.14 Search destination page



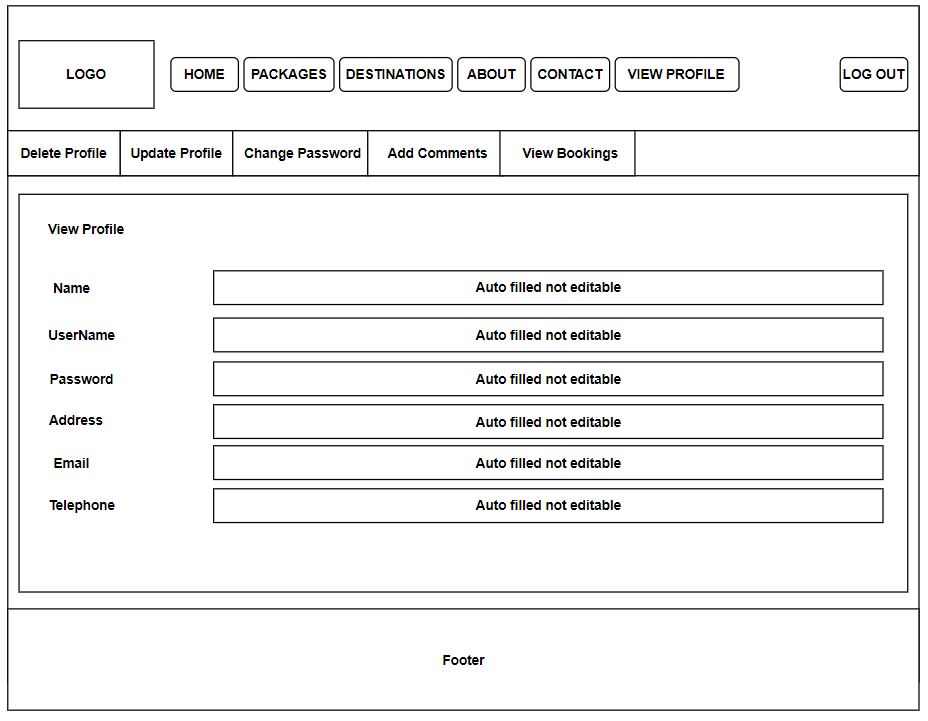
*Figure 121: Interface for Search Destination page*

### 7.1.15 Packages for destination page



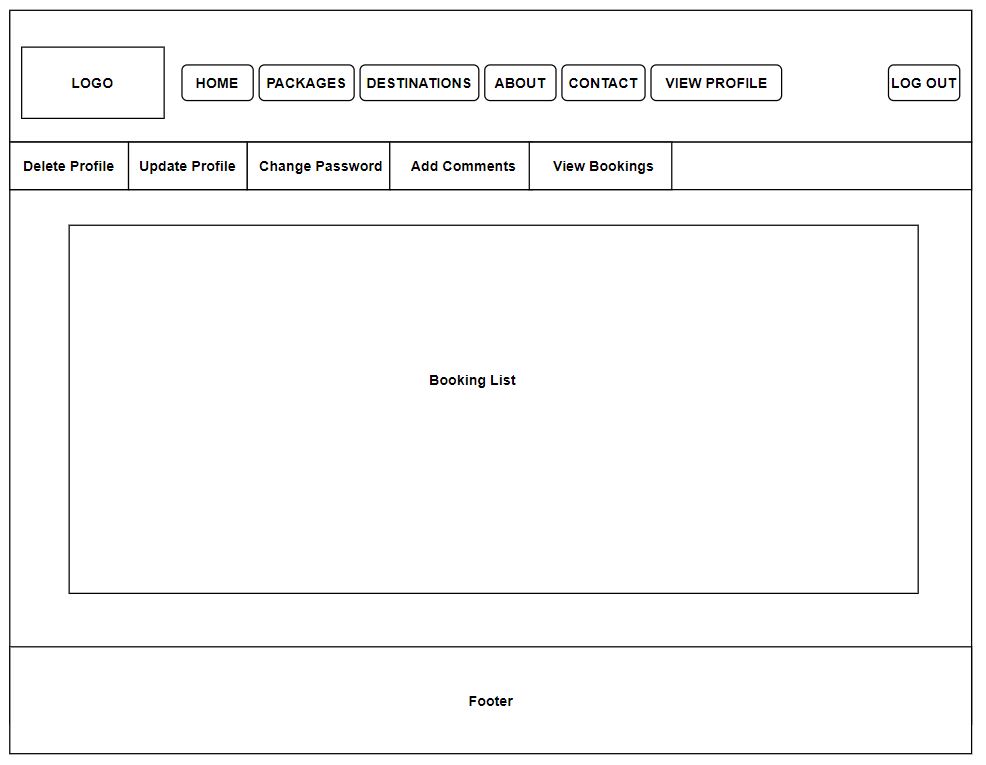
*Figure 122: Interface for Packages for destination page*

### 7.1.15 View profile page



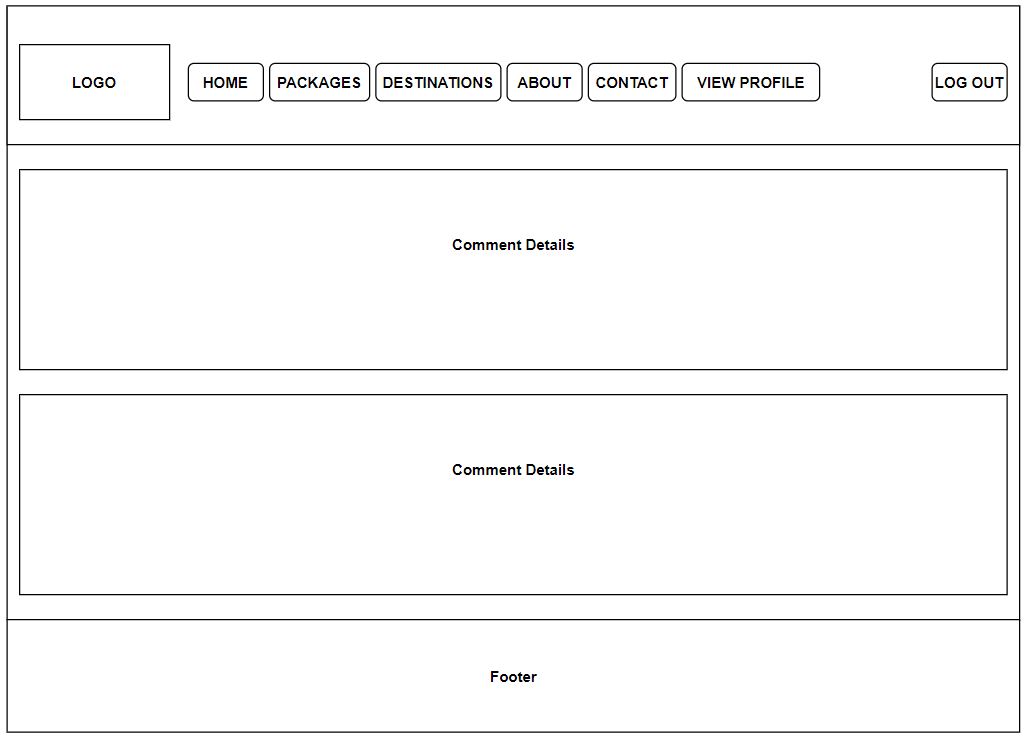
*Figure 123: Interface for user View profile page*

### 7.1.16 View booking page



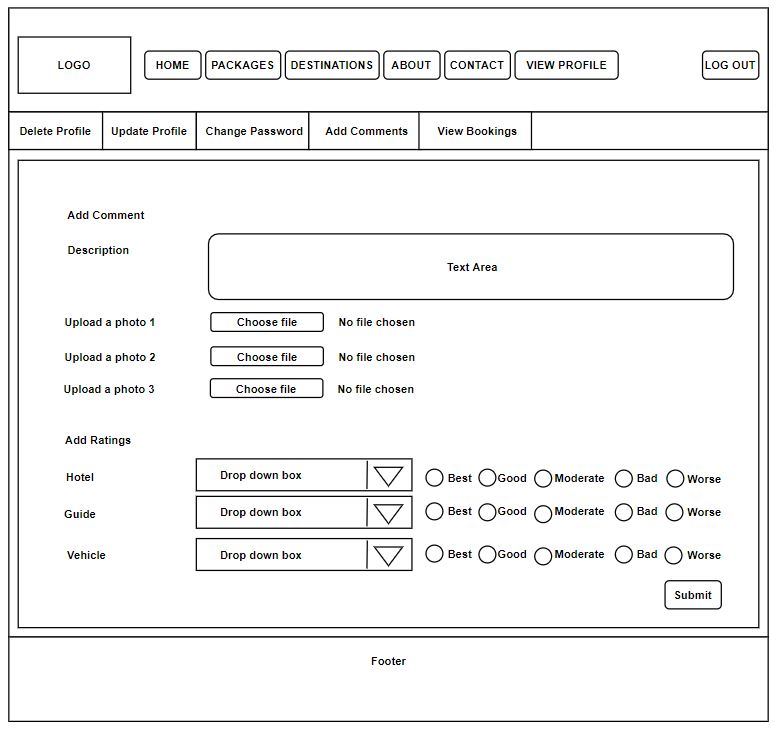
*Figure 124: Interface for user View booking*

### 7.1.17 View comment page



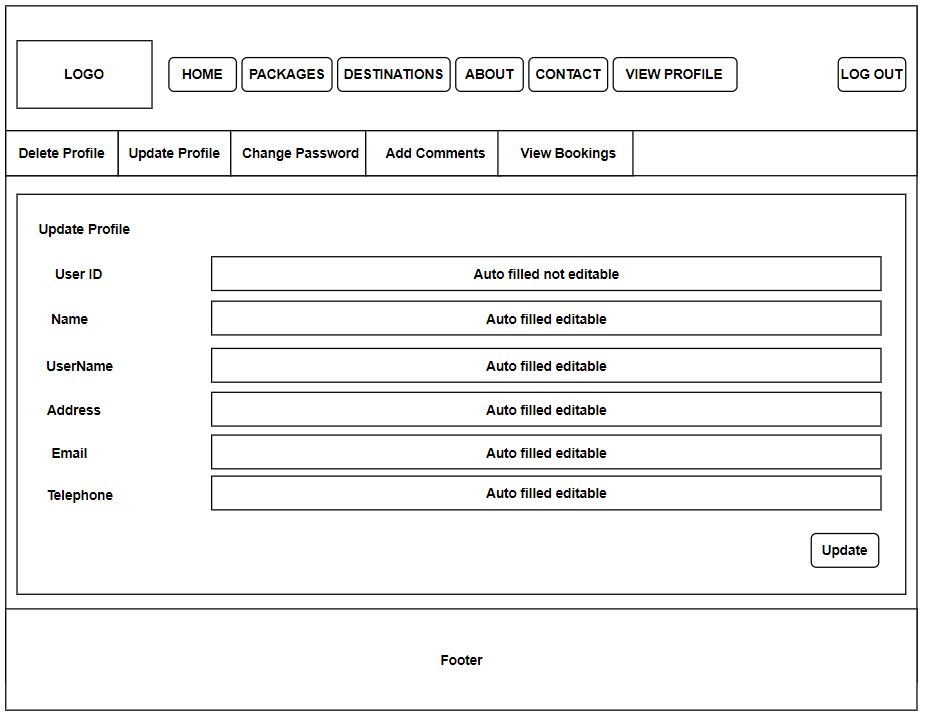
*Figure 125: Interface for View comment page*

### 7.1.18 Add comment page



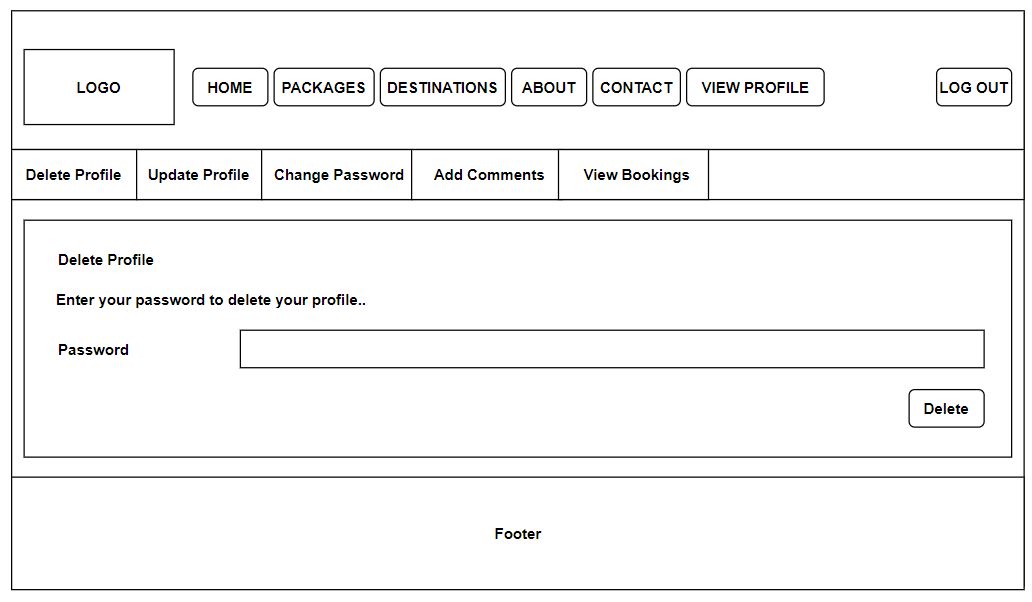
*Figure 126: Interface for Add comment page*

### 7.1.19 Update profile page



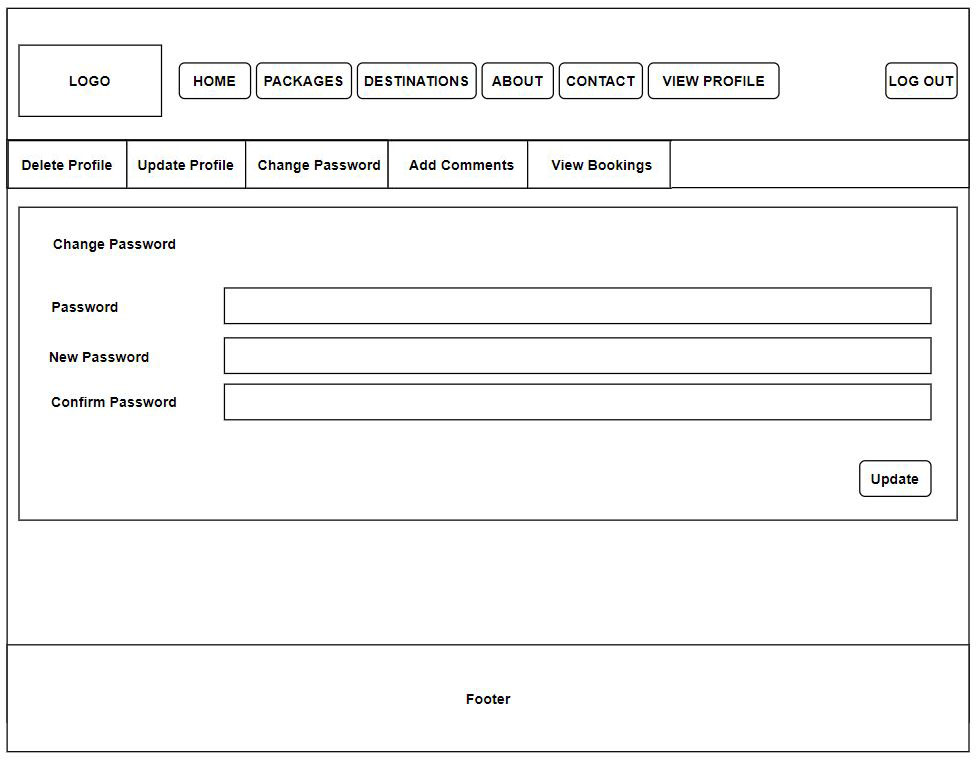
*Figure 127: Interface for Update user profile page*

### 7.1.20 Delete profile page



*Figure 128: Interface for user Delete profile*

### 7.1.21 Change the password page



*Figure 129: Interface for user Change password*

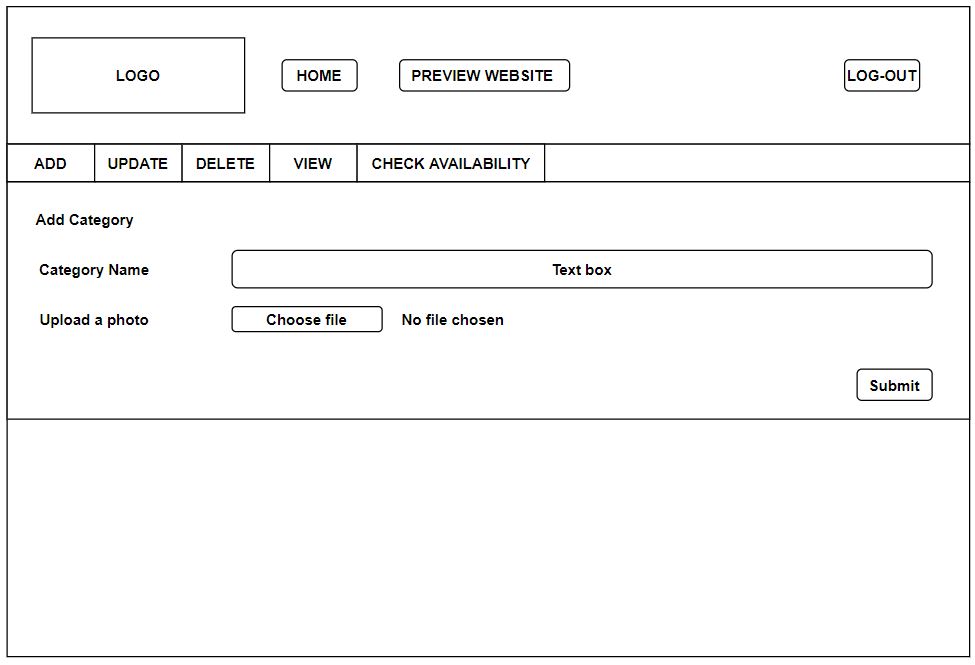
## 7.2 Admin’s interfaces

### 7.2.1 Home page



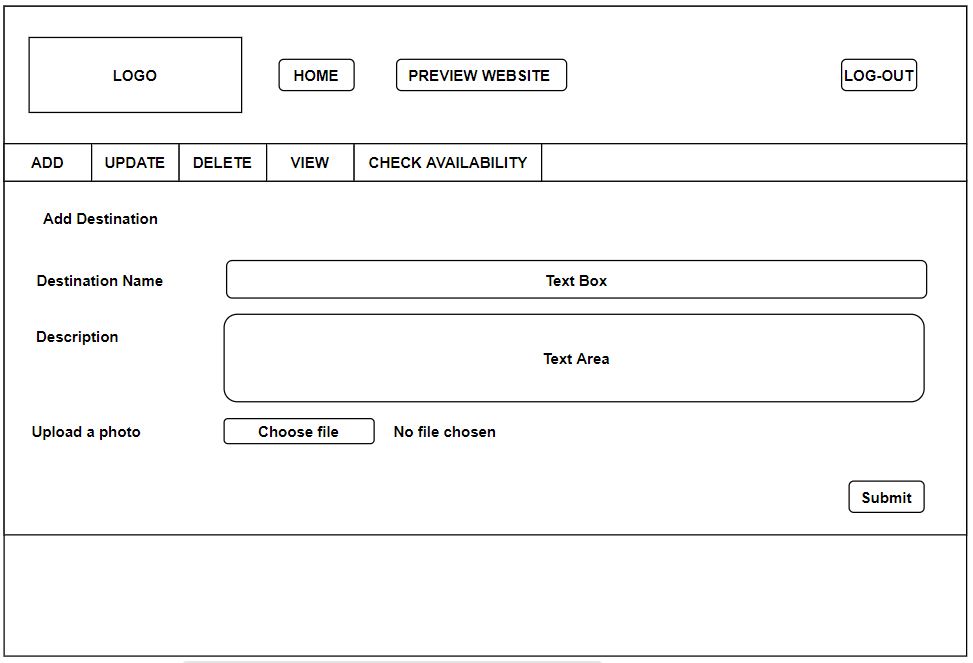
*Figure 130: Interface for admin Home page*

### 7.2.2 Add category page



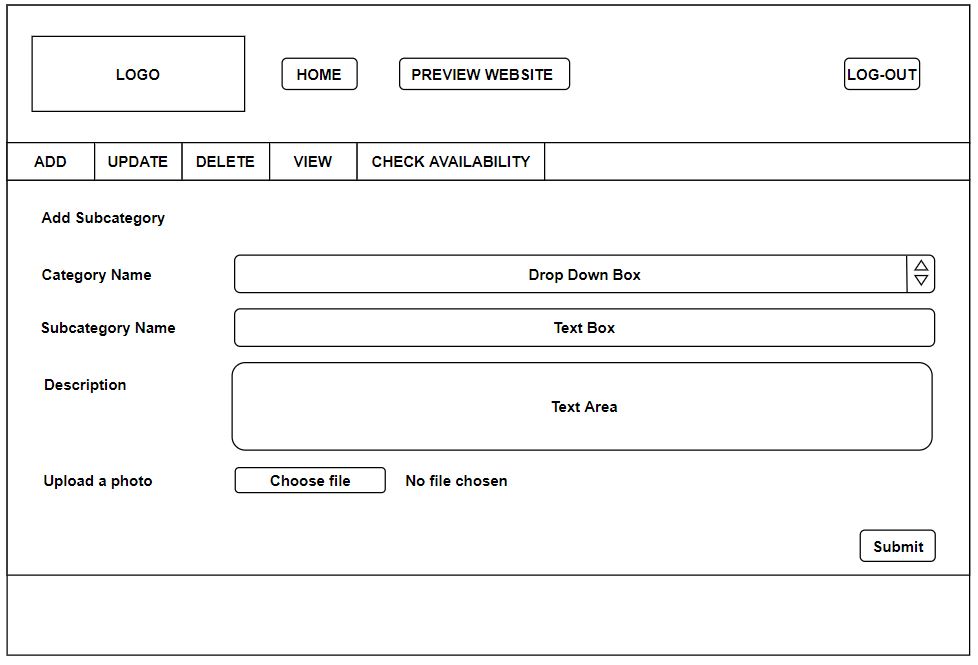
*Figure 131: Interface for admin add category page*

### 7.2.3 Add destination page



*Figure 132: Interface for admin add destination page*

### 7.2.4 Add Subcategory page

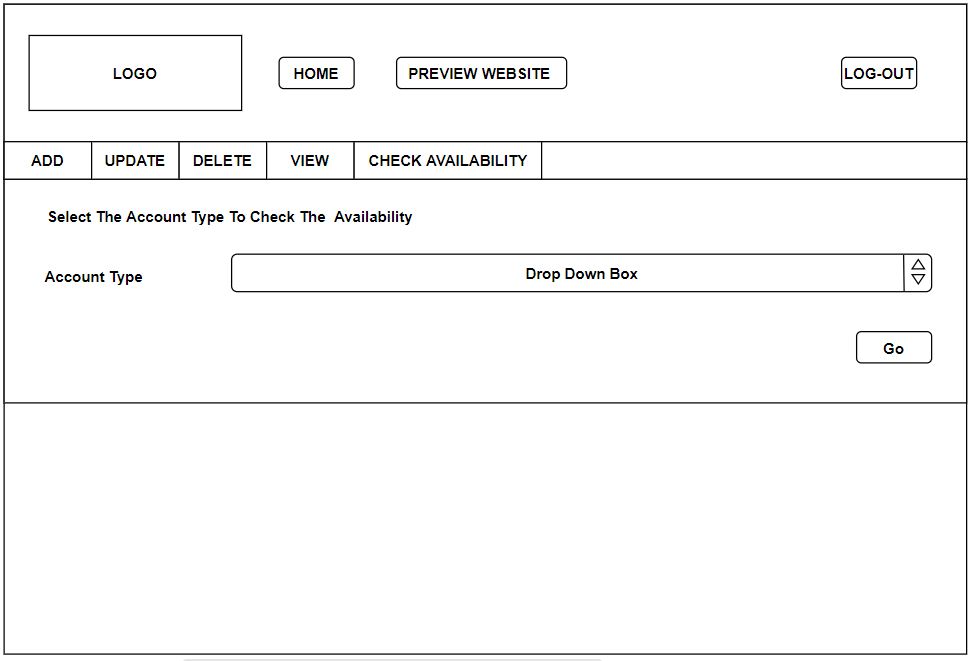


*Figure 133: Interface for admin add subcategory page*

### 7.2.5 Add package page

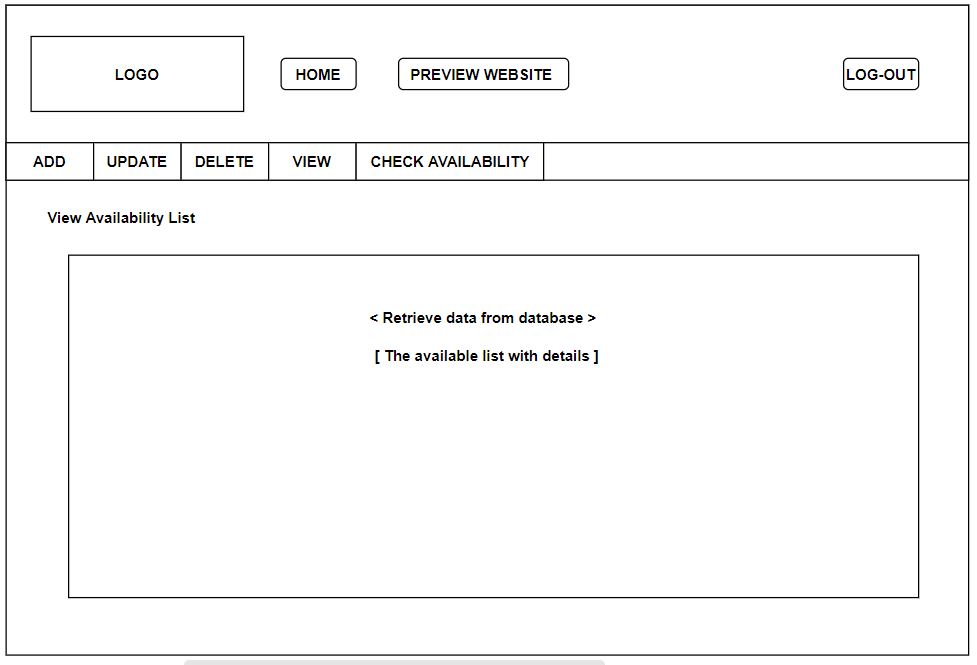
*Figure 134: Interface for admin add package page*

### 7.2.6 Check availability page1



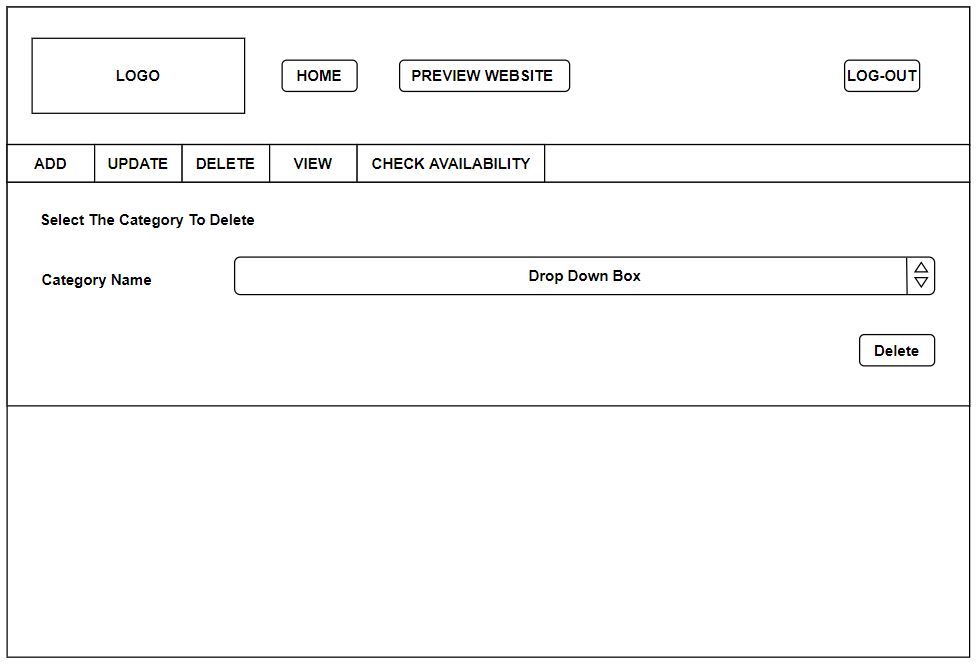
*Figure 135: Interface for admin checks availability page1*

### 7.2.7 Check availability page2



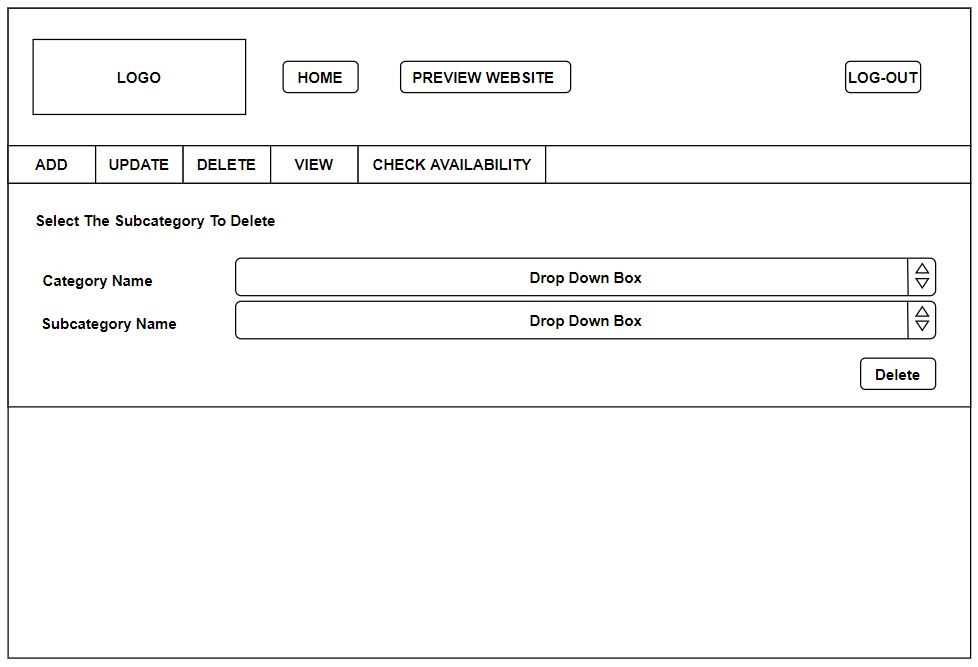
*Figure 136: Interface for admin checks availability page2*

### 7.2.8 Delete category page



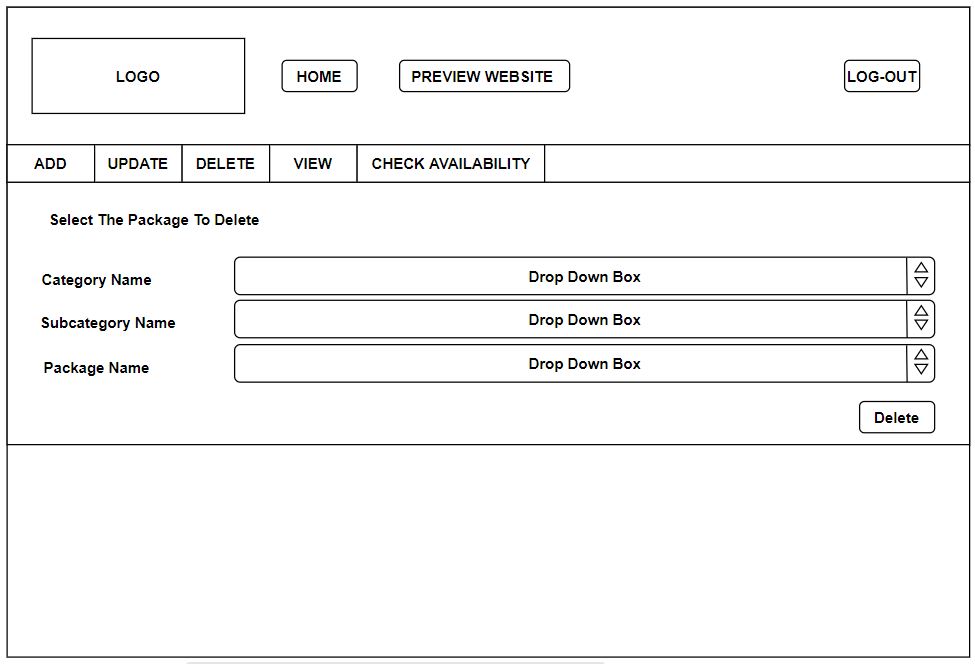
*Figure 137: Interface for admin deletes category page*

### 7.2.9 Delete subcategory page



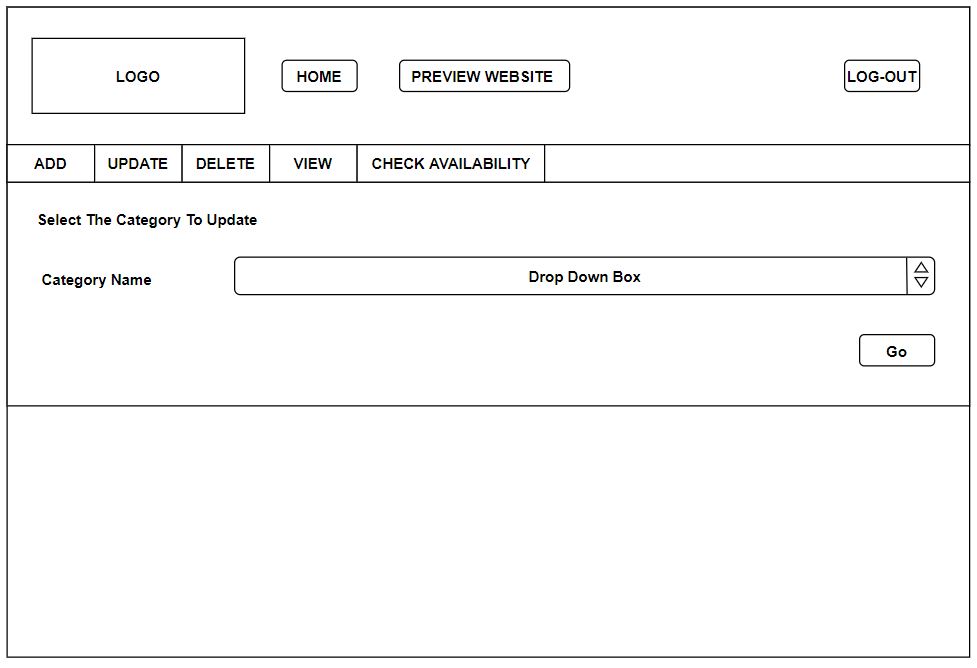
*Figure 138: Interface for admin deletes subcategory page*

### 7.2.10 Delete package page



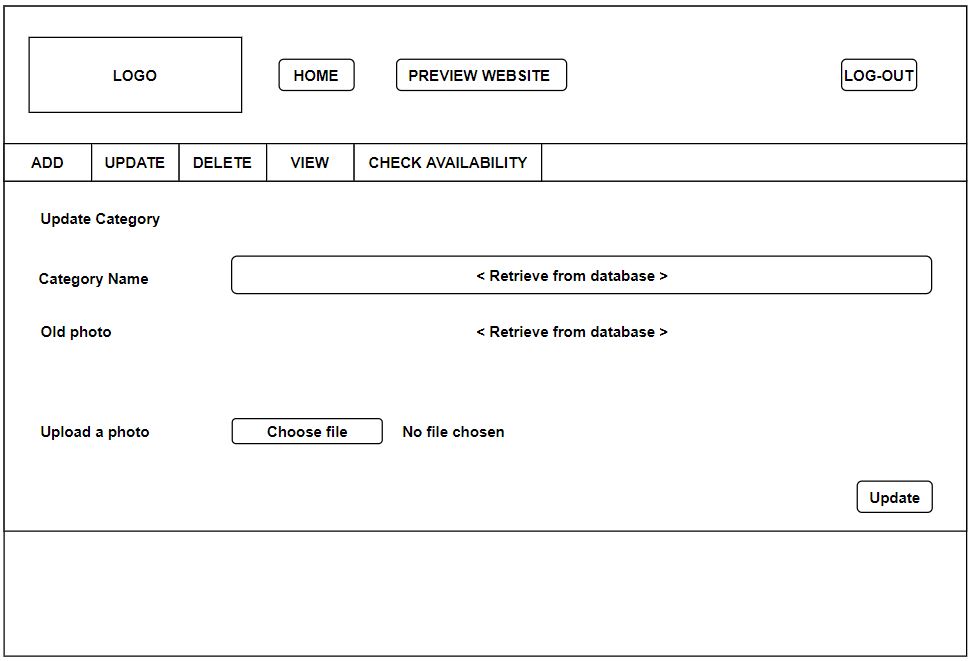
*Figure 139: Interface for admin deletes package page*

### 7.2.11 Update Category page1



*Figure 140: Interface for admin update category page1*

### 7.2.12 Update category page2



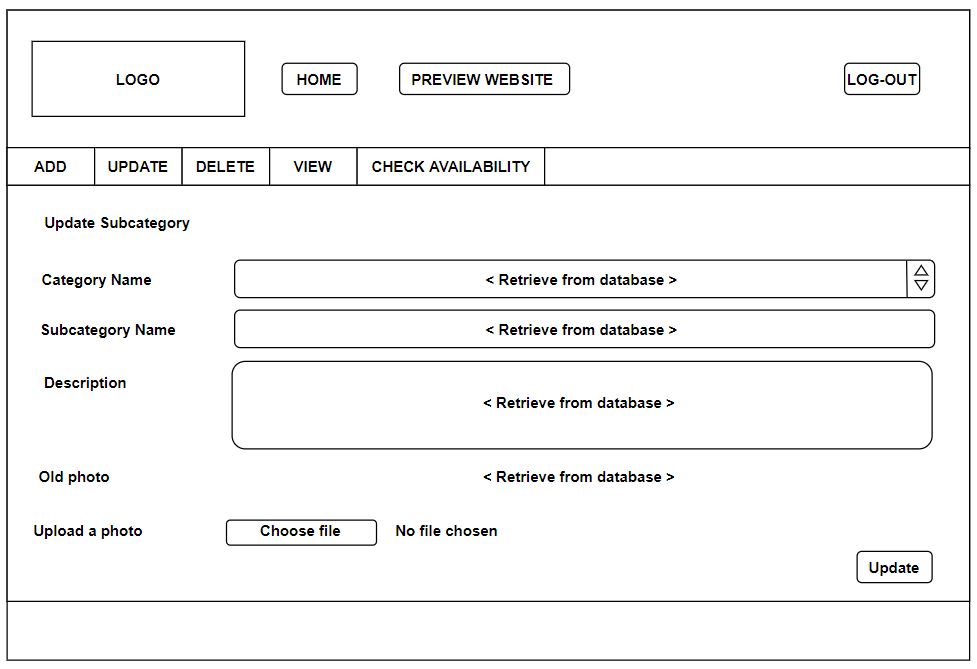
*Figure 141: Interface for admin update category page2*

### 7.2.13 Update subcategory page1



*Figure 142: Interface for admin update subcategory page1*

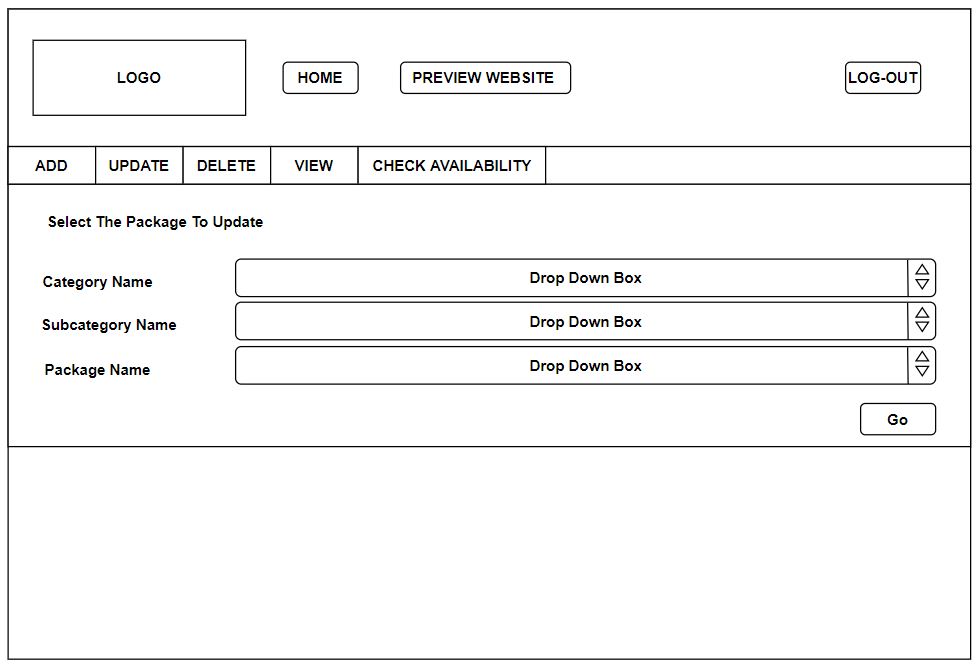
### 7.2.14 Update subcategory page2



### 

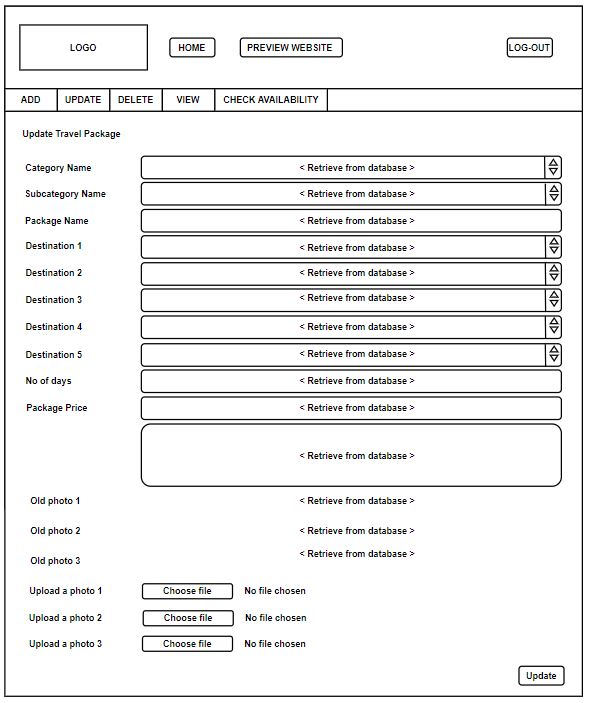
*Figure 143: Interface for admin update subcategory page2*

### 7.2.15 Update package page1



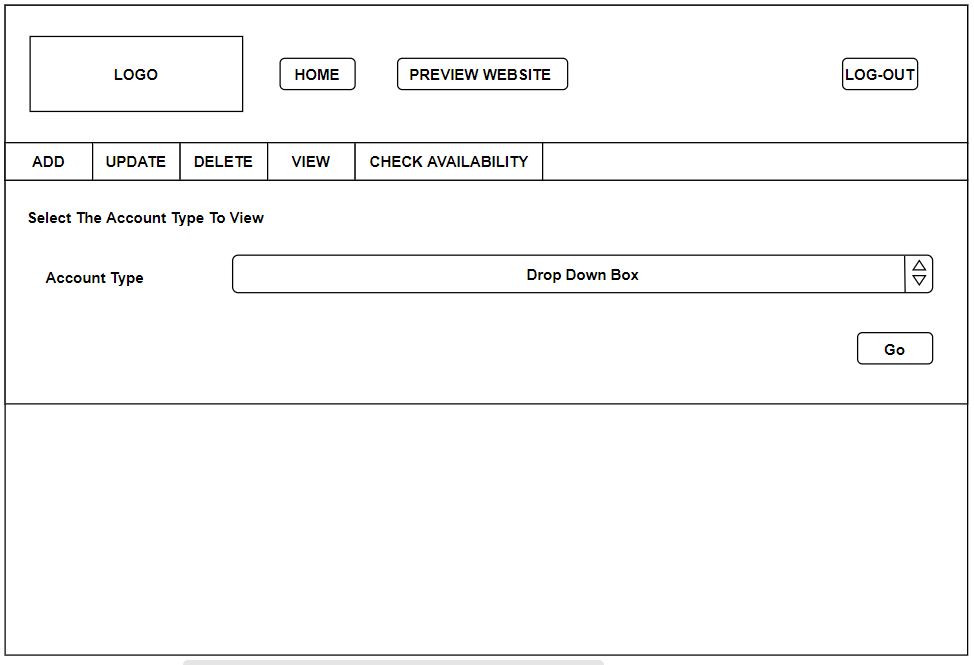
*Figure 144: Interface for admin update package page1*

### 7.2.16 Update package page2



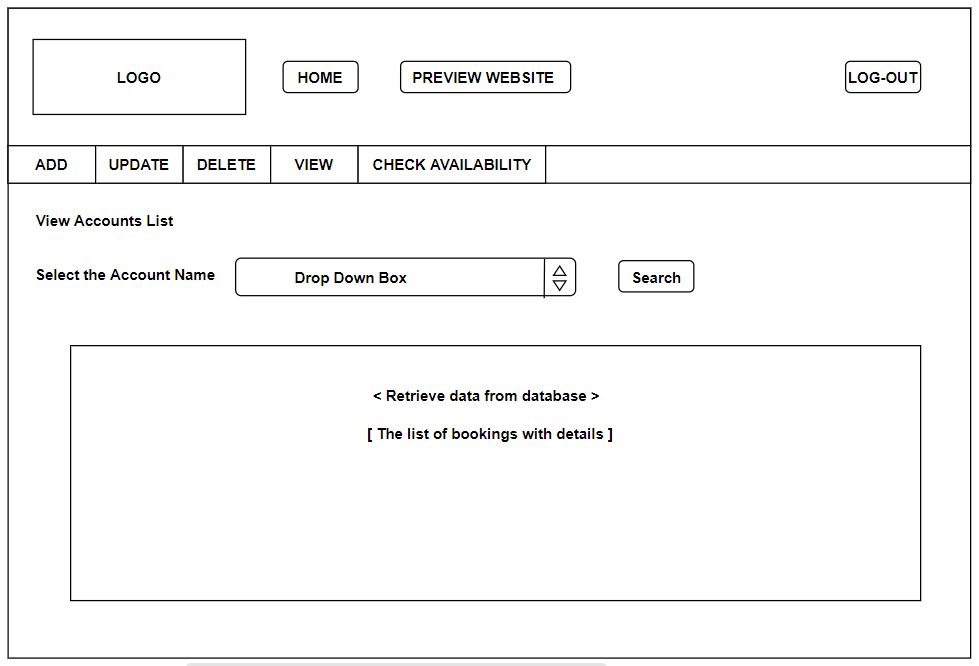
*Figure 145: Interface for admin update package page2*

### 7.2.17 View account page1



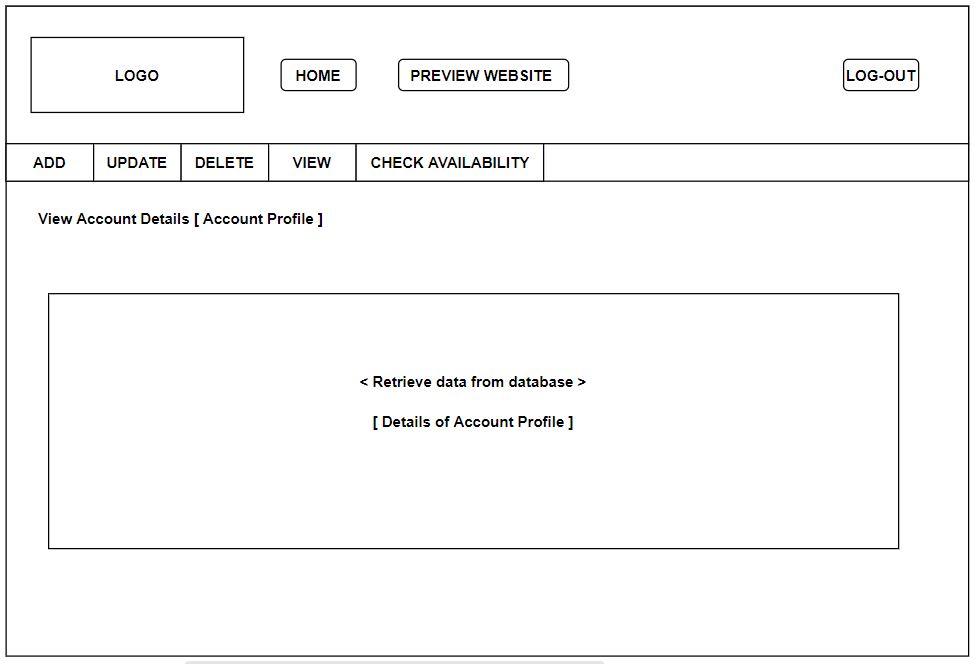
*Figure 146: Interface for admin view account page1*

### 7.2.18 View account page2



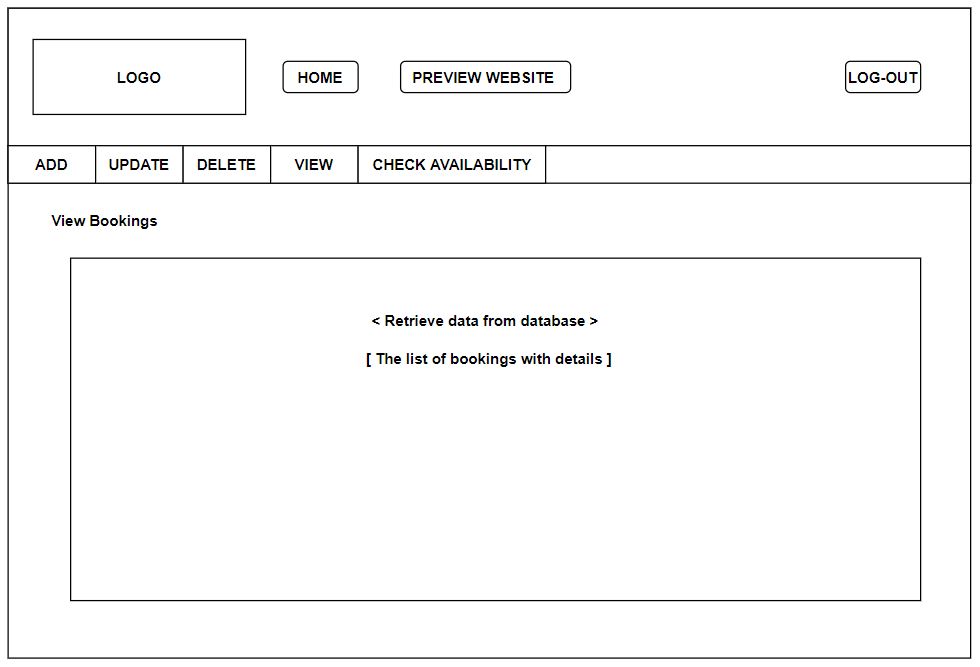
*Figure 147: Interface for admin view account page2*

### 7.2.19 View account page3



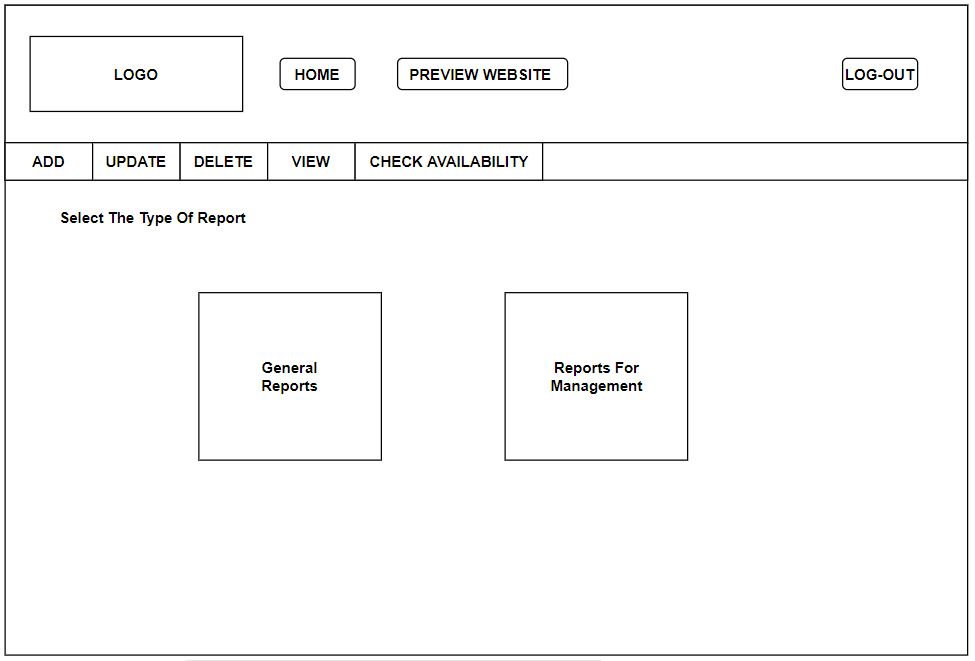
*Figure 148: Interface for admin view account page3*

### 7.2.20 View booking page



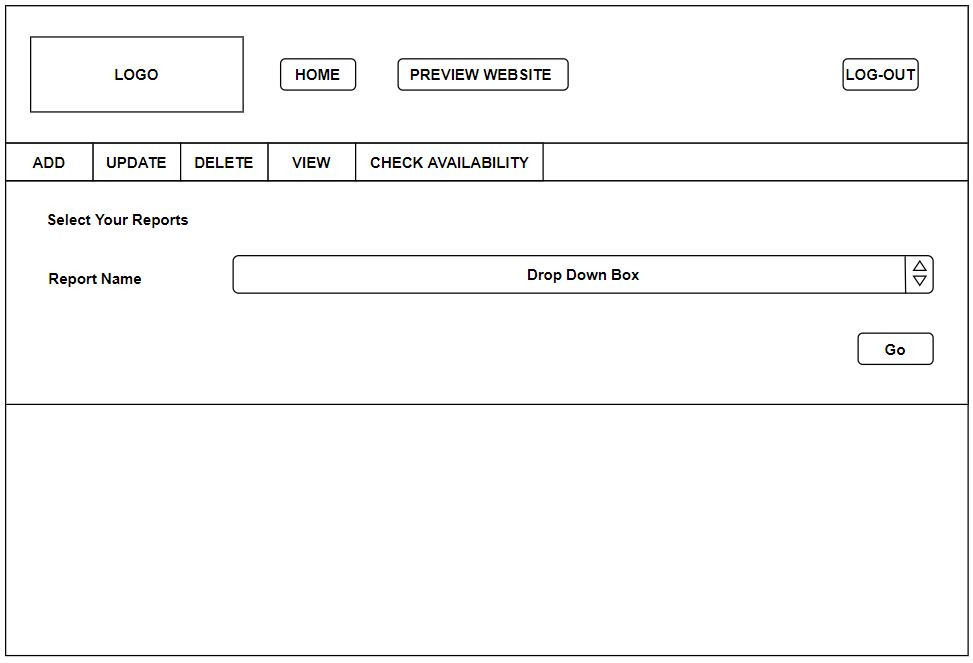
*Figure 149: Interface for admin view booking page*

### 7.2.21 View report page1



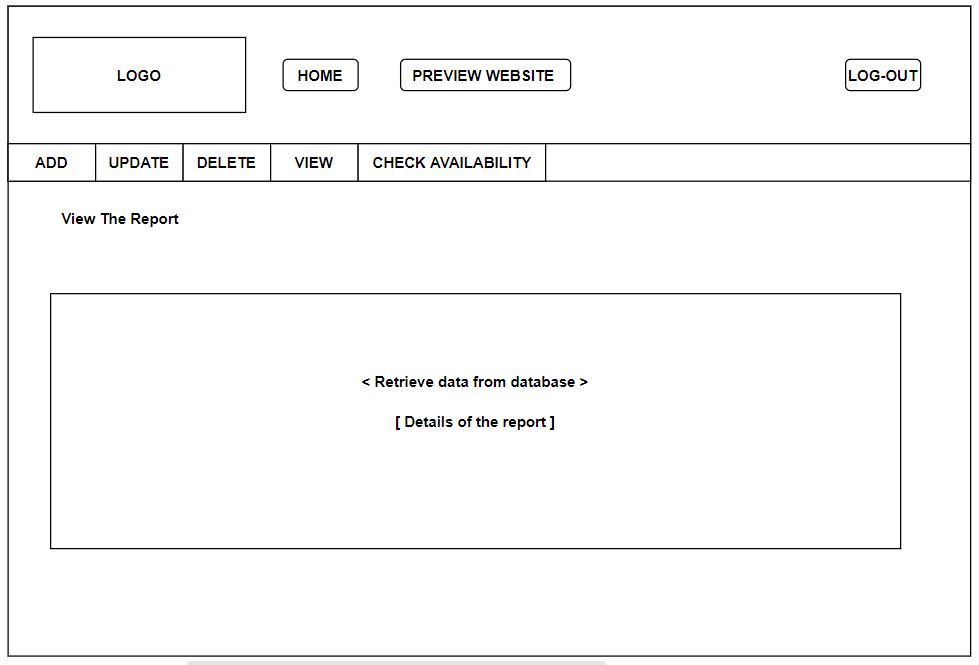
*Figure 150: Interface for admin view report page1*

### 7.2.22 View report page2



*Figure 151: Interface for admin view report page2*

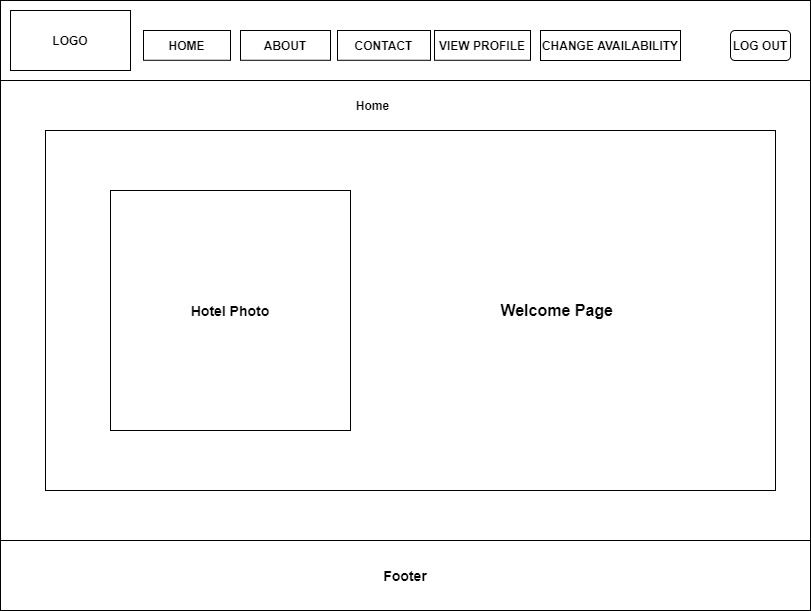
### 7.2.23 View report page3



*Figure 152: Interface for admin view report page3*

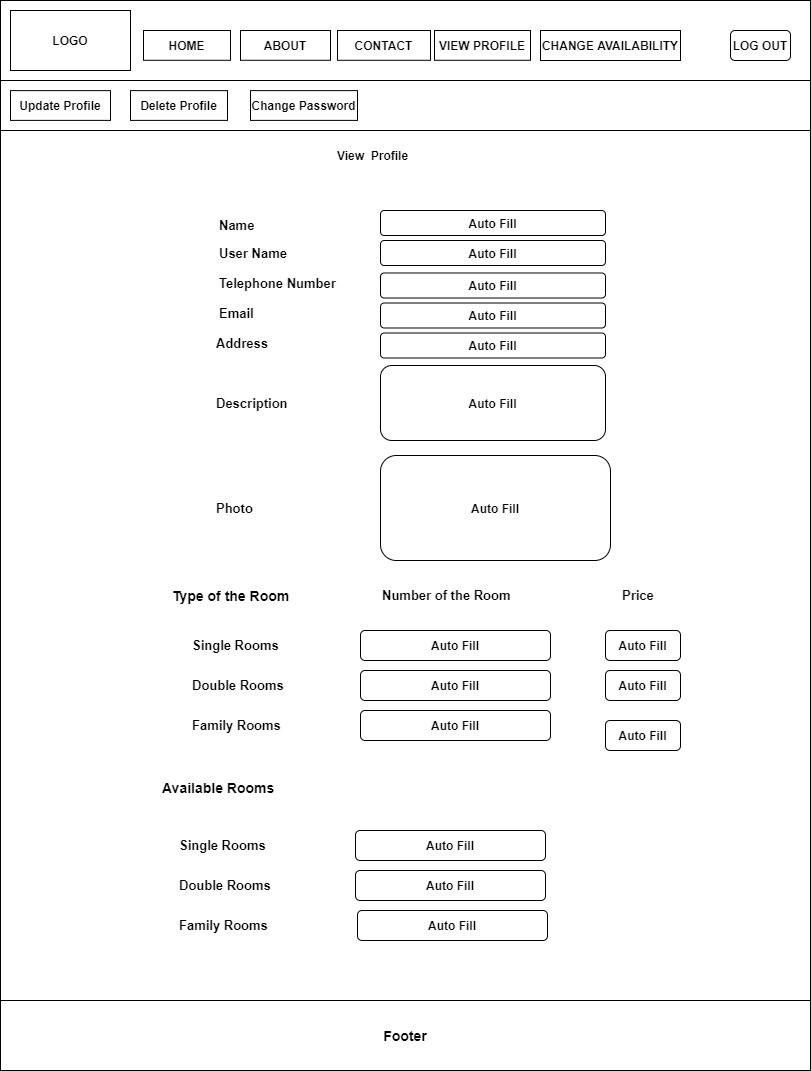
## 7.3 Hotel account’s interfaces

### 7.3.1 Home page



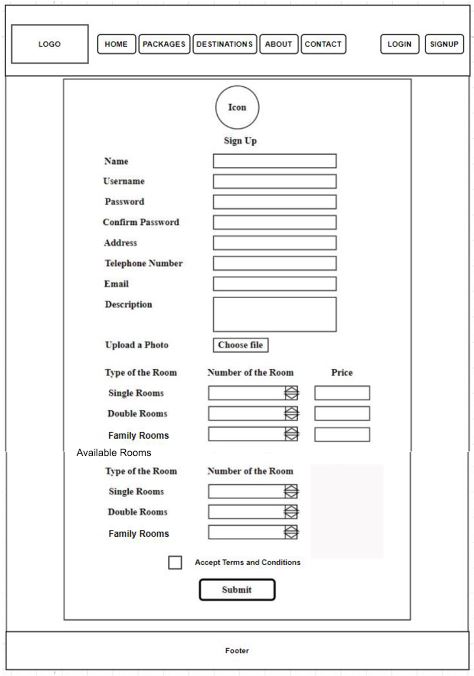
*Figure 153: Interface for hotel home page*

### 7.3.2 View profile page



*Figure 154: Interface for hotel view profile page*

### **7.3.3 Signup** page



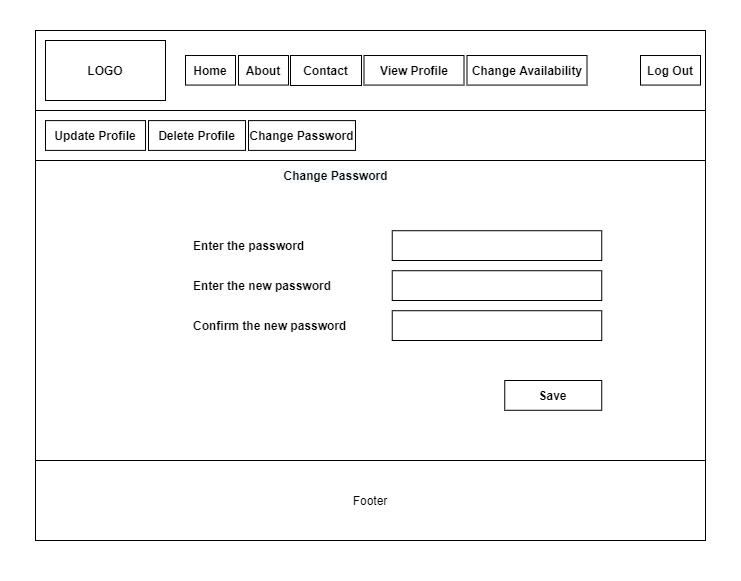
*Figure 155: Interface for hotel signup*

### 7.3.4 Update Profile page



*Figure 156: Interface for hotel update profile*

### 7.3.5 Change Password page

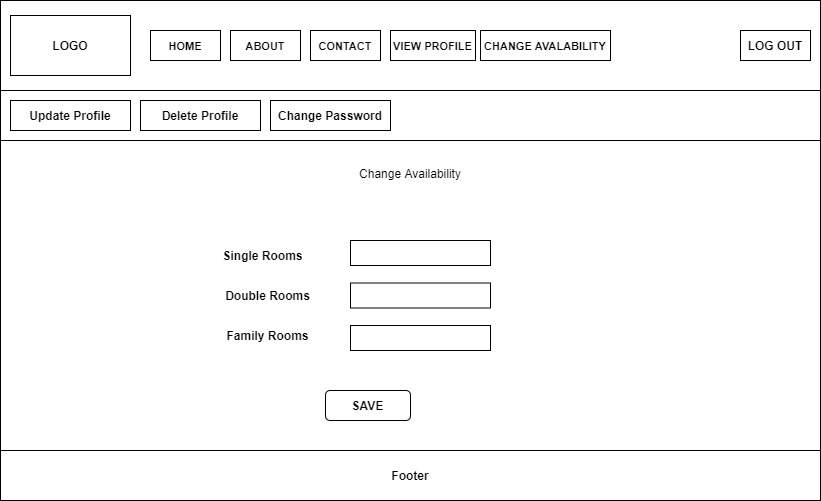


*Figure 157: Interface for hotel change password*

### 7.3.6 Delete Profile page

*Figure 158: Interface for hotel delete profile page*

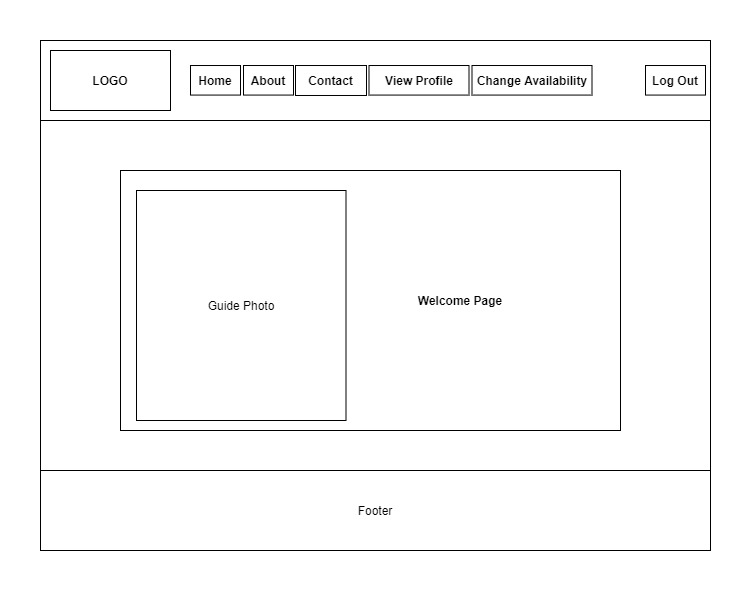
### 7.3.7 Change availability page



*Figure 159: Interface for hotel change availability page*

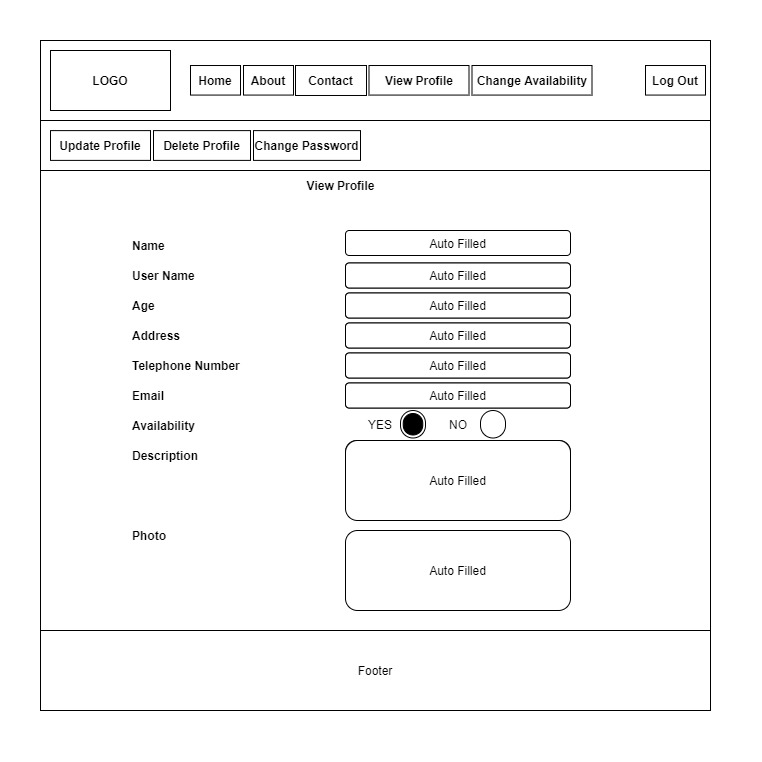
## 7.4 Guide account’s interfaces

### 7.4.1 Home page



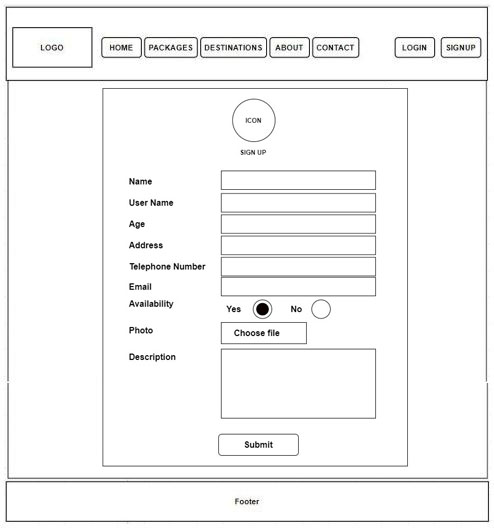
*Figure 160: Interface for guide home page*

### 7.4.2 View profile page



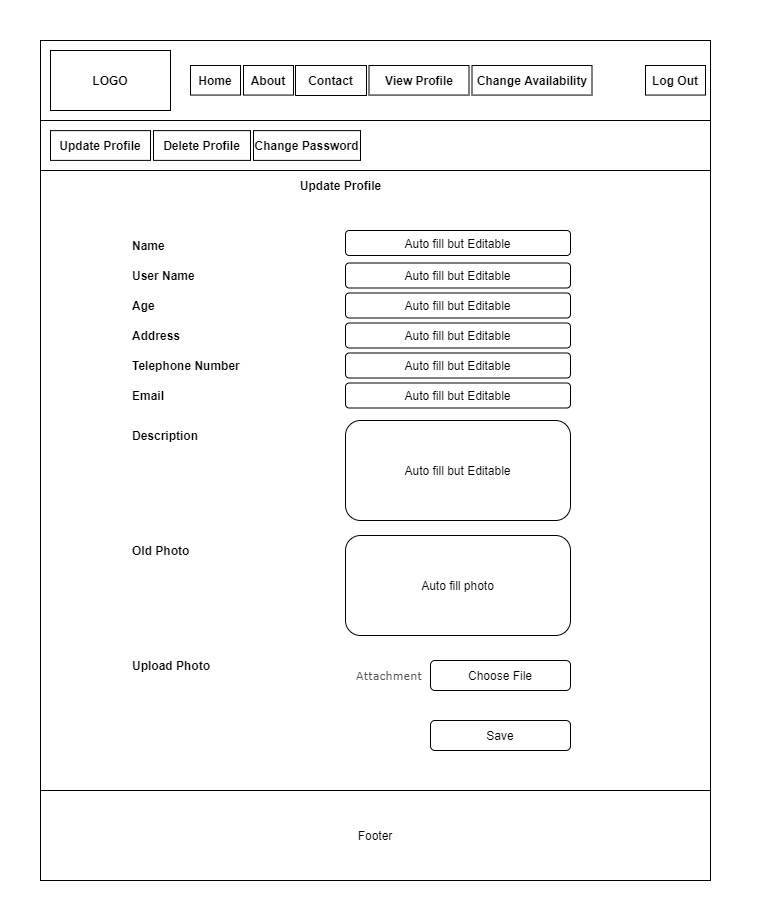
*Figure 161: Interface for guide view profile page*

### 7.4.3 Signup page



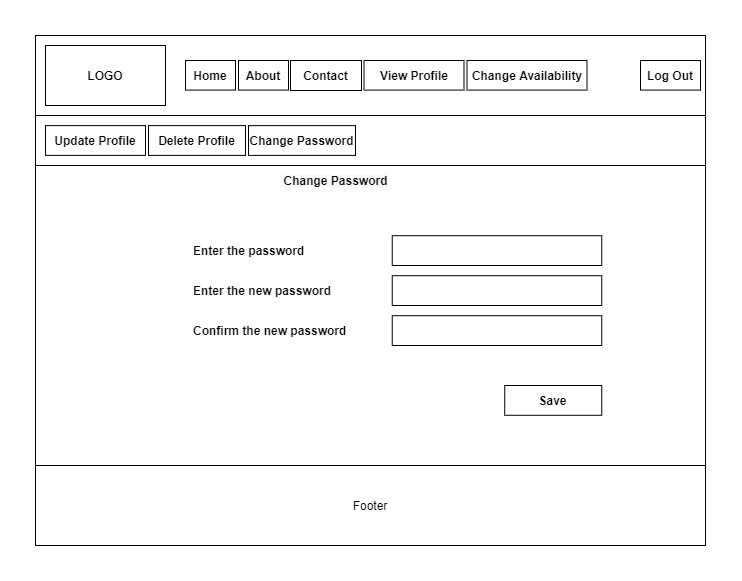
*Figure 162: Interface for guide signup page*

### 7.4.4 Update profile page



*Figure 163: Interface for guide update profile page*

### 7.4.5 Change password page



*Figure 164: Interface for guide change password page*

### 7.4.6 Delete profile



*Figure 165: Interface for guide delete profile page*

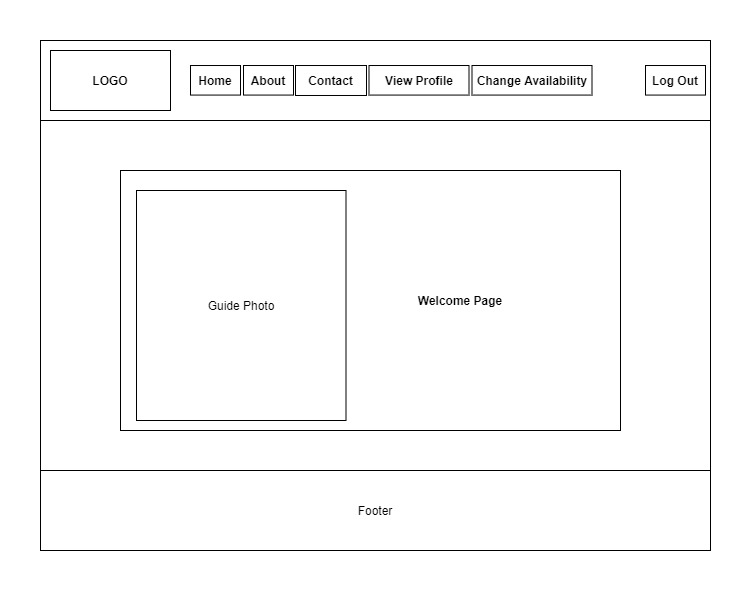
### 7.4.7 Change availability page



*Figure 166: Interface for guide change availability page*

## 7.5 Vehicle owner’s interfaces

### 7.5.1 Home page



*Figure 167: Interface for vehicle owner’s home page*

### 7.5.2 View profile page

*Figure 168: Interface for vehicle owner’s view profile page*

### 7.5.3 Signup page

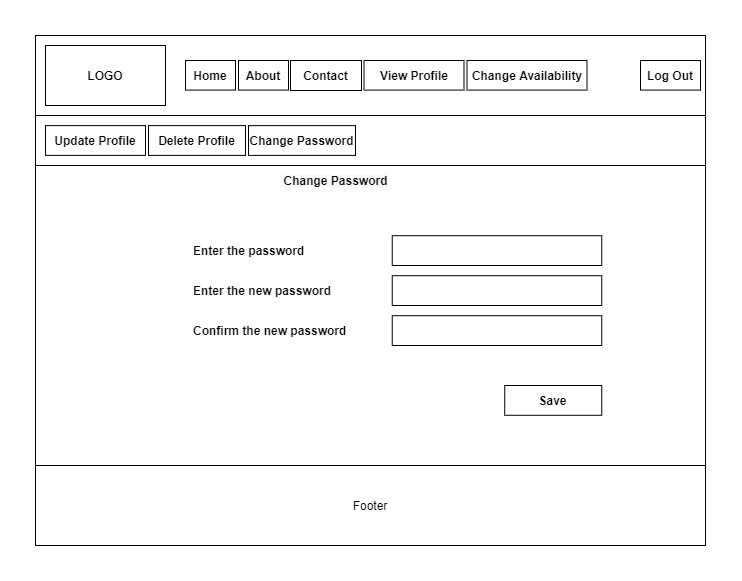
### 

*Figure 169: Interface for vehicle owner’s signup page*

### 7.5.4 Update profile page

*Figure 170: Interface for vehicle owner’s update profile page*

### 7.5.5 Change password page



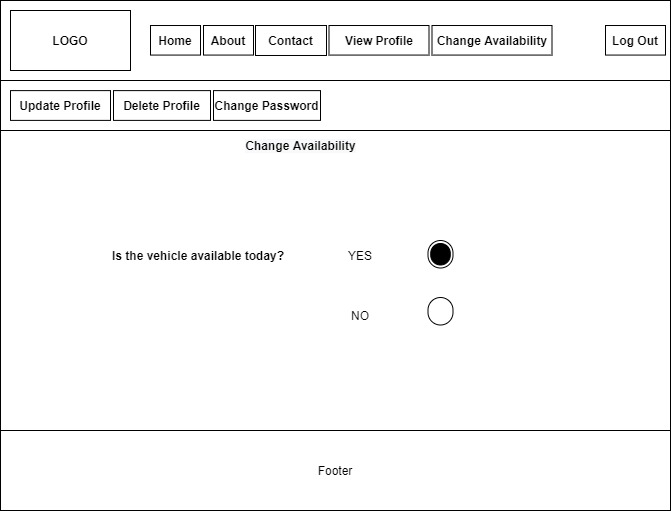
*Figure 171: Interface for vehicle owner’s change password page*

### 7.5.6 Delete profile page



*Figure 172: Interface for vehicle owner’s delete profile page*

### 7.5.7 Change availability page



*Figure 173: Interface for vehicle owner’s change availability page*

# 8. Declaration

We as members of the project titled Travel Management System, Certify that we will carry out this project according to the guidelines provided by the coordinators and supervisors of the course as well as we will not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university. To the best of our knowledge and brief, the project work will not contain any material previously published or written by another person or ourselves except where due reference is made in the text of appropriate places.

|  |  |  |
| --- | --- | --- |
| Index Number | Name of the Student | Signature |
| Asini Pathmila Silva | 18020798 |  |
| Ruwanthi Hemachandra | 18020291 |  |
| Hansaka Sandaruwan | 18020704 |  |
| Sachini Maleesha | 18020356 |  |

# 9. Reference

1. iamglyka. 2020. *Schedule Feasibility*. [online] Available at: <https://iamglyka.wordpress.com/2013/02/06/schedule-feasibility/> [Accessed 4 May 2020].
2. Web.cse.ohio-state.edu. 2020. *Requirements Document Example*. [online] Available at: <http://web.cse.ohio-state.edu/~bair.41/616/Project/Example\_Document/Req\_Doc\_Example.html> [Accessed 4 May 2020].
3. Programsformca.com. 2020. *UML Diagrams For Tour Management*. [online] Available at: <http://www.programsformca.com/2012/03/uml-diagrams-for-tour-management.html> [Accessed 4 May 2020].
4. Michael Bierman's Randomness. 2020. *Project Feasibility: Should We Include Ethical Feasibility?*. [online] Available at: <https://michaelbierman.wordpress.com/2010/09/15/project-feasibility-should-we-include-ethical-feasibility/> [Accessed 4 May 2020].
5. Simplilearn.com. 2020. *What Is Feasibility Study And Its Importance In Project Management*. [online] Available at: <https://www.simplilearn.com/feasibility-study-article> [Accessed 4 May 2020].
6. Usmani, F., 2020. *Assumptions And Constraints In Project Management | PM Study Circle*. [online] PM Study Circle. Available at: <https://pmstudycircle.com/2012/10/assumptions-and-constraints-in-project-management/> [Accessed 8 May 2020].
7. Masterofproject.com. 2020. *Master | Project Constraints Definition | 7 Common Project Constraints*. [online] Available at: <https://masterofproject.com/blog/7390/project-constraints-definition-7-common-project-constraints> [Accessed 8 May 2020].
8. Stackify. 2020. *What Is Web Application Architecture? Best Practices, Tutorials*. [online] Available at: <https://stackify.com/web-application-architecture/> [Accessed 5 July 2020].
9. Spaceotechnologies. 2020. *Web Application Architecture: Read Everything About It In 15 Minutes*. [online] Available at: <https://www.spaceotechnologies.com/web-application-architecture/> [Accessed 5 July 2020].