INF 271 Deliverable 03/04.Prototype and Technical Specification

Group Number:06

Members Details from left to right:

Name	Surname	Student Number	Email Address
Phumzile	Phakathi	215757555	u21575755@tuks.co.za
Busisiwe	Khosa	2047324	u2047324@tuks.co.za
Rethabile	Kgotlane	21601144	u21601144@tuks.co.za
Basetsana	Sekhoto	21608611	u21608611@tuks.co.za
Maria	Sadiki	22651609	u22651609@tuks.co.za

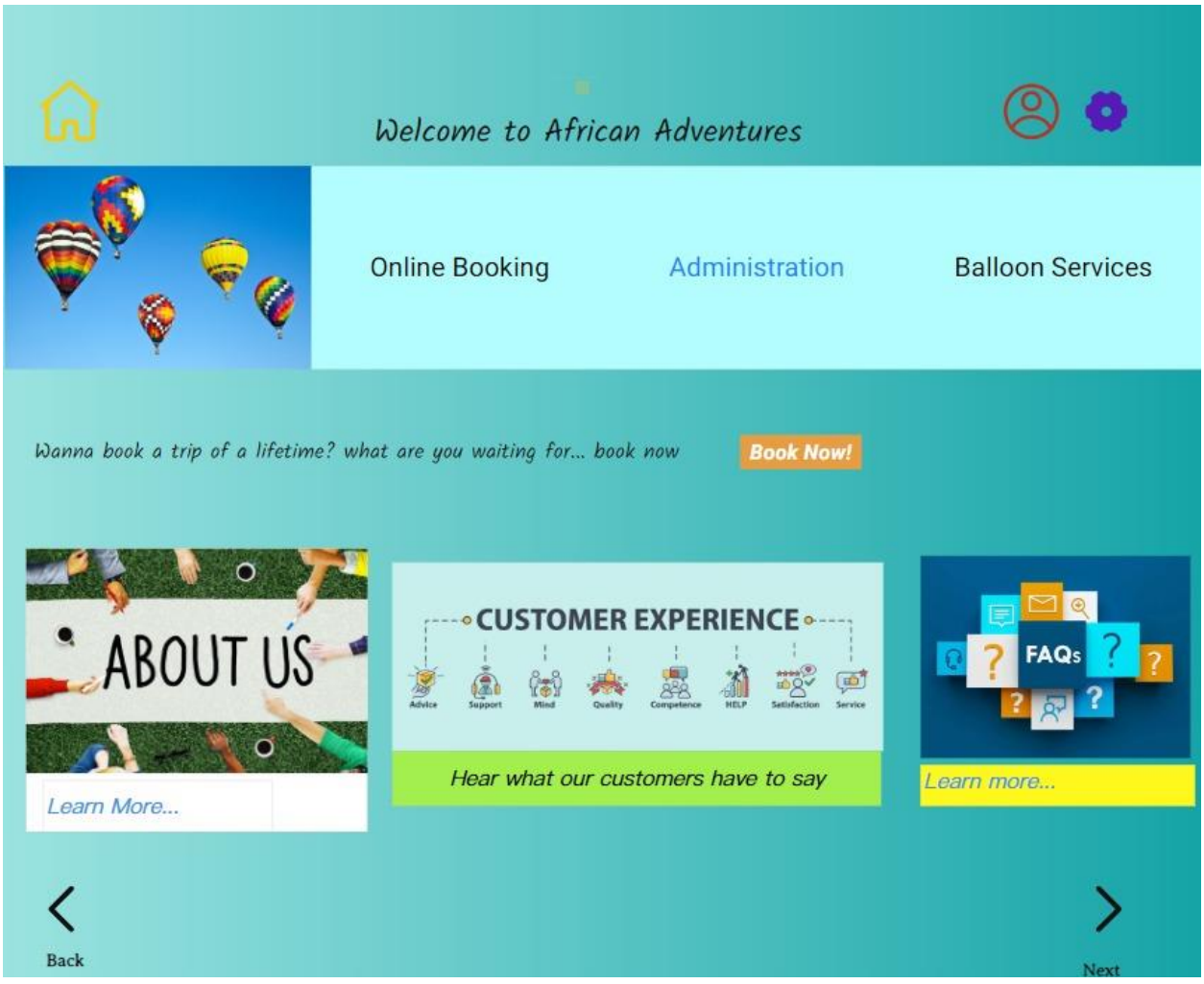
1.Introduction

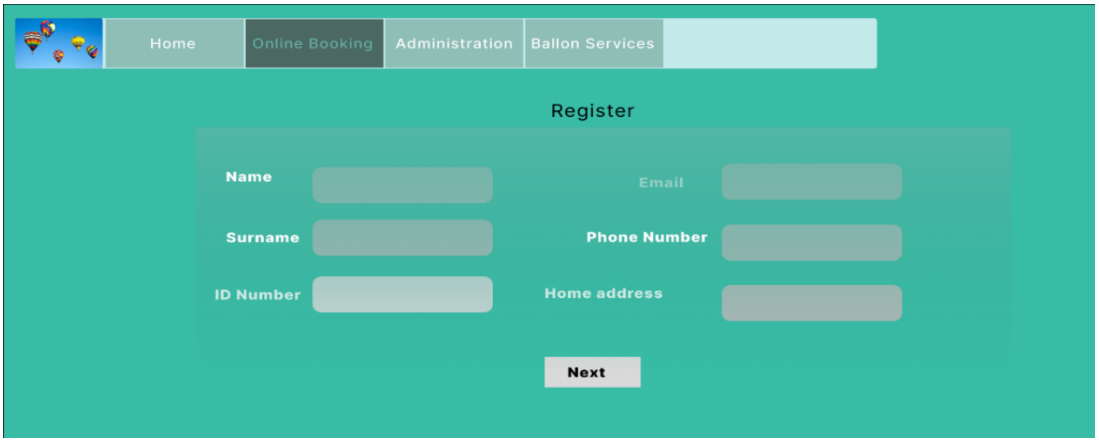
Welcome to the third and last leg of our INF 271, Group Project. Over the next pages, our team will focus on developing the prototypes and technical specifications for the African Adventures Information System. This document will be a thirteen-part process. We will begin with a set of prototype screens, which will help bring the system to life. This is followed by a series of input descriptions that help us to further understand the individual components of each screen. After which we will give the developer a design guideline by means of Design principles. From there we will focus on the technical aspect of Systems development. Which will include a use case diagram, technical narratives, and entity relationship model and test specifications for our demo project. Concluding our project by conducting a validation of each use case and the methods thereof.

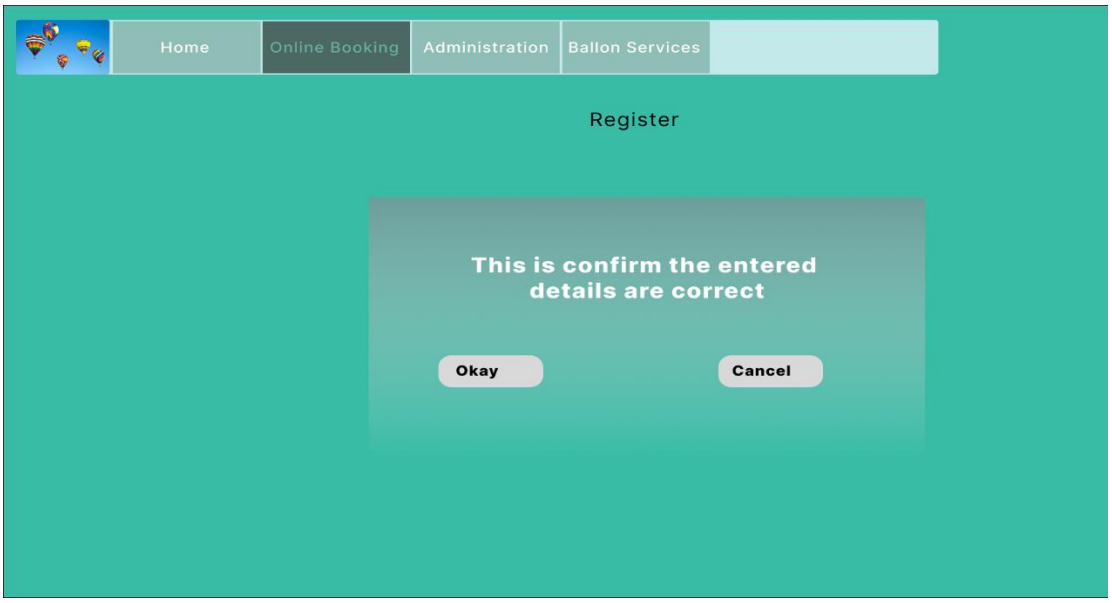
B. Prototype Content

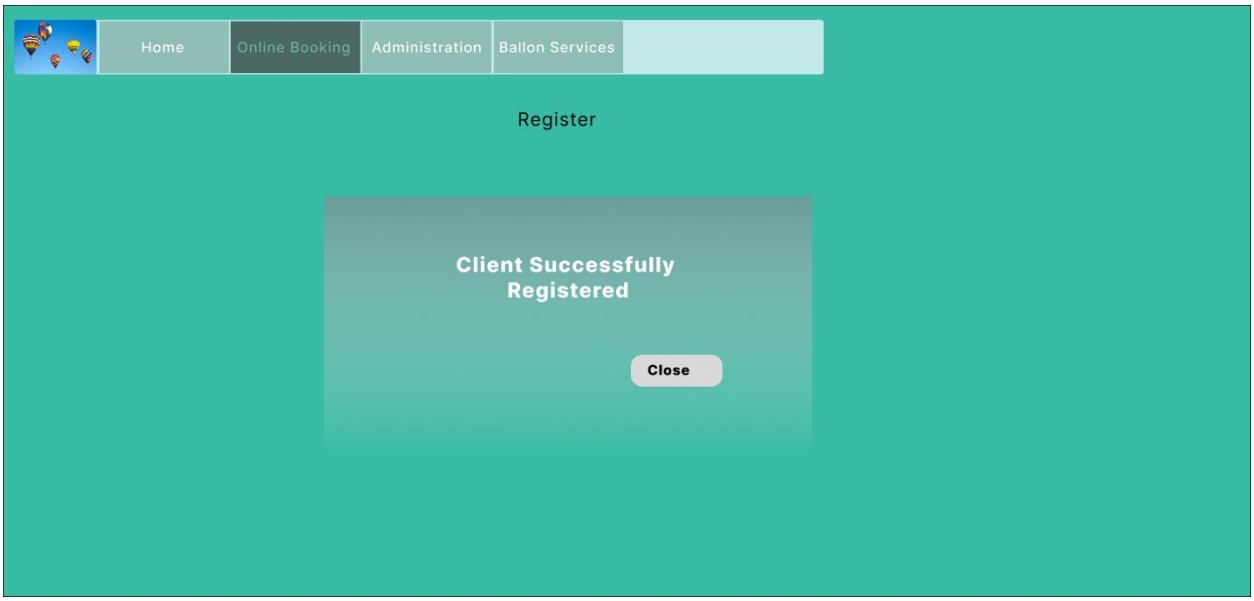
1.Screens Webpages menus and Reports

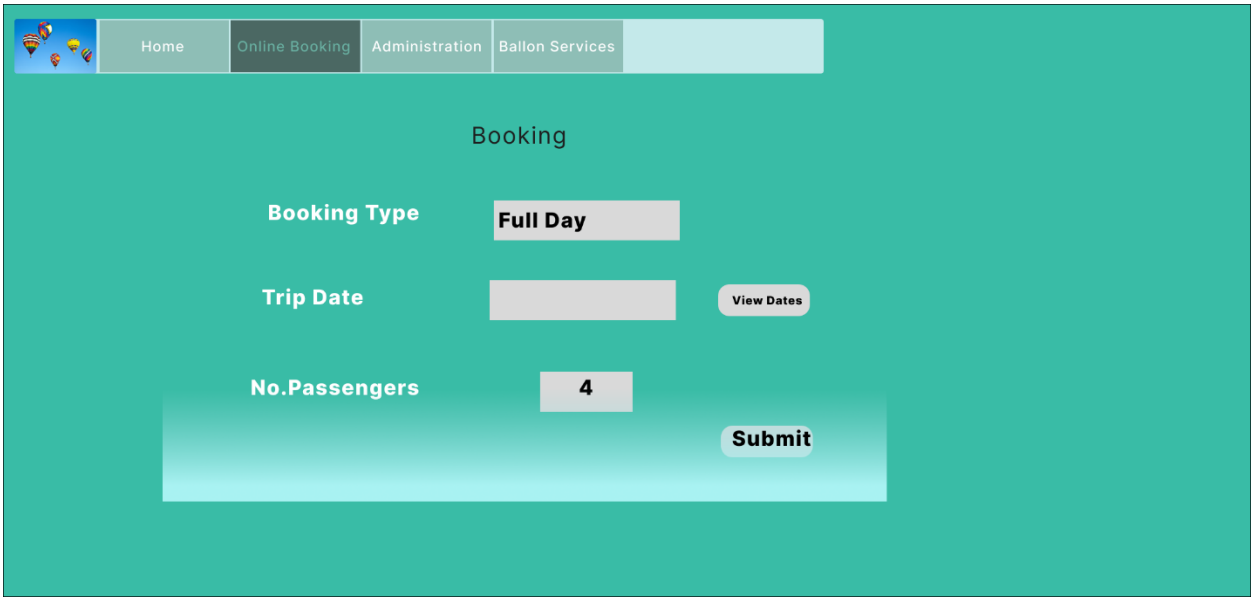
The next few pages will include our Prototypes for our proposed system for African Adventures. Prototypes are a vital component of systems development as they allow users and analysts to visualize the system according to their requirements.

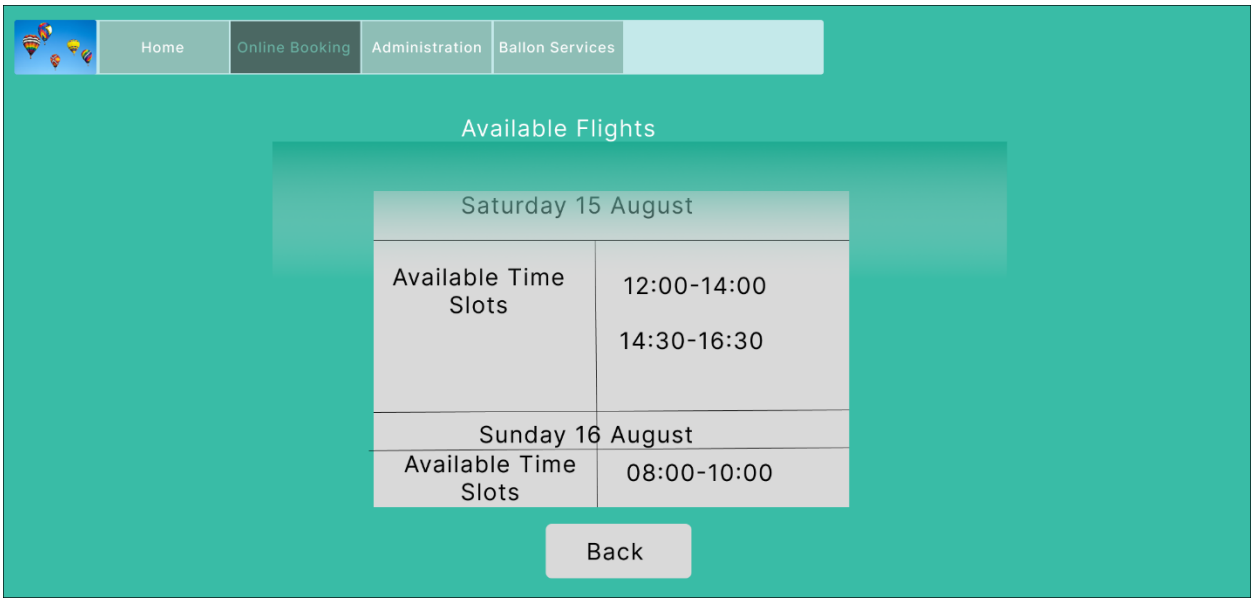


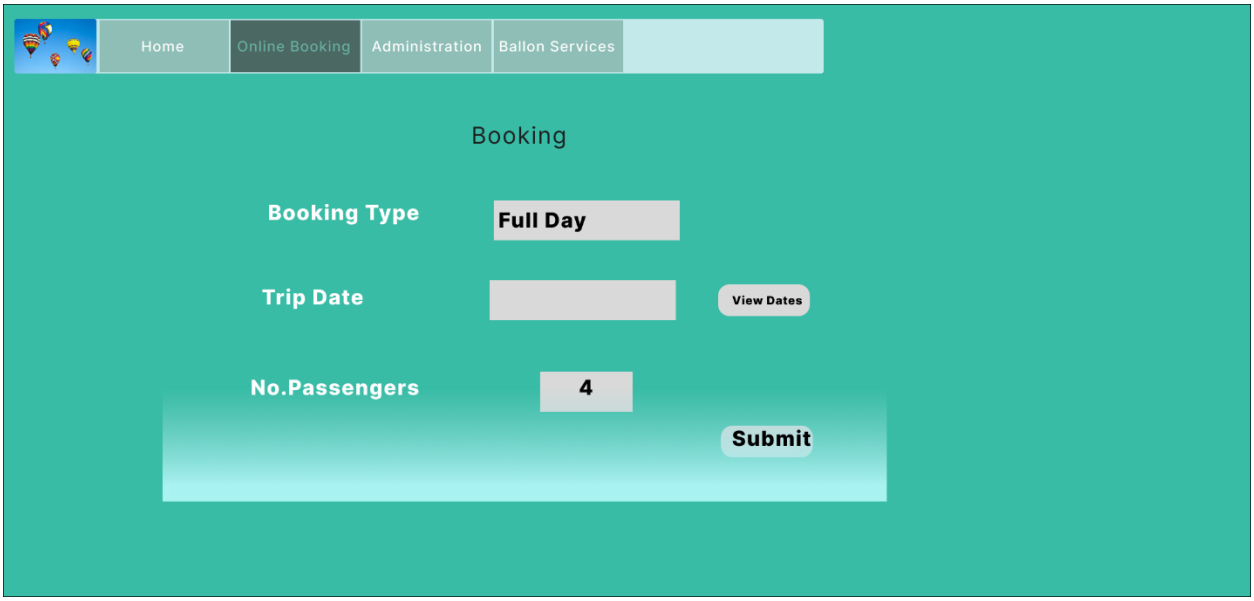

Home
Online Booking
Administration
Ballon Services
Register
Name
Email
Surname
Phone Number
ID Number
Home address
Next

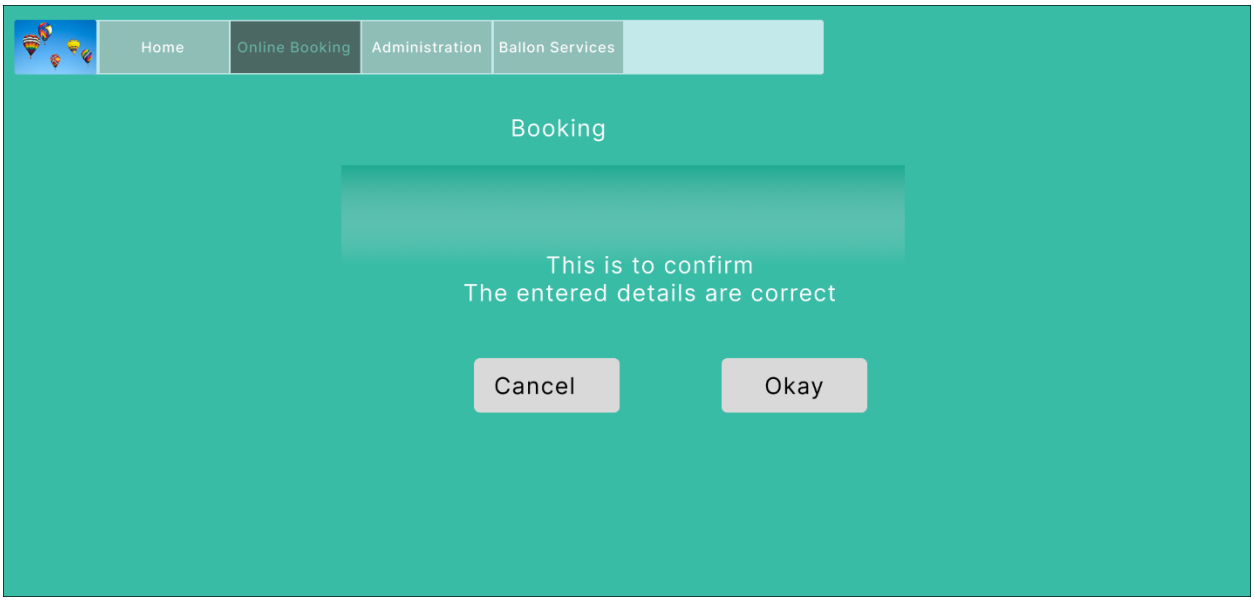

Home
Online Booking
Administration
Ballon Services
Register
This is confirm the entered details are correct
Okay
Cancel

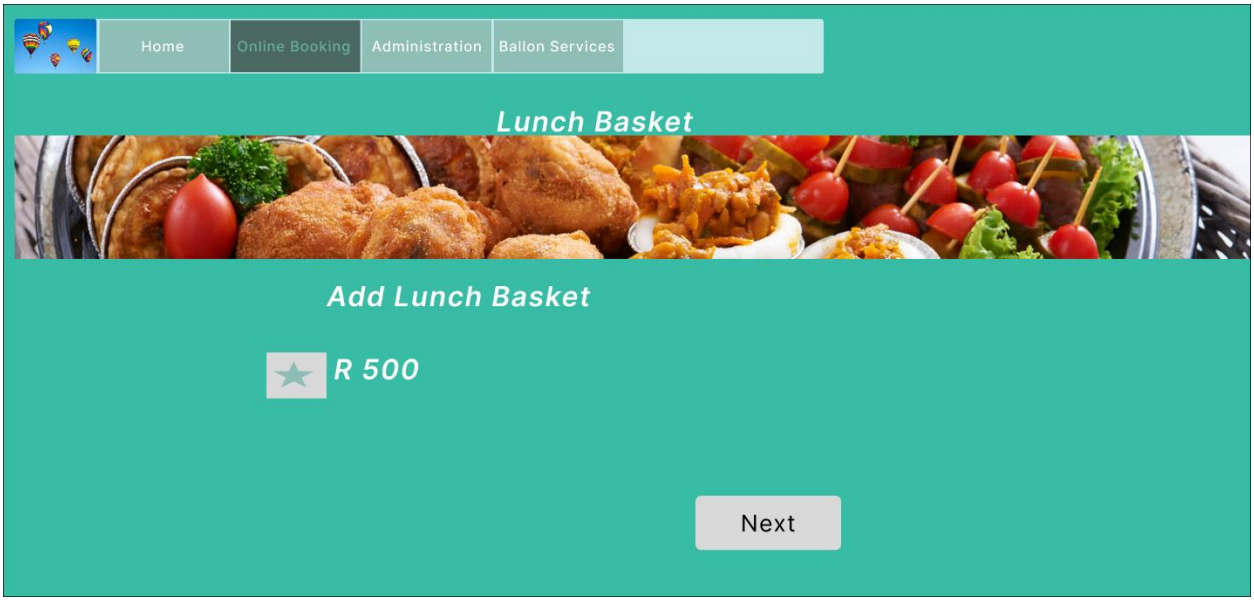

Home
Online Booking
Administration
Ballon Services
Register
Client Successfully Registered
Close

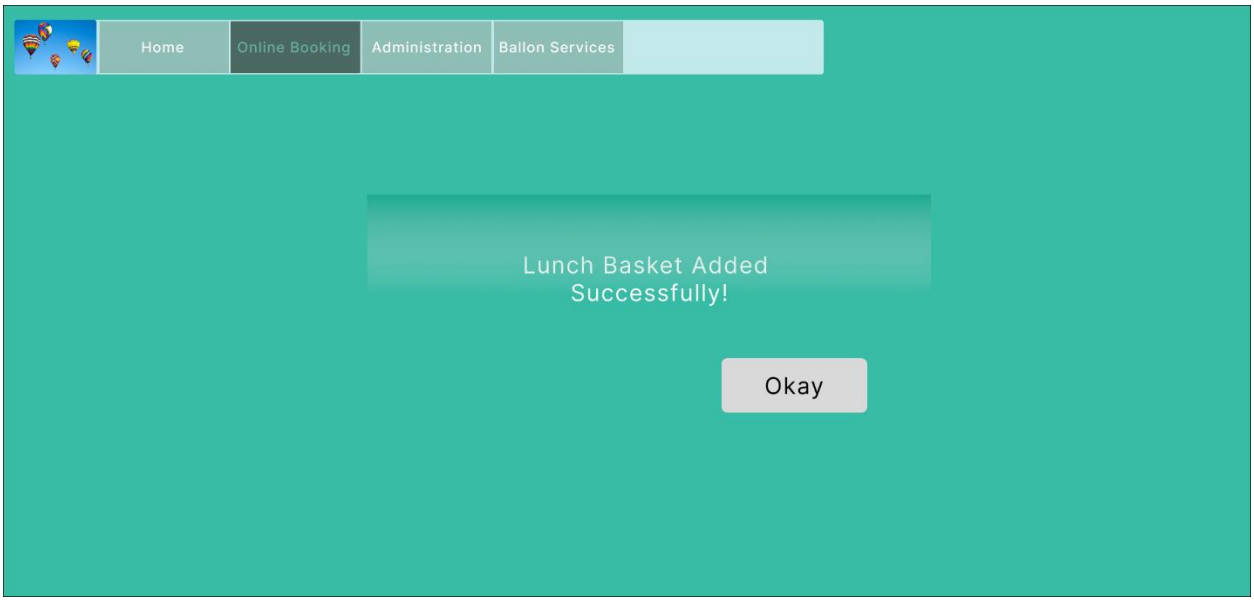

Home
Online Booking
Administration
Ballon Services
Booking
Booking Type
Full Day
Trip Date
View Dates
No.Passengers
4
Submit

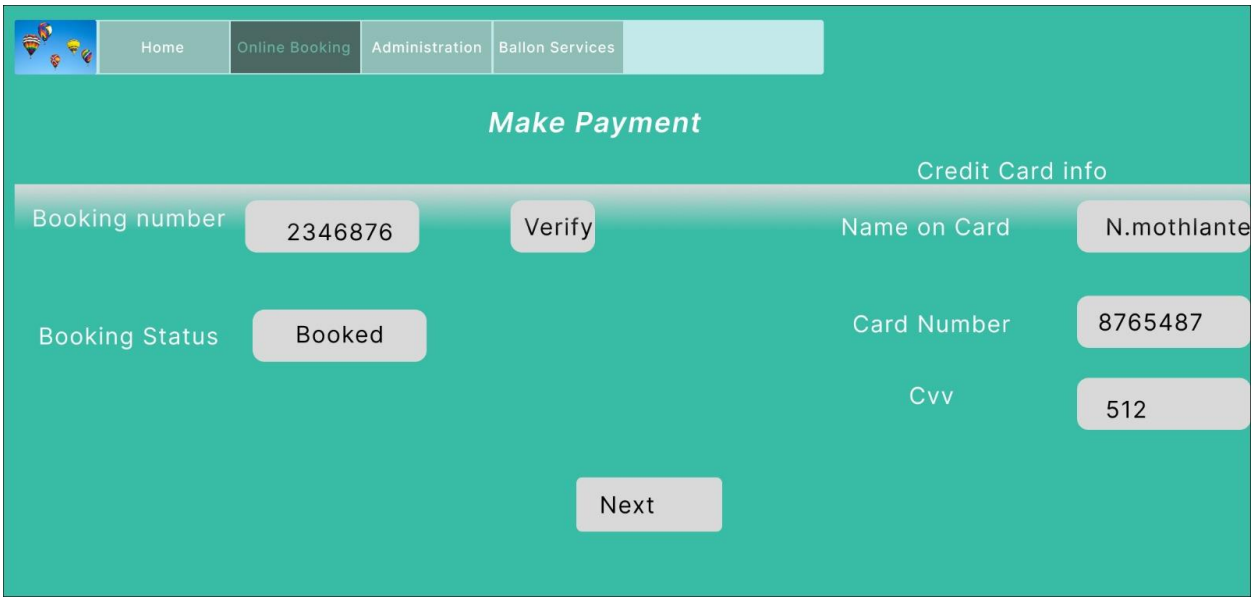

Home
Online Booking
Administration
Ballon Services
Available Flights
Saturday 15 August
Available Time Slots
12:00-14:00
14:30-16:30
Sunday 16 August
Available Time Slots
08:00-10:00
Back

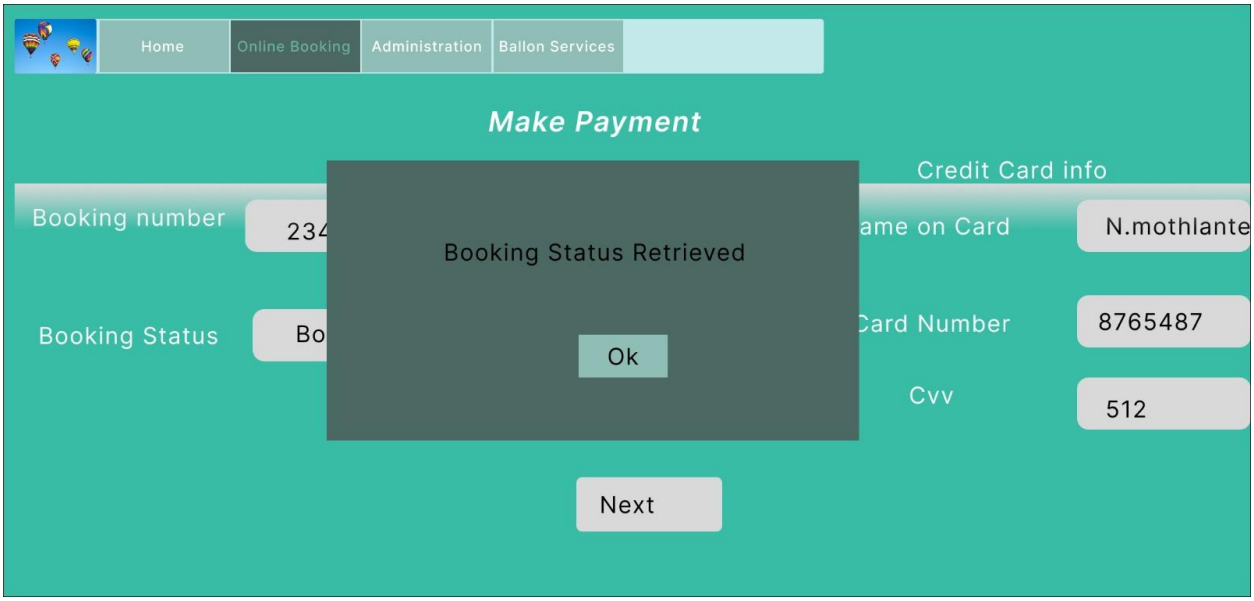

Home
Online Booking
Administration
Ballon Services
Booking
Booking Type
Full Day
Trip Date
View Dates
No.Passengers
4
Submit

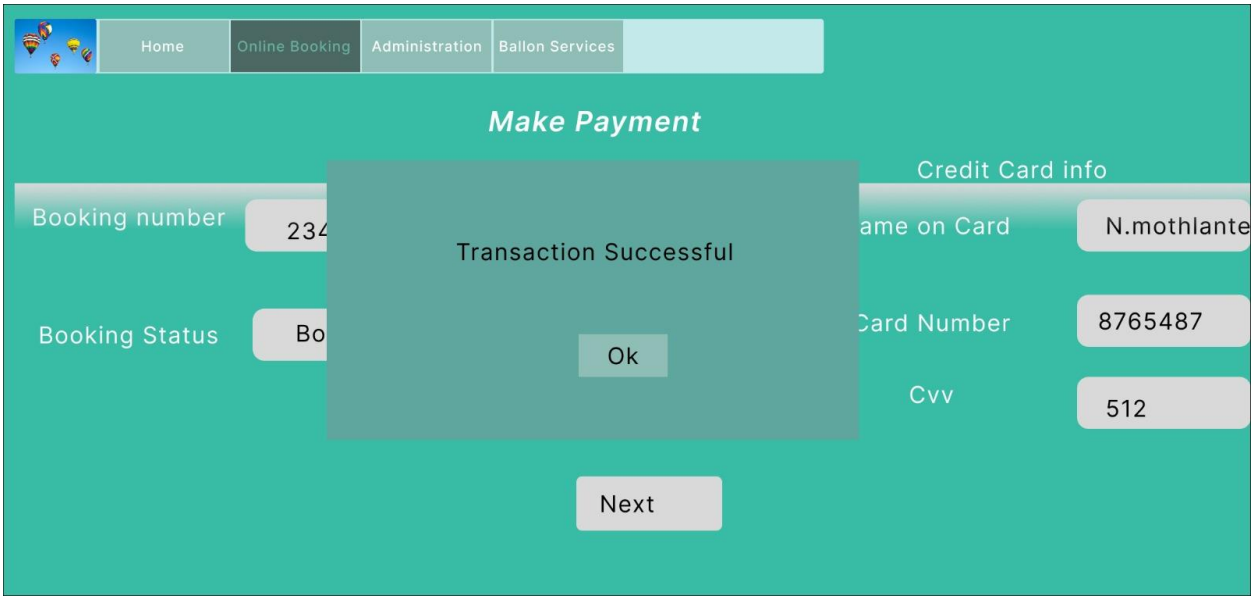

Home
Online Booking
Administration
Ballon Services
Booking
This is to confirm
The entered details are correct
Cancel
Okay

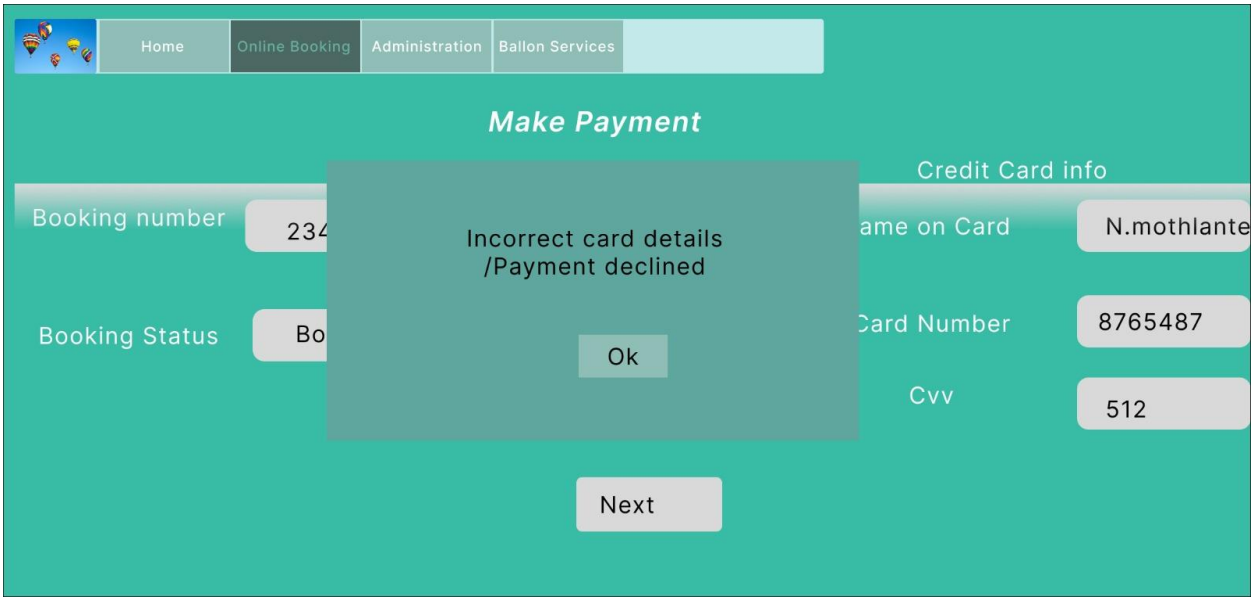

Home
Online Booking
Administration
Ballon Services
Lunch Basket
Add Lunch Basket
R 500
Next

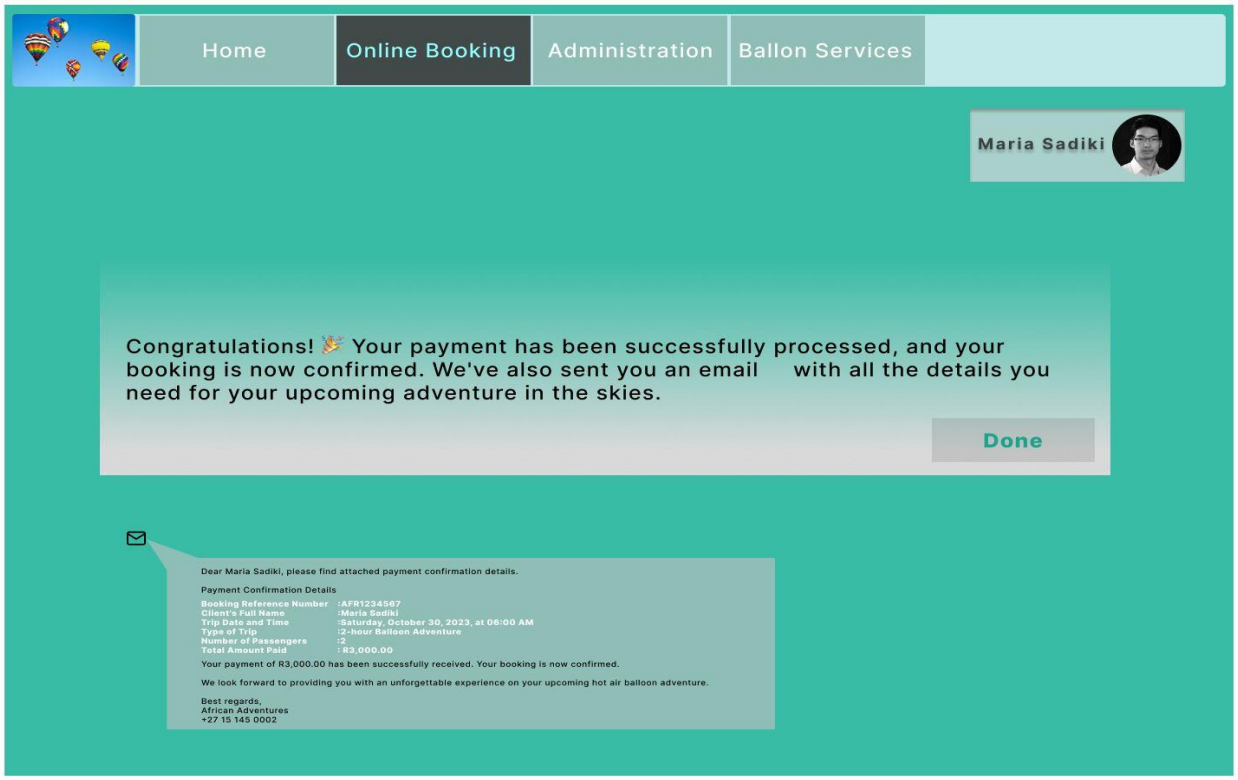

Home
Online Booking
Administration
Ballon Services
Lunch Basket Added Successfully!
Okay

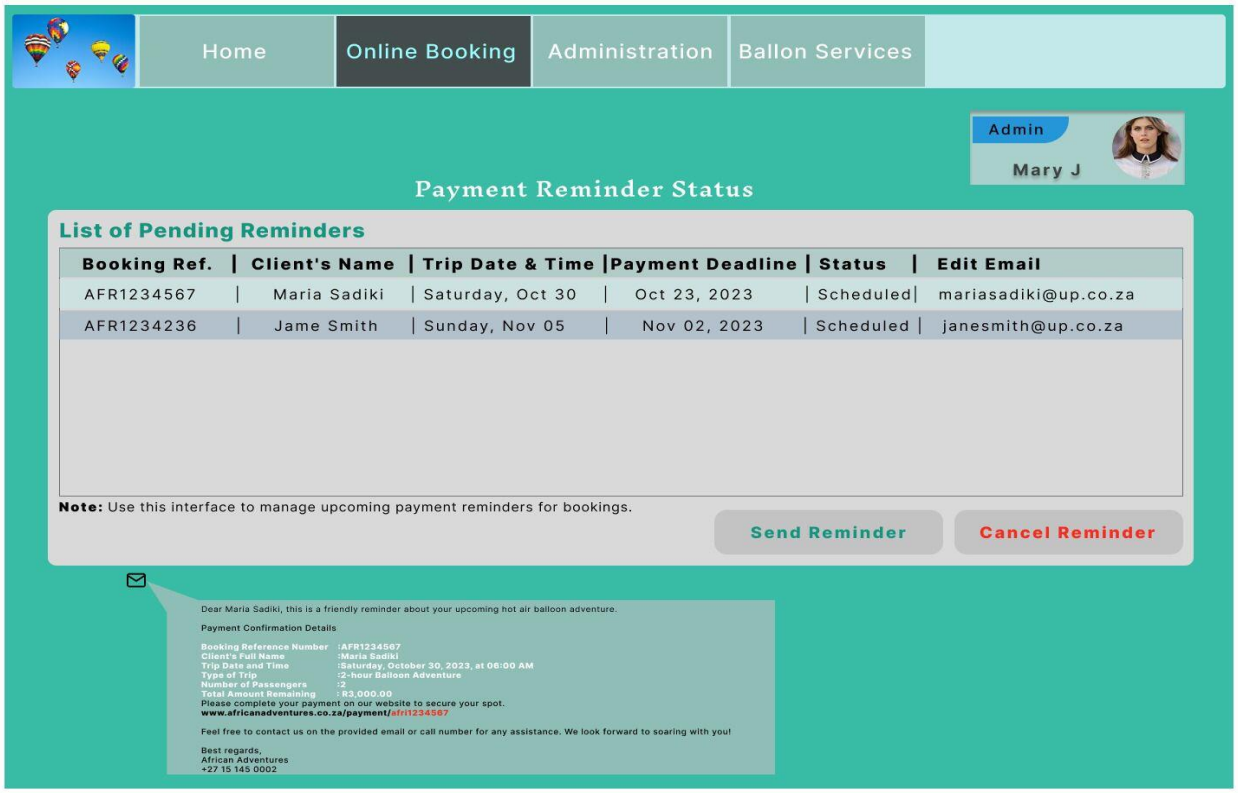

Home
Online Booking
Administration
Ballon Services
Make Payment
Credit Card info
Booking number
2346876
Verify
Name on Card
N.mothlante
Booking Status
Booked
Card Number
8765487
Cvv
512
Next

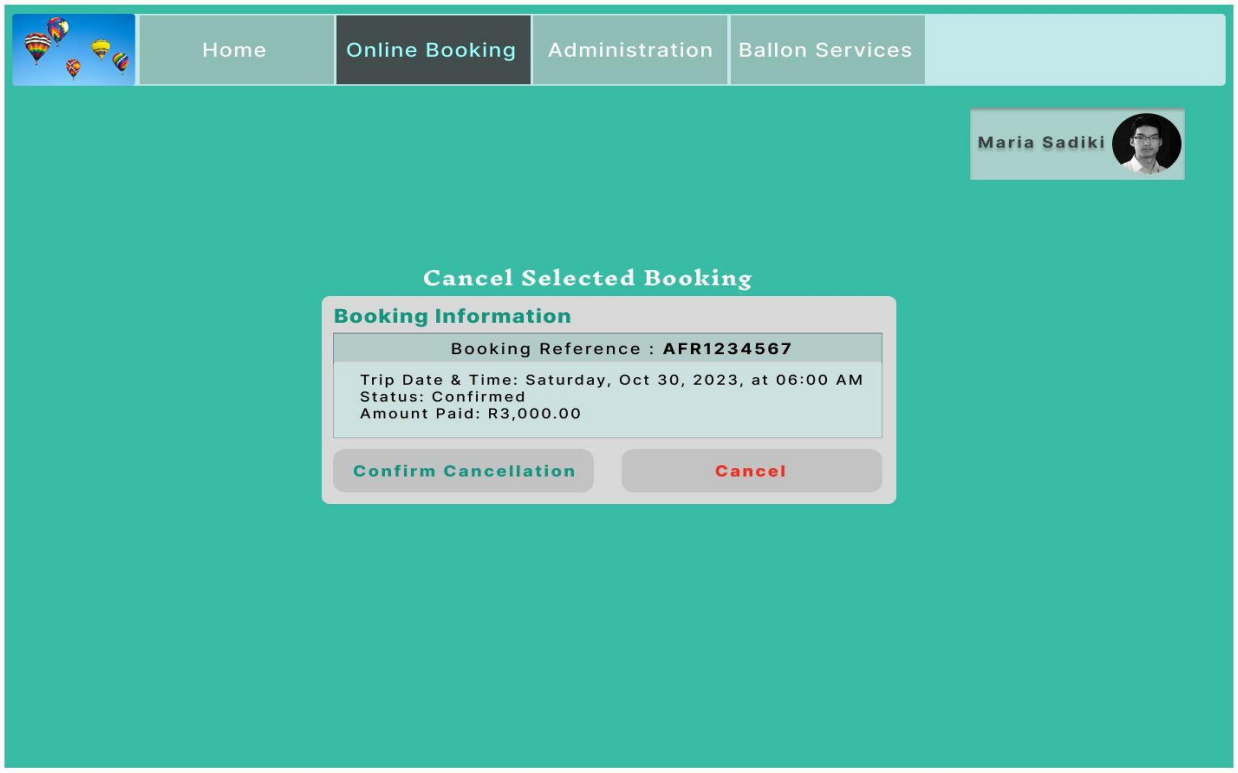

Home
Online Booking
Administration
Ballon Services
Make Payment
Booking number
234
Booking Status
Bo
Booking Status Retrieved
Ok
Credit Card info
ame on Card
N.mothlante
Card Number
8765487
Cvv
512
Next

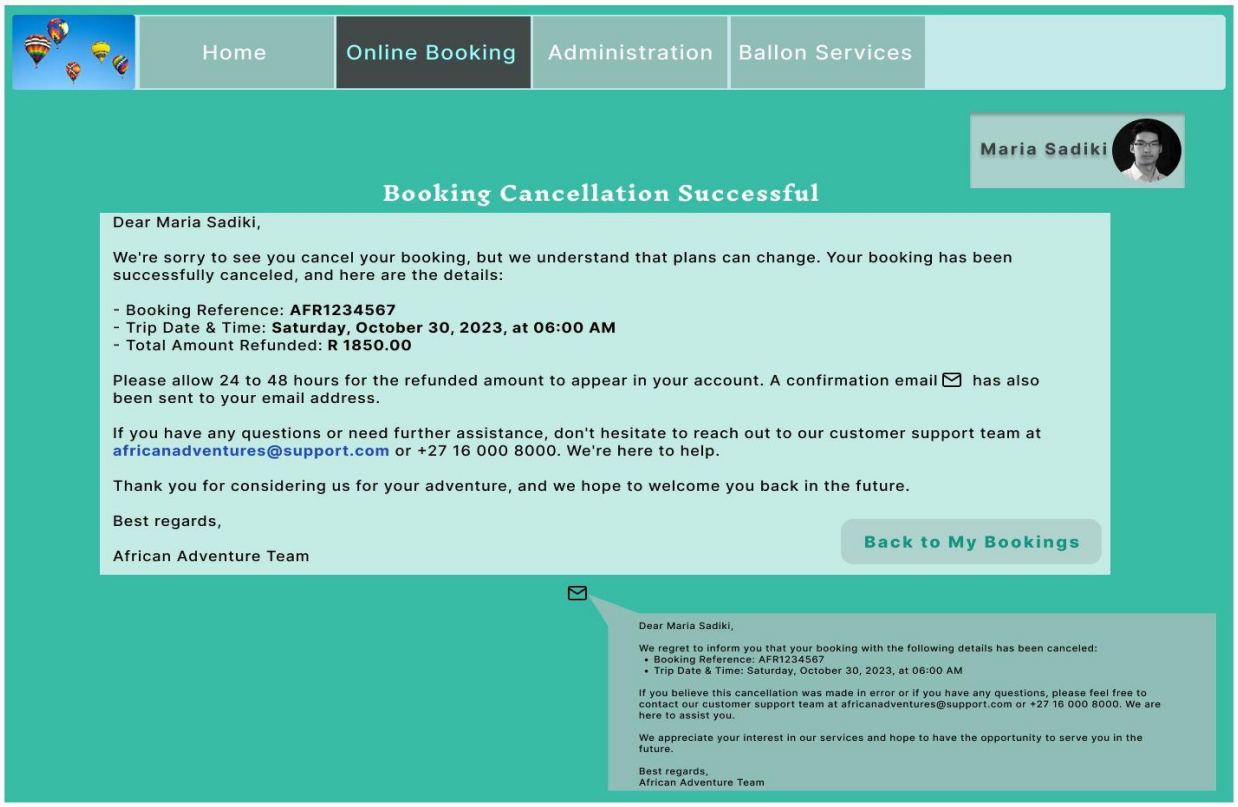

Home
Online Booking
Administration
Ballon Services
Make Payment
Booking number
234
Booking Status
Bo
Transaction Successful
Ok
Credit Card info
ame on Card
N.mothlante
Card Number
8765487
Cvv
512
Next

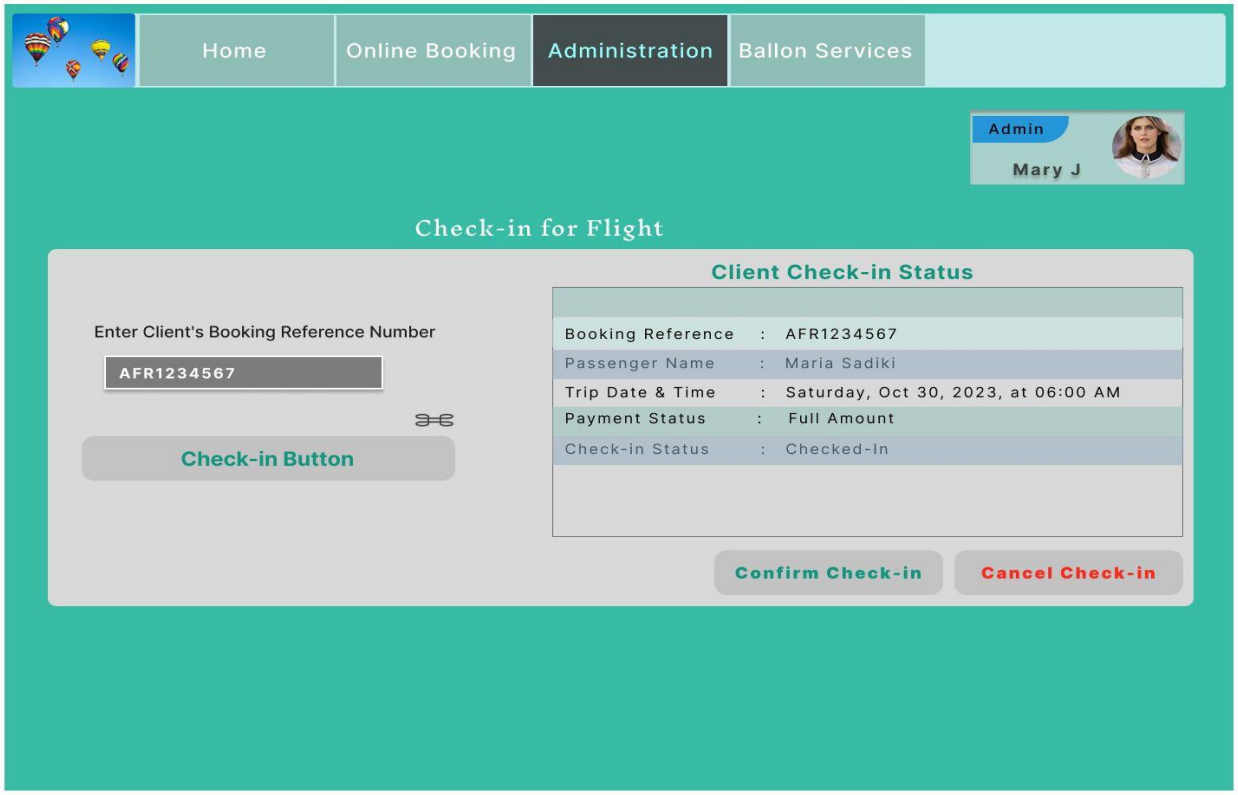

Home
Online Booking
Administration
Ballon Services
Make Payment
Booking number
234
Booking Status
Bo
Incorrect card details
/Payment declined
Ok
Credit Card info
ame on Card
N.mothlante
Card Number
8765487
Cvv
512
Next

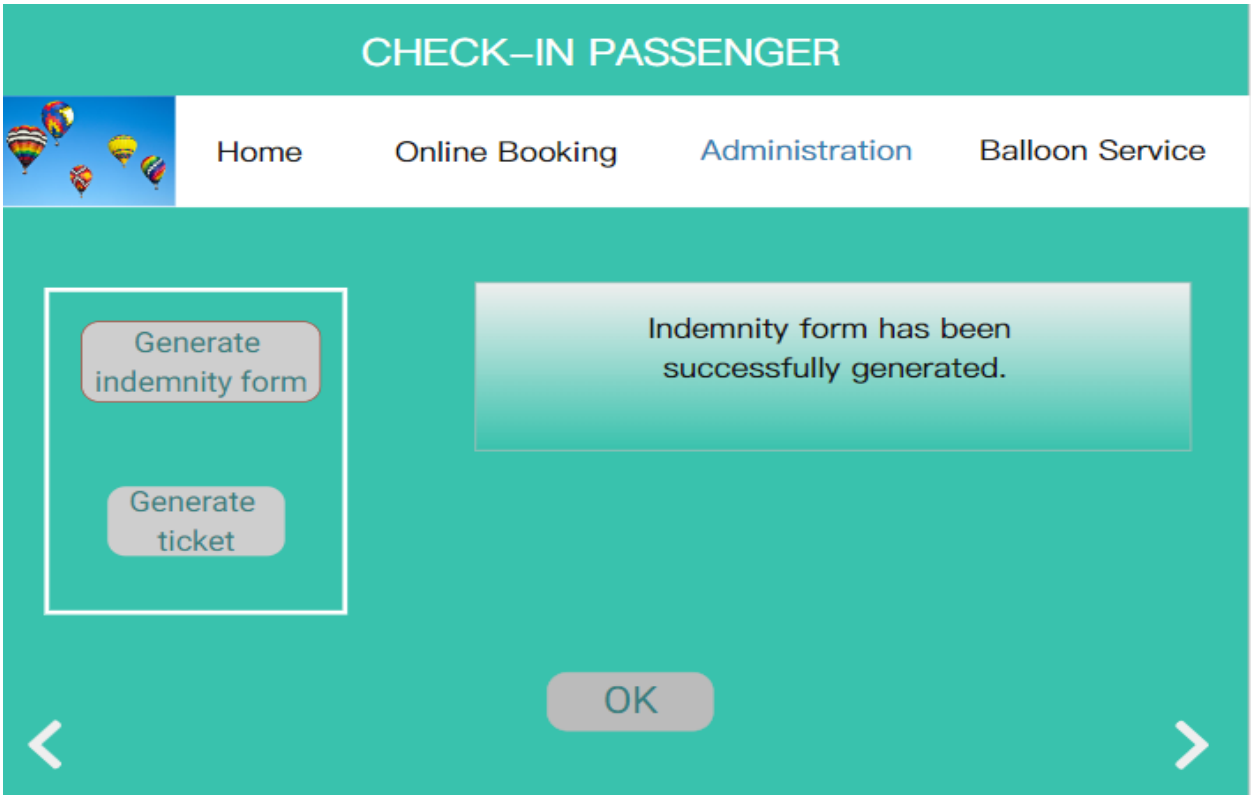

Home
Online Booking
Administration
Ballon Services
Maria Sadiki
Congratulations! 🎉 Your payment has been successfully processed, and your booking is now confirmed. We've also sent you an email with all the details you need for your upcoming adventure in the skies.
Done
Dear Maria Sadiki, please find attached payment confirmation details.
Payment Confirmation Details
Booking Reference Number :AFR1234567
Client's Full Name :Maria Sadiki
Trip Date and Time :Saturday, October 30, 2023, at 06:00 AM
Type of Trip :2-hour Balloon Adventure
Number of Passengers :2
Total Amount Paid : R3,000.00
Your payment of R3,000.00 has been successfully received. Your booking is now confirmed.
We look forward to providing you with an unforgettable experience on your upcoming hot air balloon adventure.
Best regards,
African Adventures
+27 15 145 0002

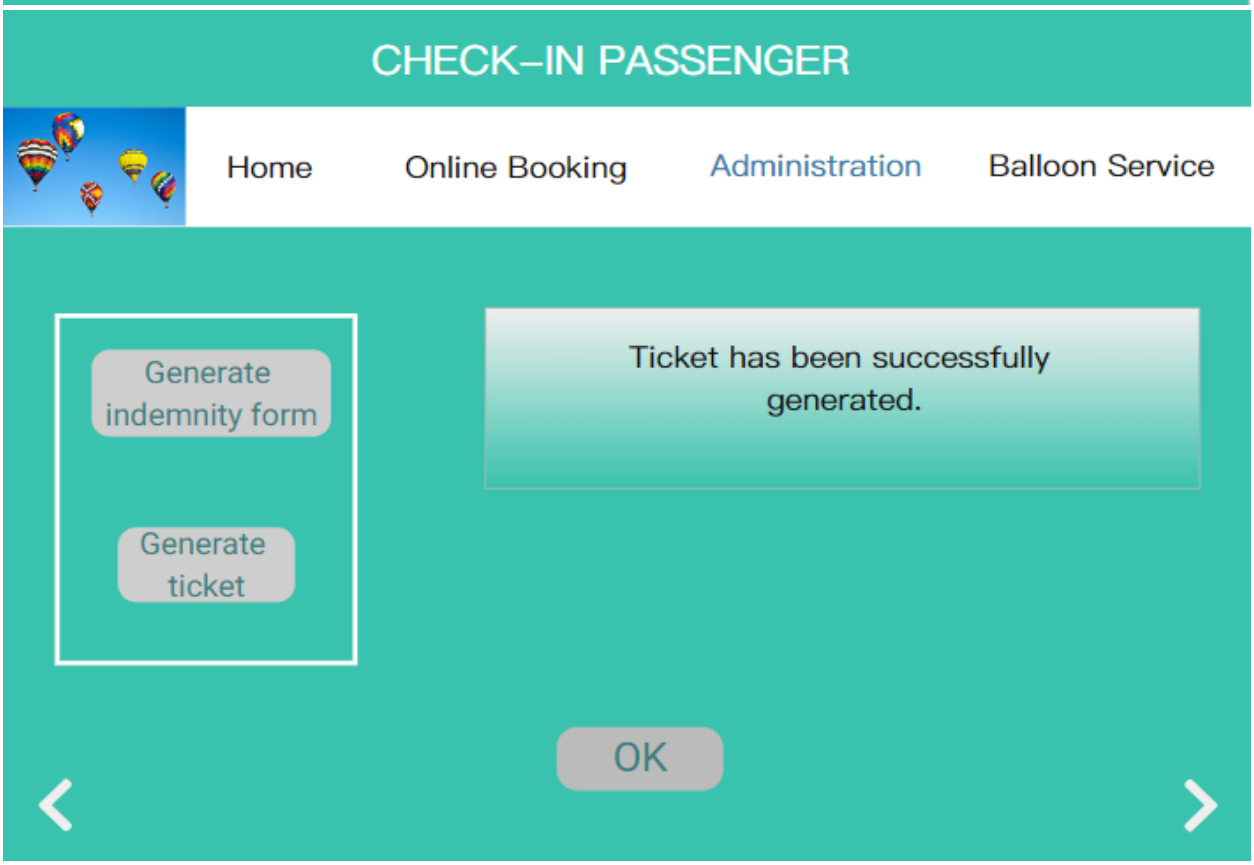

Home
Online Booking
Administration
Ballon Services
Admin
Mary J
Payment Reminder Status
List of Pending Reminders
Booking Ref. | Client's Name | Trip Date & Time | Payment Deadline | Status | Edit Email
AFR1234567 | Maria Sadiki | Saturday, Oct 30 | Oct 23, 2023 | Scheduled | mariasadiki@up.co.za
AFR1234236 | Jame Smith | Sunday, Nov 05 | Nov 02, 2023 | Scheduled | janesmith@up.co.za
Note: Use this interface to manage upcoming payment reminders for bookings.
Send Reminder
Cancel Reminder
Dear Maria Sadiki, this is a friendly reminder about your upcoming hot air balloon adventure.
Payment Confirmation Details
Booking Reference Number :AFR1234567
Client's Full Name :Maria Sadiki
Trip Date and Time :Saturday, October 30, 2023, at 06:00 AM
Type of Trip :2-hour Balloon Adventure
Number of Passengers :2
Total Amount Remaining : R3,000.00
Please complete your payment on our website to secure your spot.
www.africanadventures.co.za/payment/afri1234567
Feel free to contact us on the provided email or call number for any assistance. We look forward to soaring with you!
Best regards,
African Adventures
+27 15 145 0002

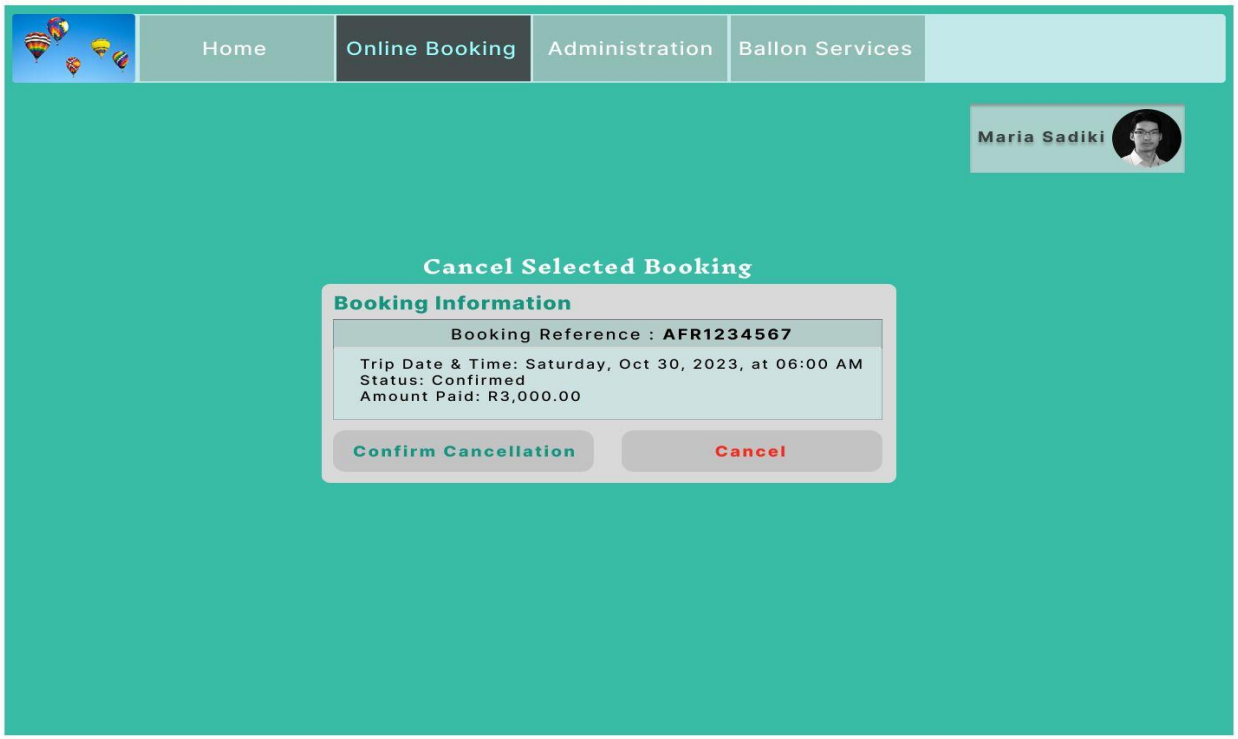

Home
Online Booking
Administration
Ballon Services
Maria Sadiki
Cancel Selected Booking
Booking Information
Booking Reference : AFR1234567
Trip Date & Time: Saturday, Oct 30, 2023, at 06:00 AM
Status: Confirmed
Amount Paid: R3,000.00
Confirm Cancellation
Cancel

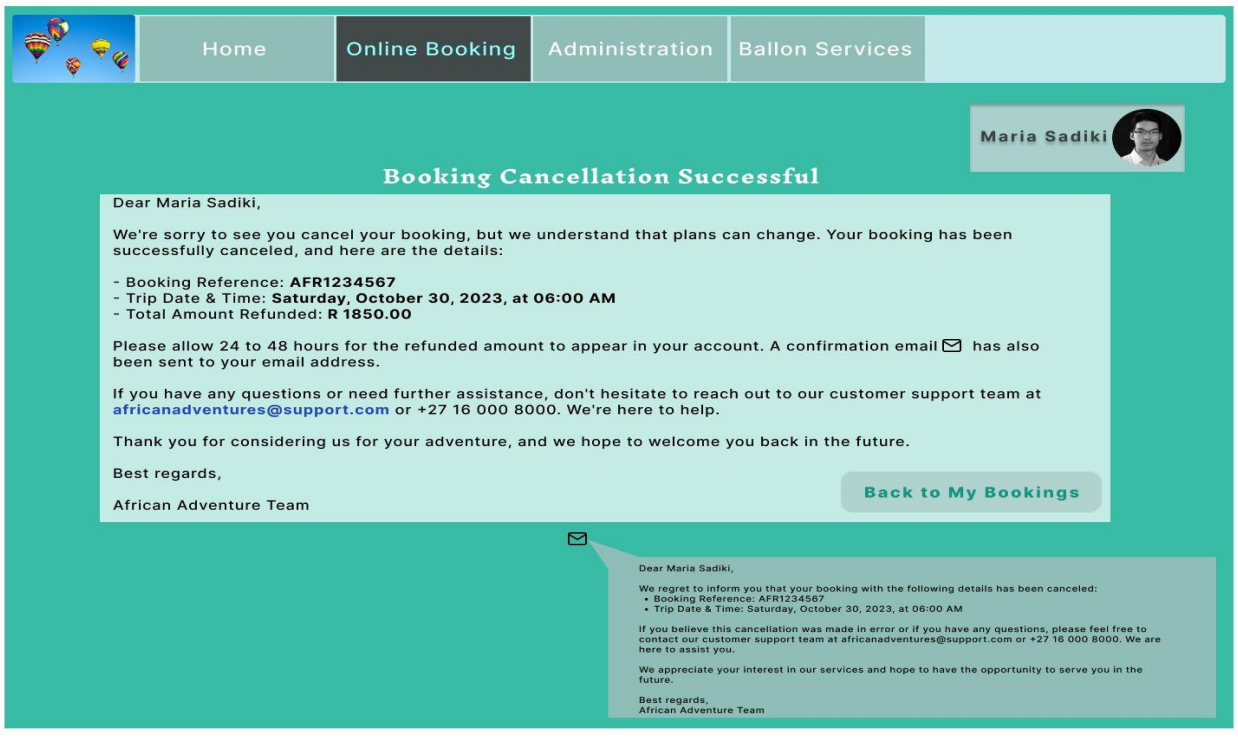

Home
Online Booking
Administration
Ballon Services
Maria Sadiki
Booking Cancellation Successful
Dear Maria Sadiki,
We're sorry to see you cancel your booking, but we understand that plans can change. Your booking has been successfully canceled, and here are the details:
- Booking Reference: AFR1234567
- Trip Date & Time: Saturday, October 30, 2023, at 06:00 AM
- Total Amount Refunded: R 1850.00
Please allow 24 to 48 hours for the refunded amount to appear in your account. A confirmation email has also been sent to your email address.
If you have any questions or need further assistance, don't hesitate to reach out to our customer support team at africanadventures@support.com or +27 16 000 8000. We're here to help.
Thank you for considering us for your adventure, and we hope to welcome you back in the future.
Best regards,
African Adventure Team
Back to My Bookings
Dear Maria Sadiki,
We regret to inform you that your booking with the following details has been canceled:
• Booking Reference: AFR1234567
• Trip Date & Time: Saturday, October 30, 2023, at 06:00 AM
If you believe this cancellation was made in error or if you have any questions, please feel free to contact our customer support team at africanadventures@support.com or +27 16 000 8000. We are here to assist you.
We appreciate your interest in our services and hope to have the opportunity to serve you in the future.
Best regards,
African Adventure Team

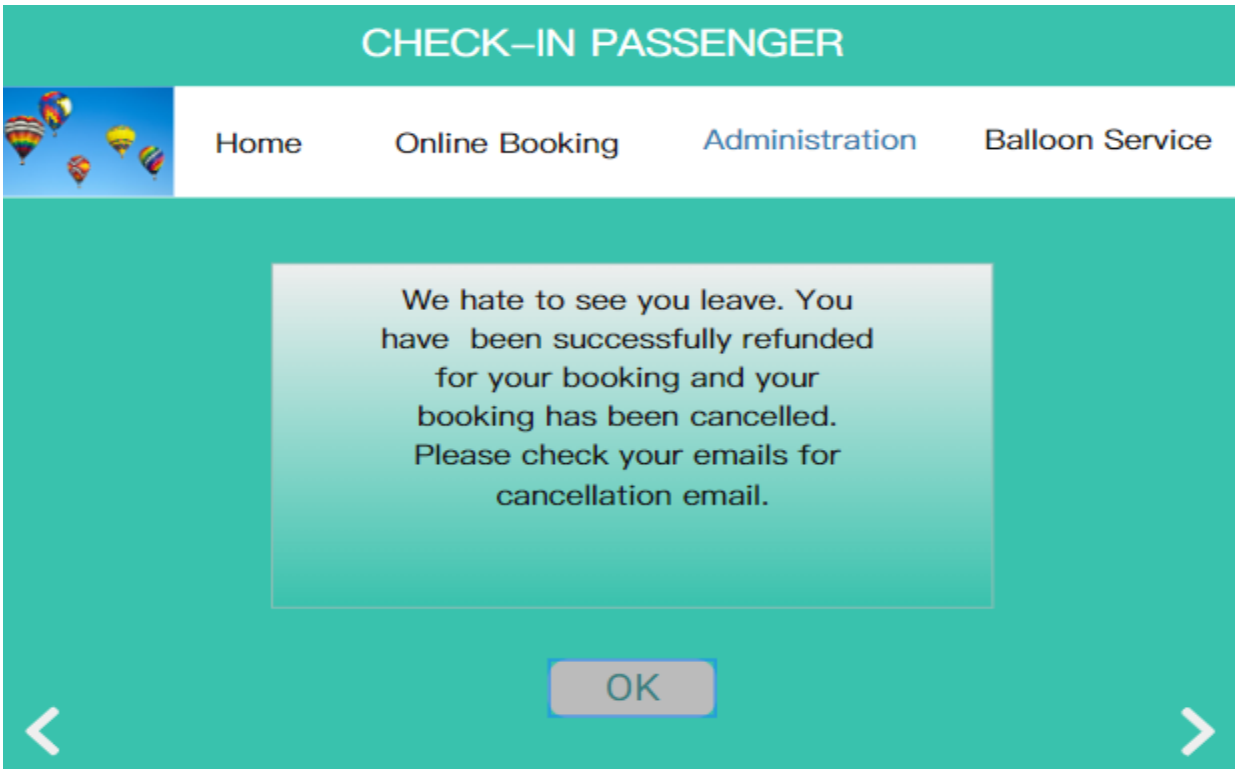

Home
Online Booking
Administration
Ballon Services
Admin
Mary J
Check-in for Flight
Enter Client's Booking Reference Number
AFR1234567
Check-in Button
Client Check-in Status
Booking Reference : AFR1234567
Passenger Name : Maria Sadiki
Trip Date & Time : Saturday, Oct 30, 2023, at 06:00 AM
Payment Status : Full Amount
Check-in Status : Checked-In
Confirm Check-in
Cancel Check-in

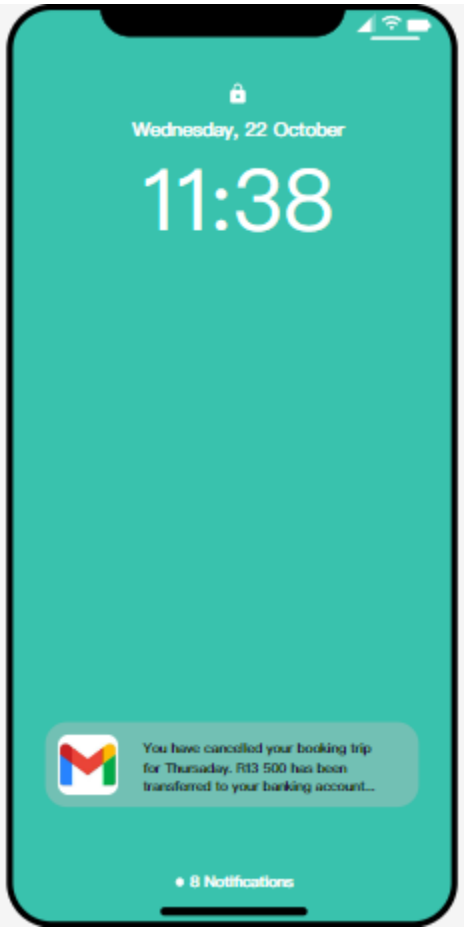

CHECK–IN PASSENGER
Home
Online Booking
Administration
Balloon Service
Generate indemnity form
Generate ticket
Indemnity form has been successfully generated.
OK

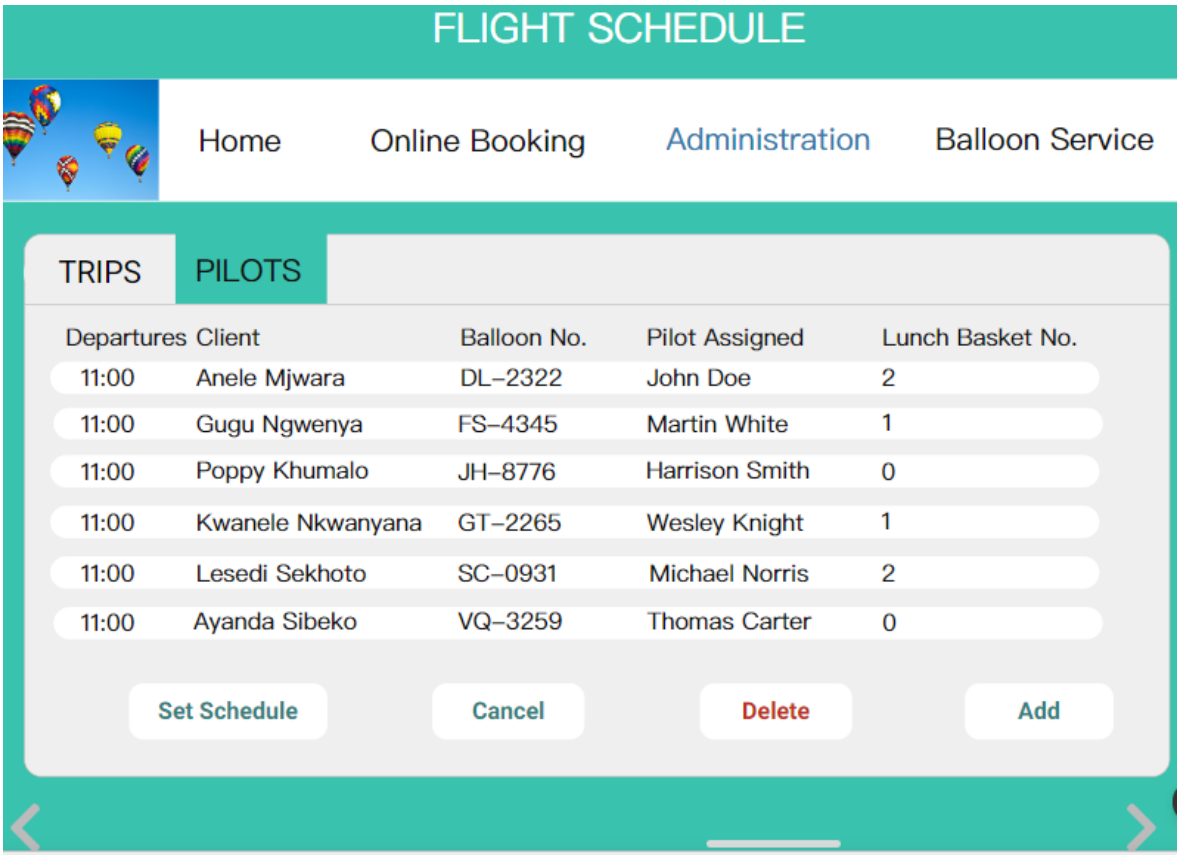

CHECK–IN PASSENGER
Home
Online Booking
Administration
Balloon Service
Generate indemnity form
Generate ticket
Ticket has been successfully generated.
OK

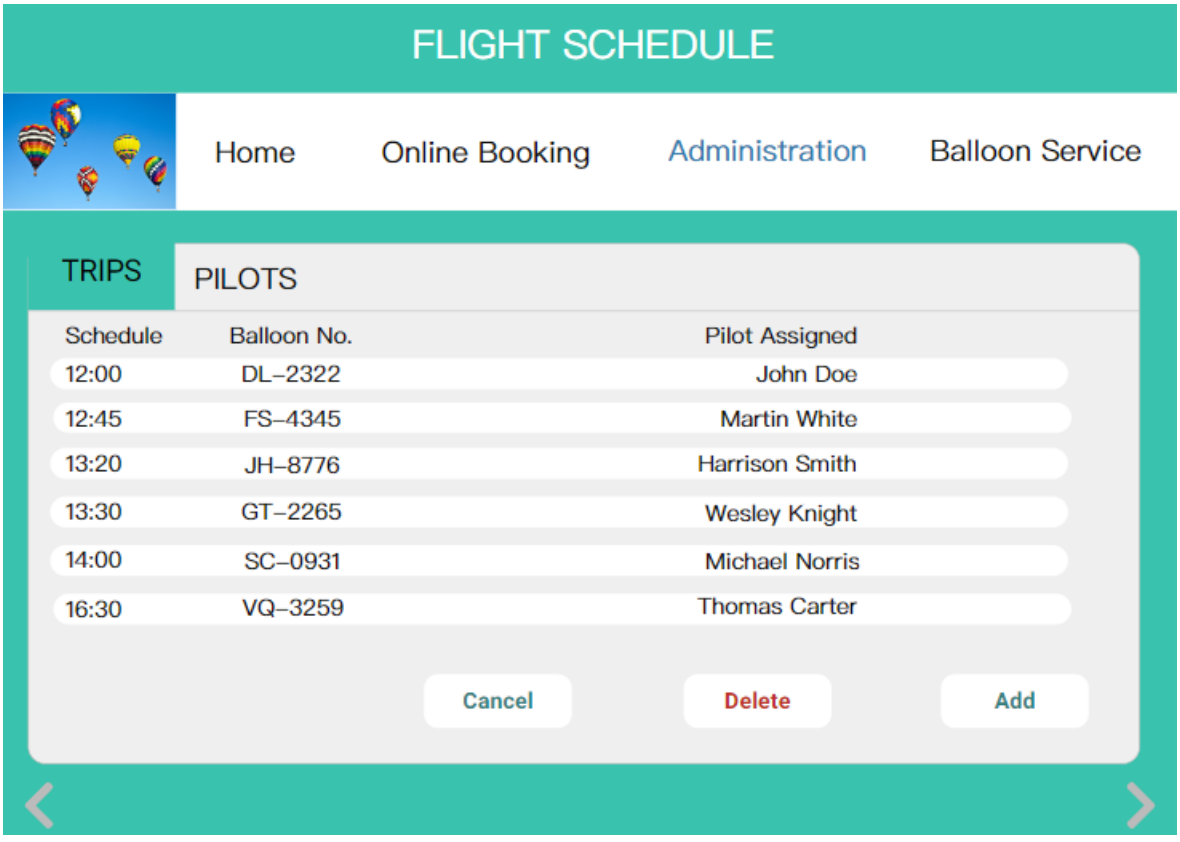

Home
Online Booking
Administration
Ballon Services
Maria Sadiki
Cancel Selected Booking
Booking Information
Booking Reference : AFR1234567
Trip Date & Time: Saturday, Oct 30, 2023, at 06:00 AM
Status: Confirmed
Amount Paid: R3,000.00
Confirm Cancellation
Cancel

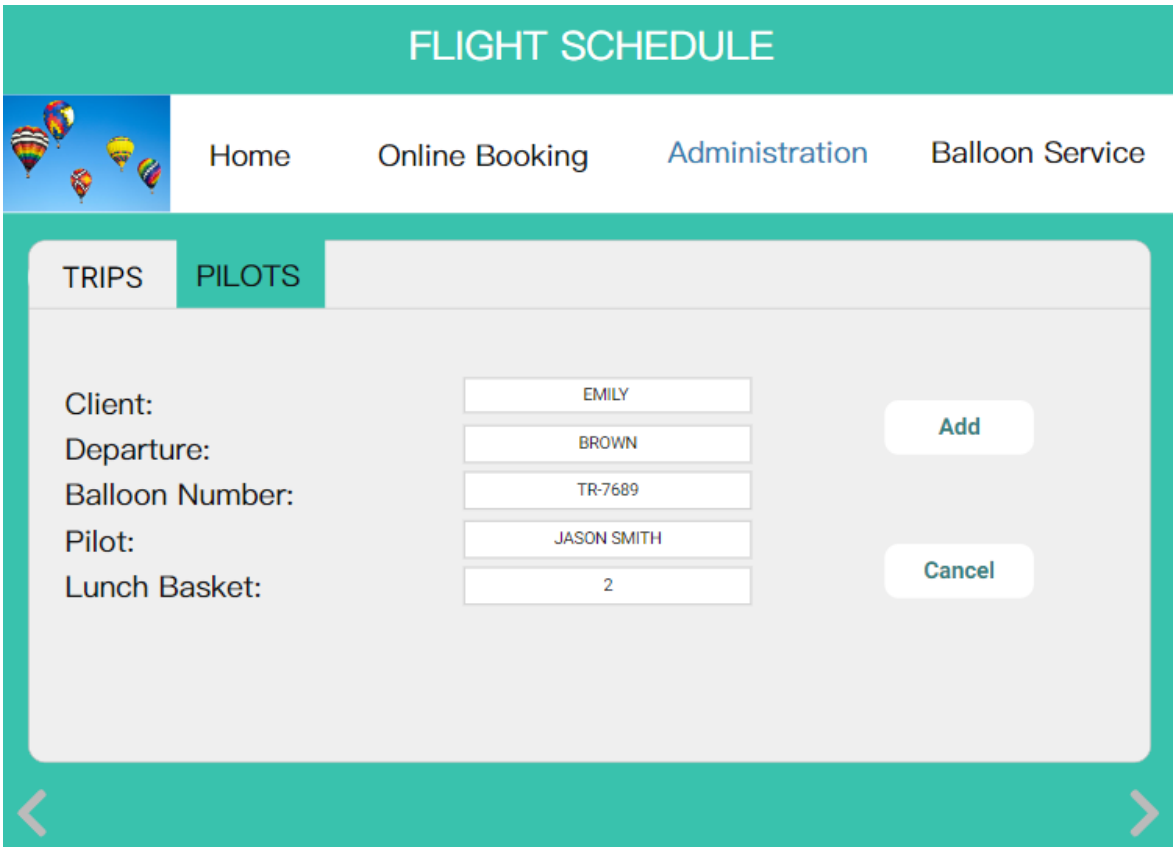

Home
Online Booking
Administration
Ballon Services
Maria Sadiki
Booking Cancellation Successful
Dear Maria Sadiki,
We're sorry to see you cancel your booking, but we understand that plans can change. Your booking has been successfully canceled, and here are the details:
- Booking Reference: AFR1234567
- Trip Date & Time: Saturday, October 30, 2023, at 06:00 AM
- Total Amount Refunded: R 1850.00
Please allow 24 to 48 hours for the refunded amount to appear in your account. A confirmation email has also been sent to your email address.
If you have any questions or need further assistance, don't hesitate to reach out to our customer support team at africanadventures@support.com or +27 16 000 8000. We're here to help.
Thank you for considering us for your adventure, and we hope to welcome you back in the future.
Best regards,
African Adventure Team
Back to My Bookings
Dear Maria Sadiki,
We regret to inform you that your booking with the following details has been canceled:
• Booking Reference: AFR1234567
• Trip Date & Time: Saturday, October 30, 2023, at 06:00 AM
If you believe this cancellation was made in error or if you have any questions, please feel free to contact our customer support team at africanadventures@support.com or +27 16 000 8000. We are here to assist you.
We appreciate your interest in our services and hope to have the opportunity to serve you in the future.
Best regards,
African Adventure Team

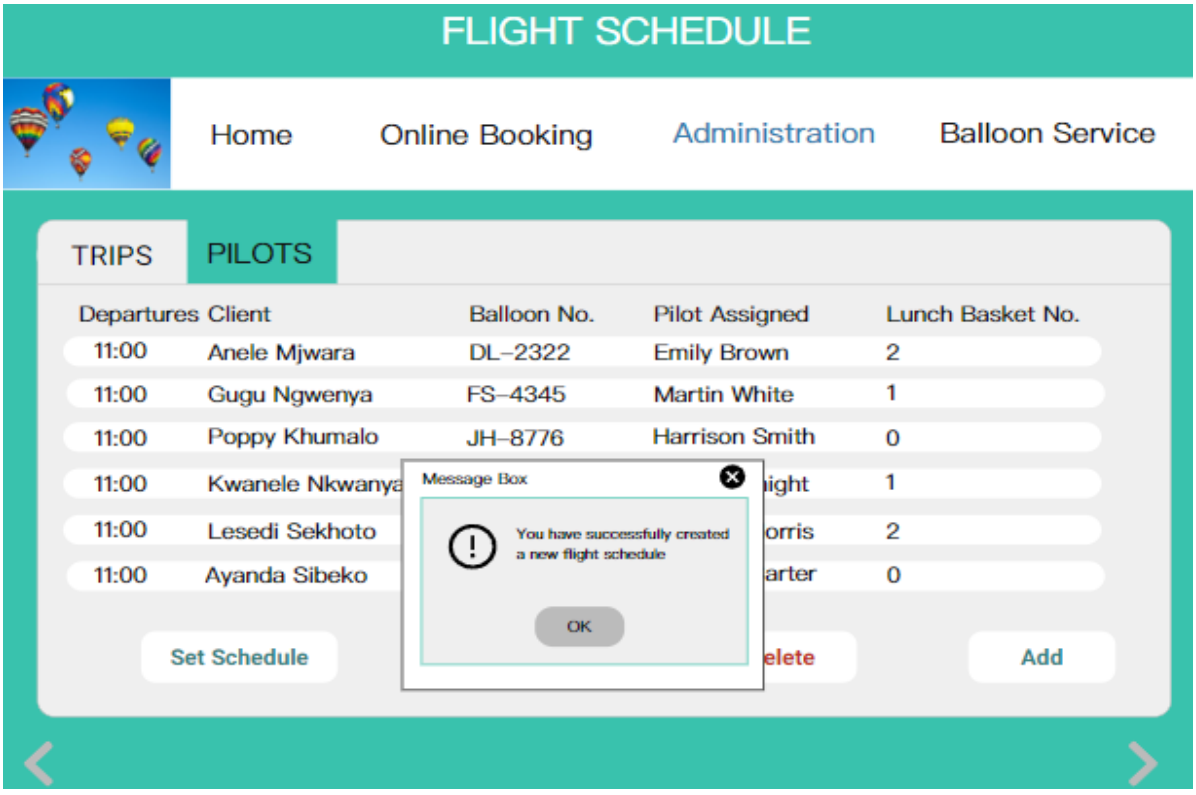

CHECK–IN PASSENGER
Home
Online Booking
Administration
Balloon Service
We hate to see you leave. You have been successfully refunded for your booking and your booking has been cancelled. Please check your emails for cancellation email.
OK

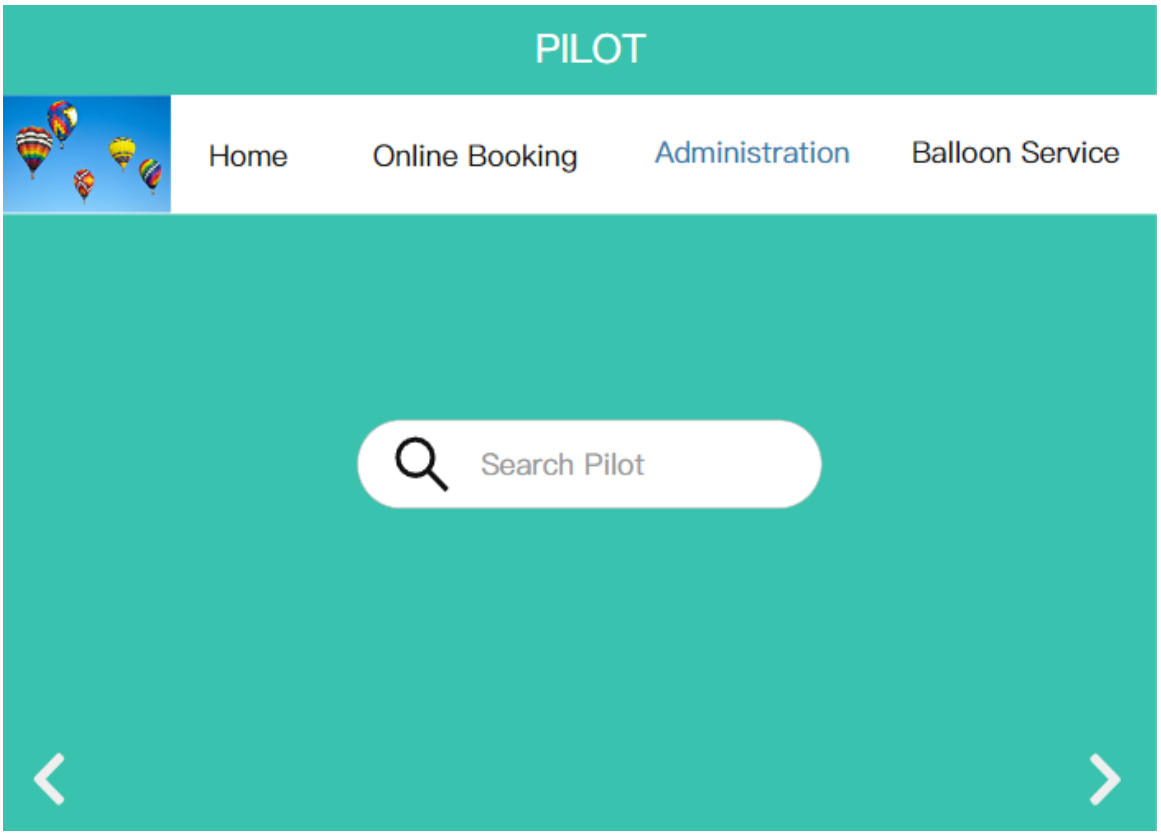

Wednesday, 22 October
11:38
You have cancelled your booking trip for Thursday. R13 500 has been transferred to your banking account...
8 Notifications

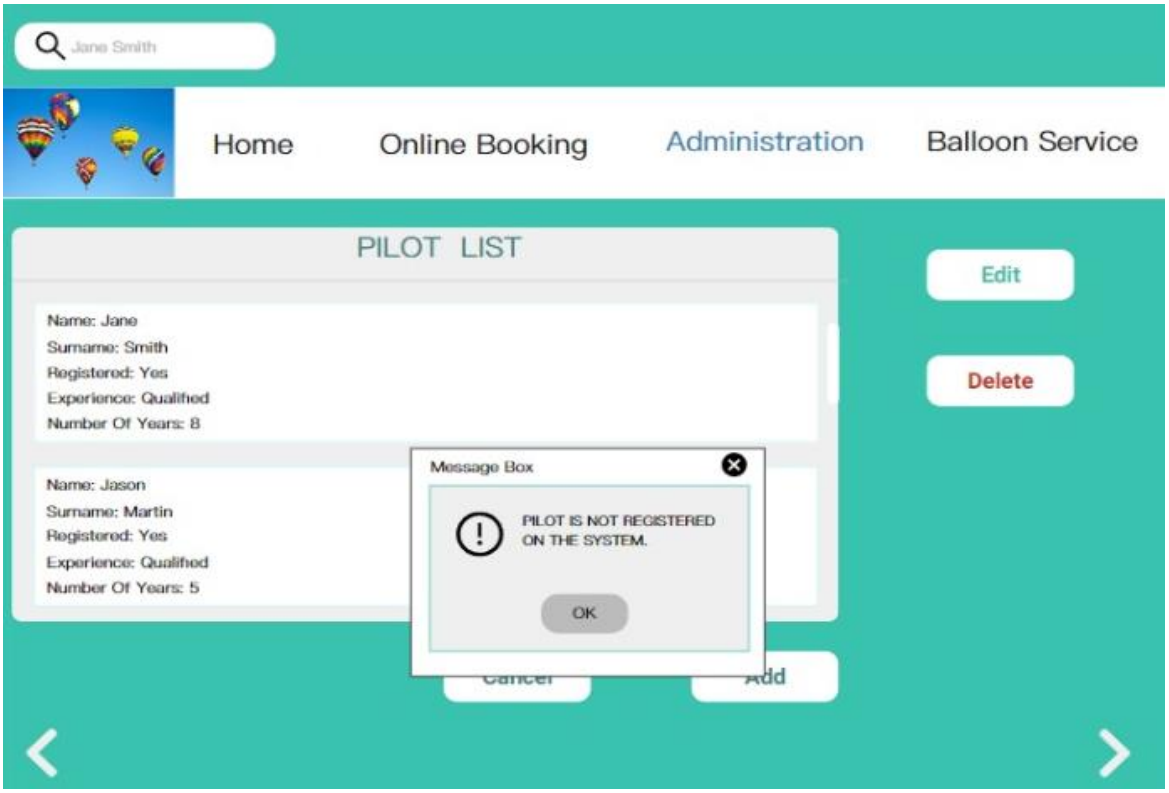

FLIGHT SCHEDULE
Home
Online Booking
Administration
Balloon Service
TRIPS
PILOTS
Departures Client Balloon No. Pilot Assigned Lunch Basket No.
11:00 Anele Mjwara DL–2322 John Doe 2
11:00 Gugu Ngwenya FS–4345 Martin White 1
11:00 Poppy Khumalo JH–8776 Harrison Smith 0
11:00 Kwanele Nkwanyana GT–2265 Wesley Knight 1
11:00 Lesedi Sekhoto SC–0931 Michael Norris 2
11:00 Ayanda Sibeko VQ–3259 Thomas Carter 0
Set Schedule
Cancel
Delete
Add

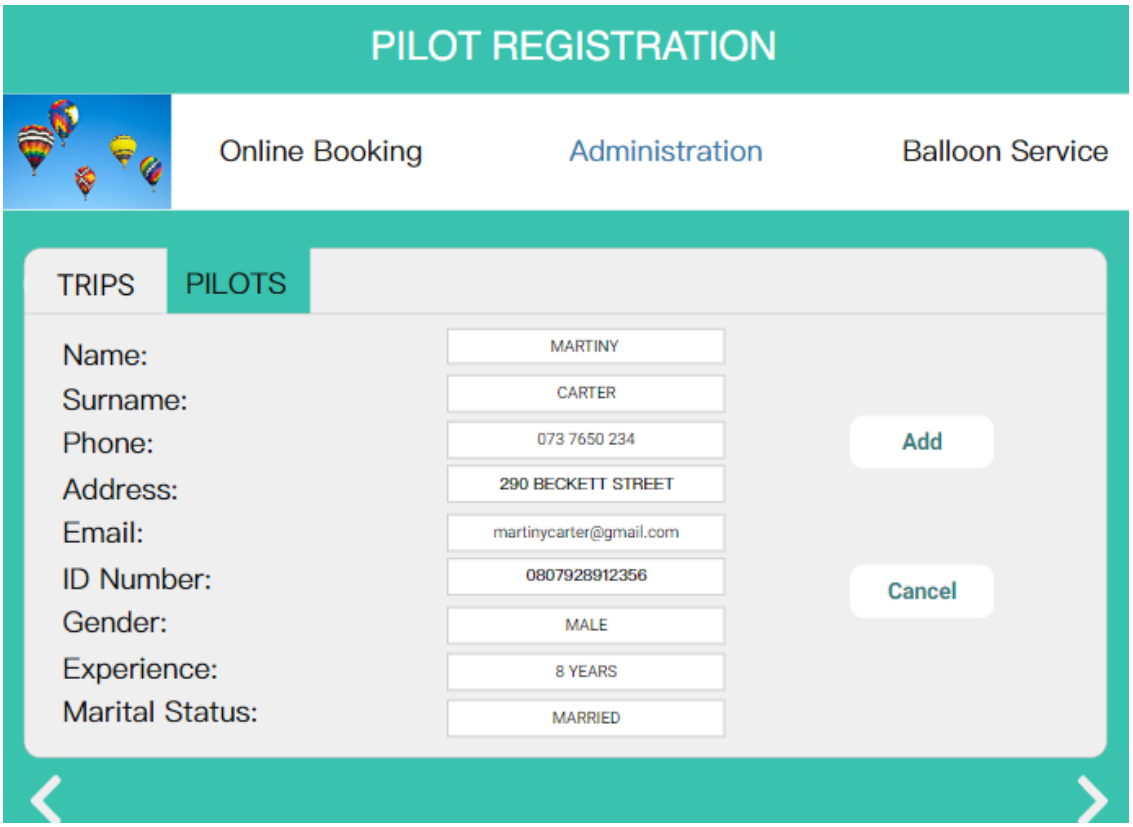

FLIGHT SCHEDULE
Home
Online Booking
Administration
Balloon Service
TRIPS
PILOTS
Schedule Balloon No. Pilot Assigned
12:00 DL–2322 John Doe
12:45 FS–4345 Martin White
13:20 JH–8776 Harrison Smith
13:30 GT–2265 Wesley Knight
14:00 SC–0931 Michael Norris
16:30 VQ–3259 Thomas Carter
Cancel
Delete
Add

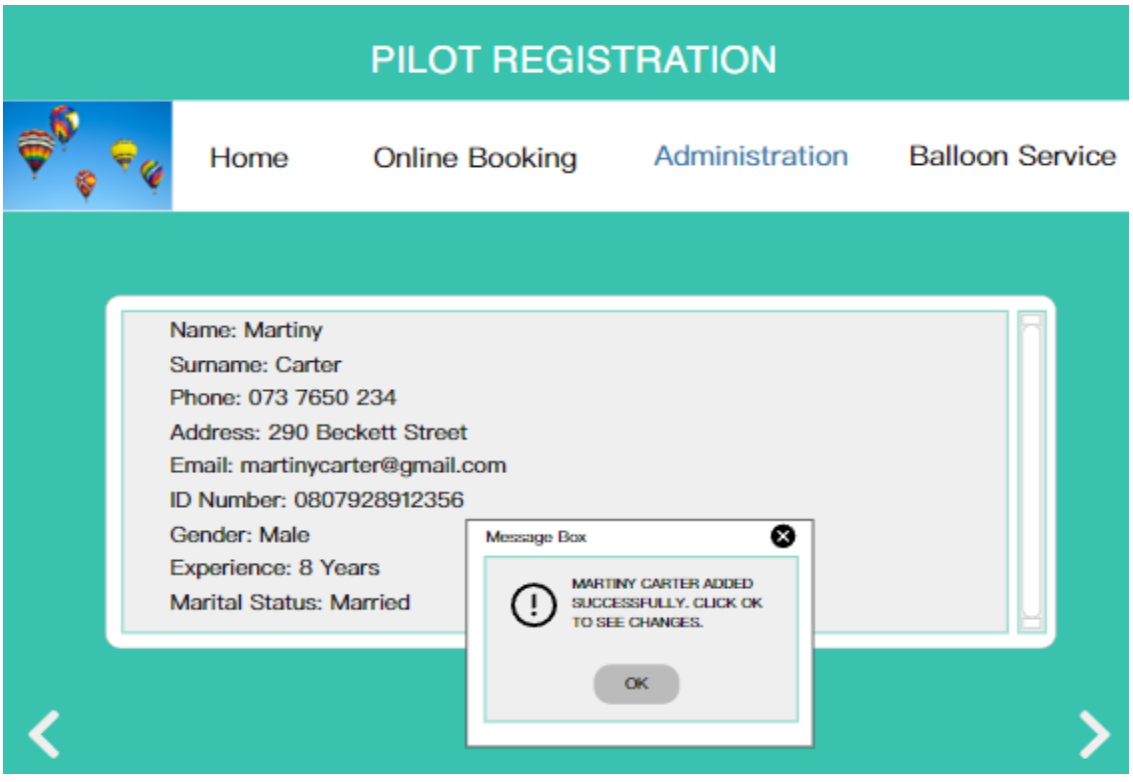

FLIGHT SCHEDULE
Home
Online Booking
Administration
Balloon Service
TRIPS
PILOTS
Client:
Departure:
Balloon Number:
Pilot:
Lunch Basket:
EMILY
BROWN
TR-7689
JASON SMITH
2
Add
Cancel

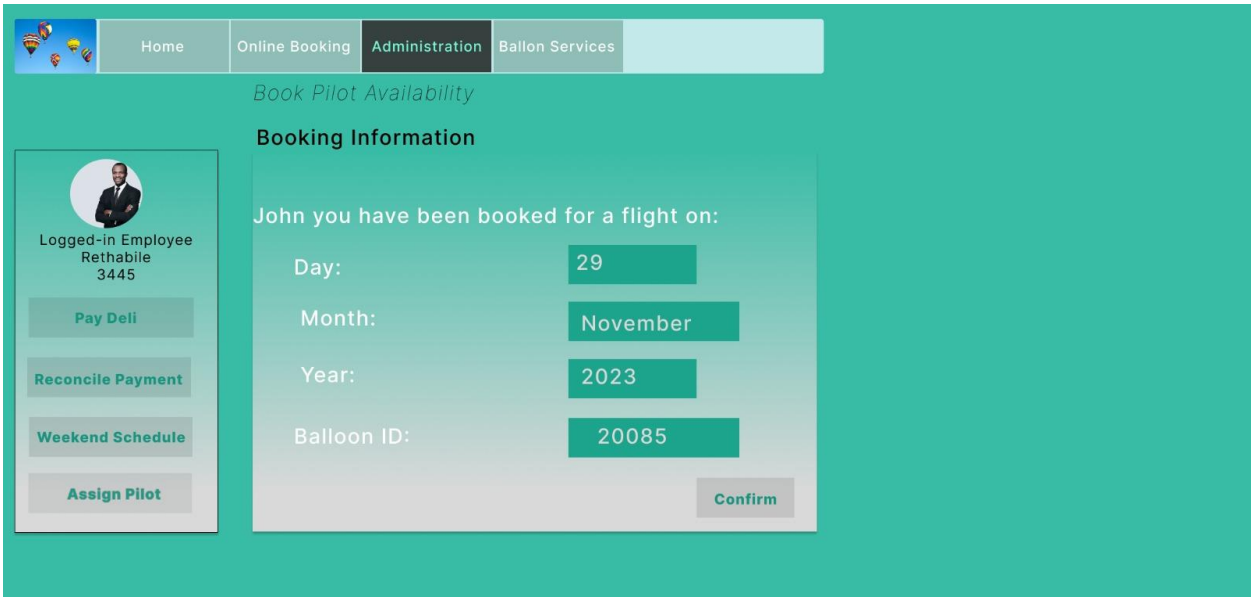

FLIGHT SCHEDULE
Home
Online Booking
Administration
Balloon Service
TRIPS
PILOTS
Departures
Client
Balloon No.
Pilot Assigned
Lunch Basket No.
11:00 Anele Mjwara DL–2322 Emily Brown 2
11:00 Gugu Ngwenya FS–4345 Martin White 1
11:00 Poppy Khumalo JH–8776 Harrison Smith 0
11:00 Kwanele Nkwanya
11:00 Lesedi Sekhoto
11:00 Ayanda Sibeko
Message Box
You have successfully created a new flight schedule
OK
Set Schedule
Add

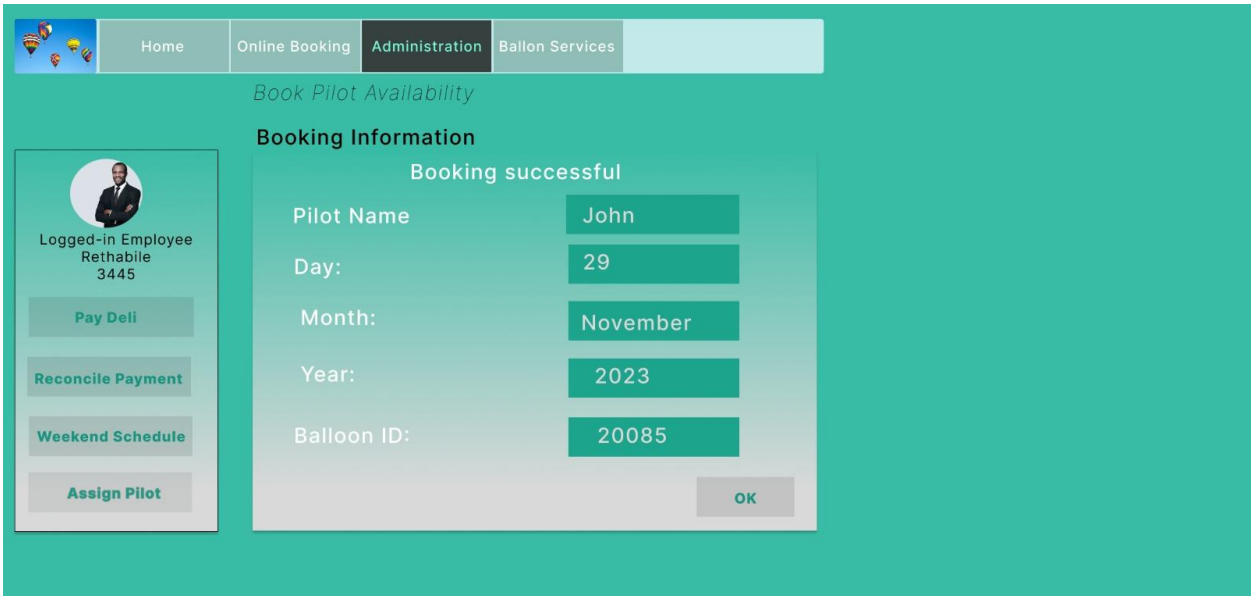

PILOT
Home
Online Booking
Administration
Balloon Service
Search Pilot

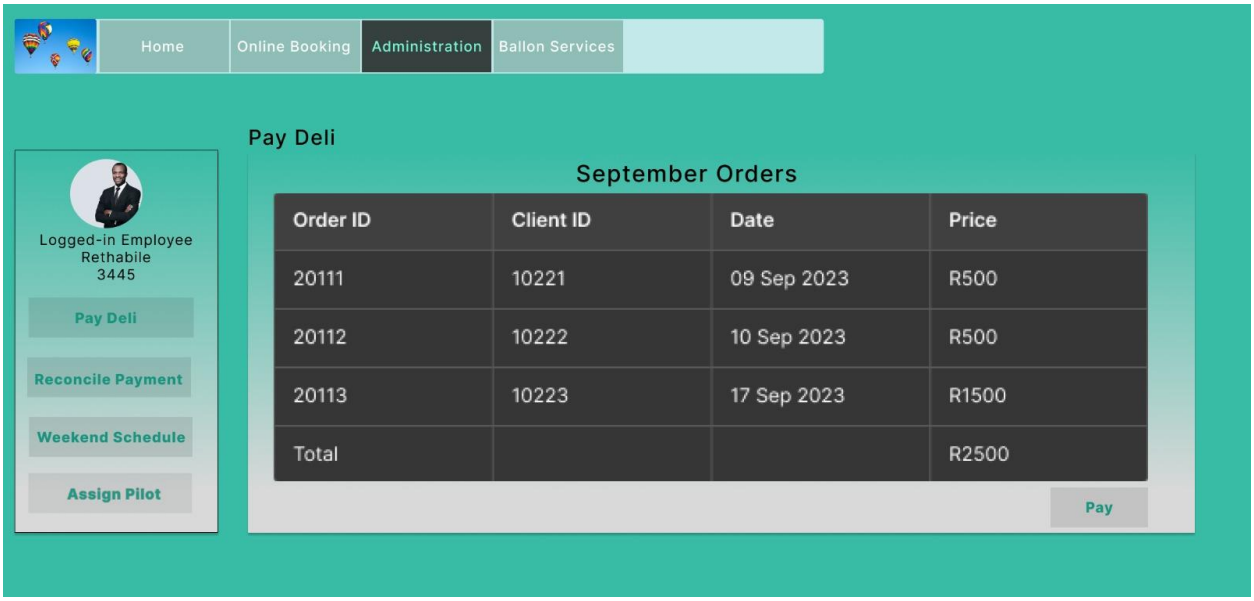

Jane Smith
Home
Online Booking
Administration
Balloon Service
PILOT LIST
Name: Jane
Surname: Smith
Registered: Yes
Experience: Qualified
Number Of Years: 8
Name: Jason
Surname: Martin
Registered: Yes
Experience: Qualified
Number Of Years: 5
Edit
Delete
Message Box
PILOT IS NOT REGISTERED ON THE SYSTEM.
OK
Add

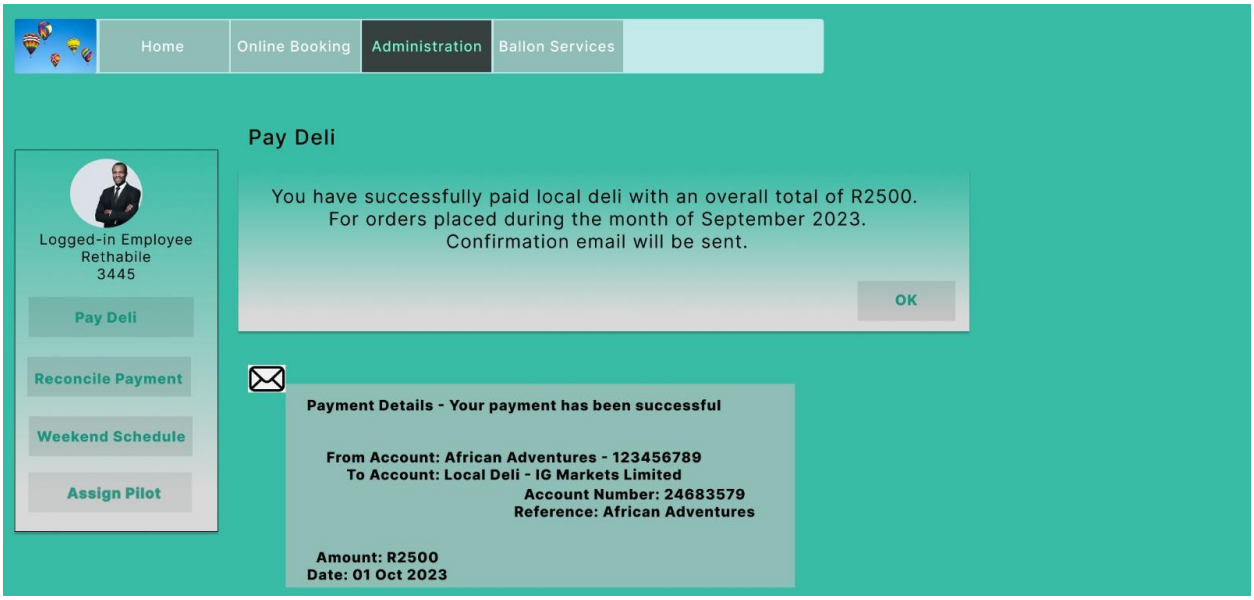

PILOT REGISTRATION
Online Booking
Administration
Balloon Service
TRIPS
PILOTS
Name:
MARTINY
Surname:
CARTER
Phone:
073 7650 234
Address:
290 BECKETT STREET
Email:
martinycarter@gmail.com
ID Number:
0807928912356
Gender:
MALE
Experience:
8 YEARS
Marital Status:
MARRIED
Add
Cancel

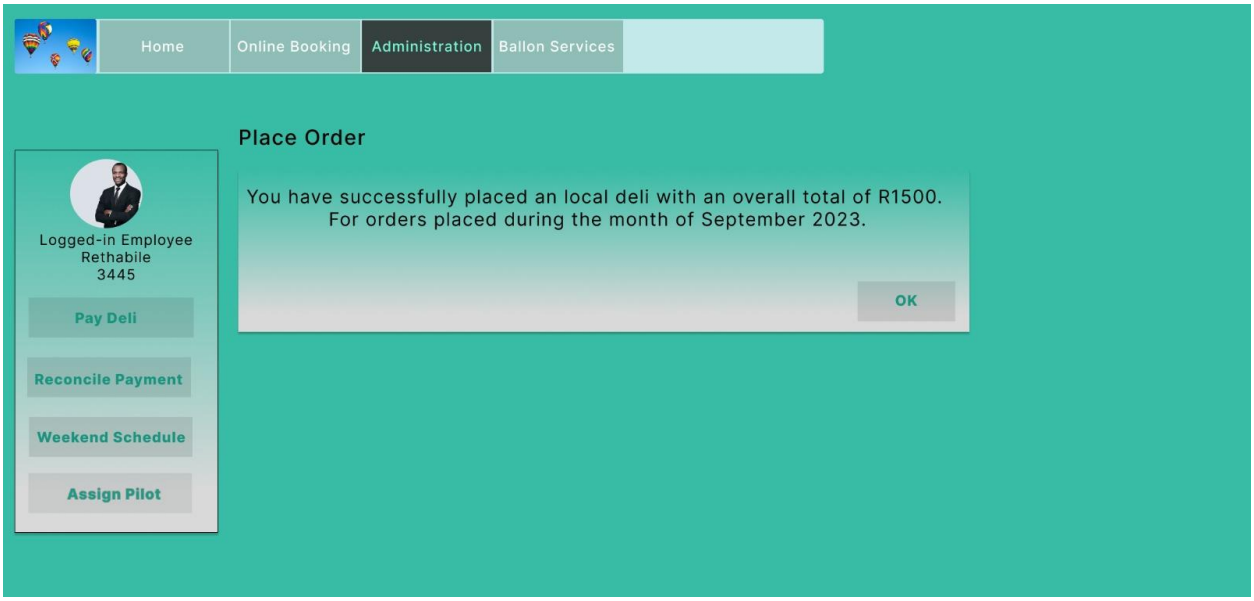

PILOT REGISTRATION
Home
Online Booking
Administration
Balloon Service
Name: Martiny
Surname: Carter
Phone: 073 7650 234
Address: 290 Beckett Street
Email: martinycarter@gmail.com
ID Number: 0807928912356
Gender: Male
Experience: 8 Years
Marital Status: Married
Message Box
MARTINY CARTER ADDED SUCCESSFULLY. CLICK OK TO SEE CHANGES.
OK

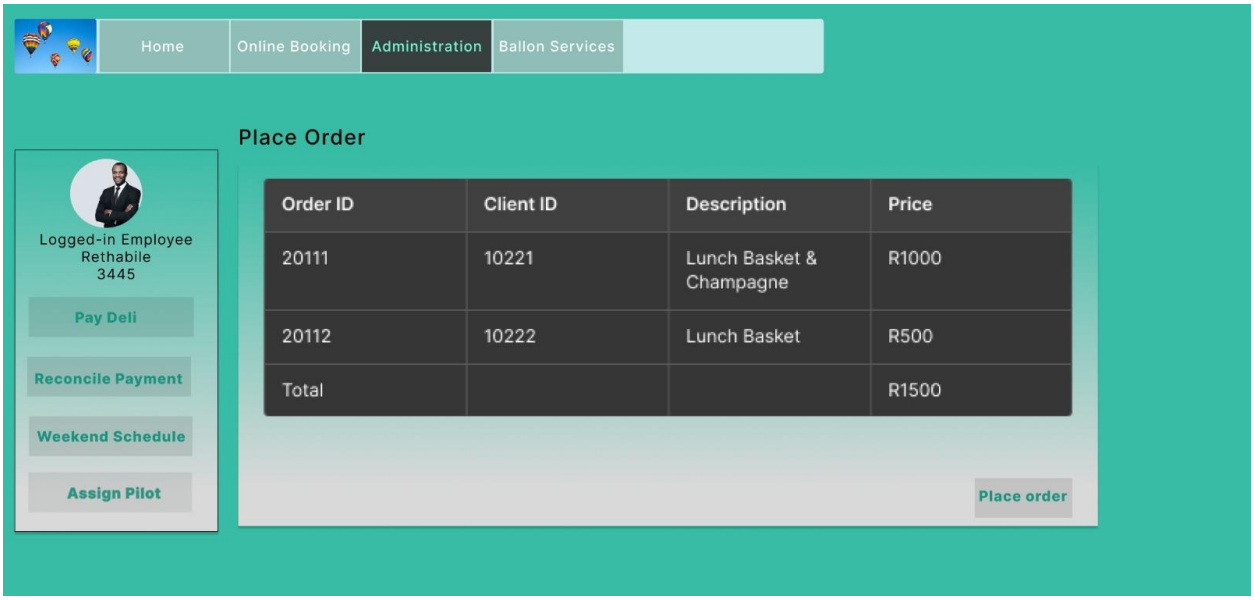

Home
Online Booking
Administration
Ballon Services
Book Pilot Availability
Booking Information
Logged-in Employee
Rethabile
3445
Pay Deli
Reconcile Payment
Weekend Schedule
Assign Pilot
John you have been booked for a flight on:
Day:
29
Month:
November
Year:
2023
Balloon ID:
20085
Confirm

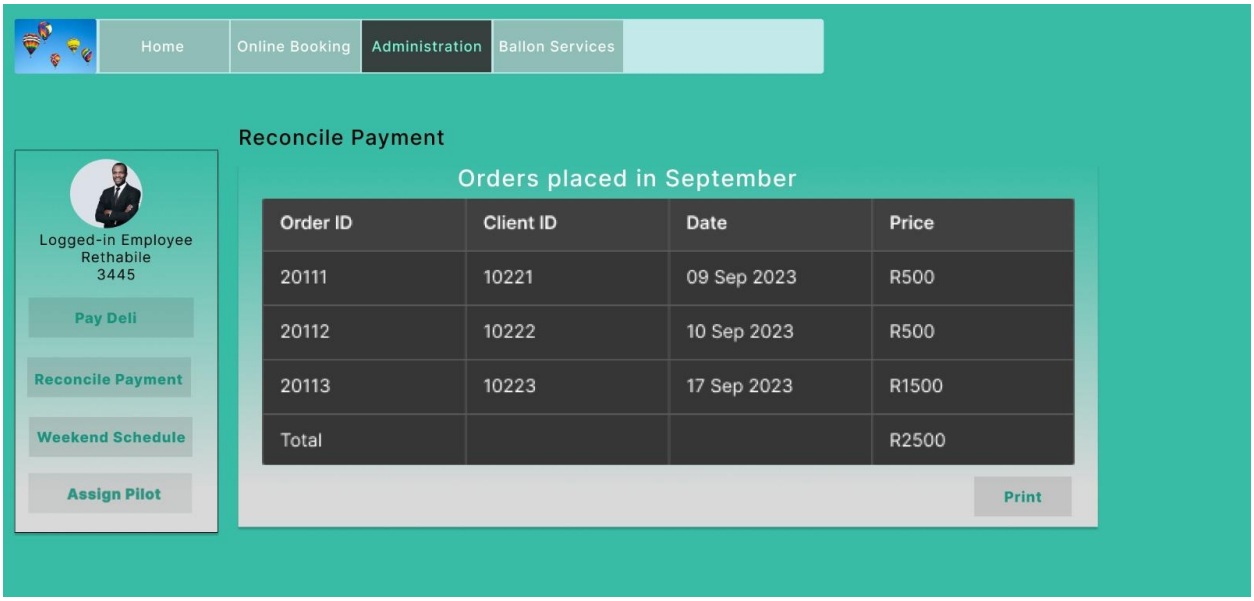

Home
Online Booking
Administration
Ballon Services
Book Pilot Availability
Booking Information
Logged-in Employee
Rethabile
3445
Pay Deli
Reconcile Payment
Weekend Schedule
Assign Pilot
Booking successful
Pilot Name
John
Day:
29
Month:
November
Year:
2023
Balloon ID:
20085
OK

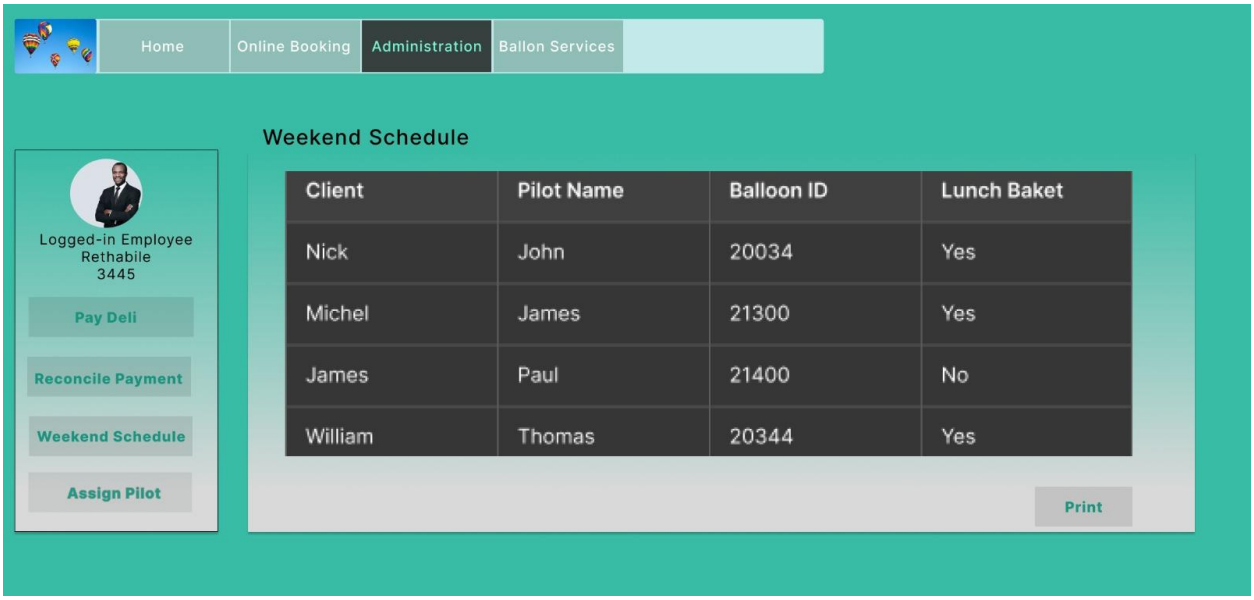

Home
Online Booking
Administration
Ballon Services
Pay Deli
Logged-in Employee
Rethabile
3445
Pay Deli
Reconcile Payment
Weekend Schedule
Assign Pilot
September Orders
Order ID
Client ID
Date
Price
20111
10221
09 Sep 2023
R500
20112
10222
10 Sep 2023
R500
20113
10223
17 Sep 2023
R1500
Total
R2500
Pay

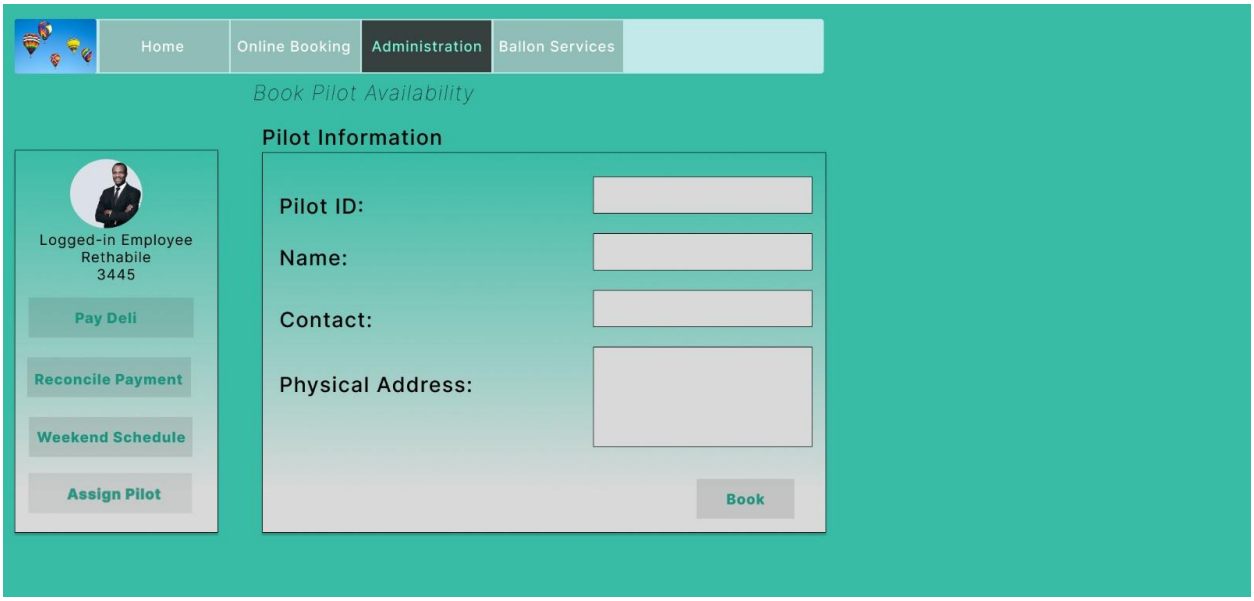

Home
Online Booking
Administration
Ballon Services
Pay Deli
Logged-in Employee
Rethabile
3445
Pay Deli
Reconcile Payment
Weekend Schedule
Assign Pilot
You have successfully paid local deli with an overall total of R2500.
For orders placed during the month of September 2023.
Confirmation email will be sent.
OK
Payment Details - Your payment has been successful
From Account: African Adventures - 123456789
To Account: Local Deli - IG Markets Limited
Account Number: 24683579
Reference: African Adventures
Amount: R2500
Date: 01 Oct 2023


Home
Online Booking
Administration
Ballon Services
Logged-in Employee
Rethabile
3445
Pay Deli
Reconcile Payment
Weekend Schedule
Assign Pilot
Place Order
You have successfully placed an local deli with an overall total of R1500.
For orders placed during the month of September 2023.
OK

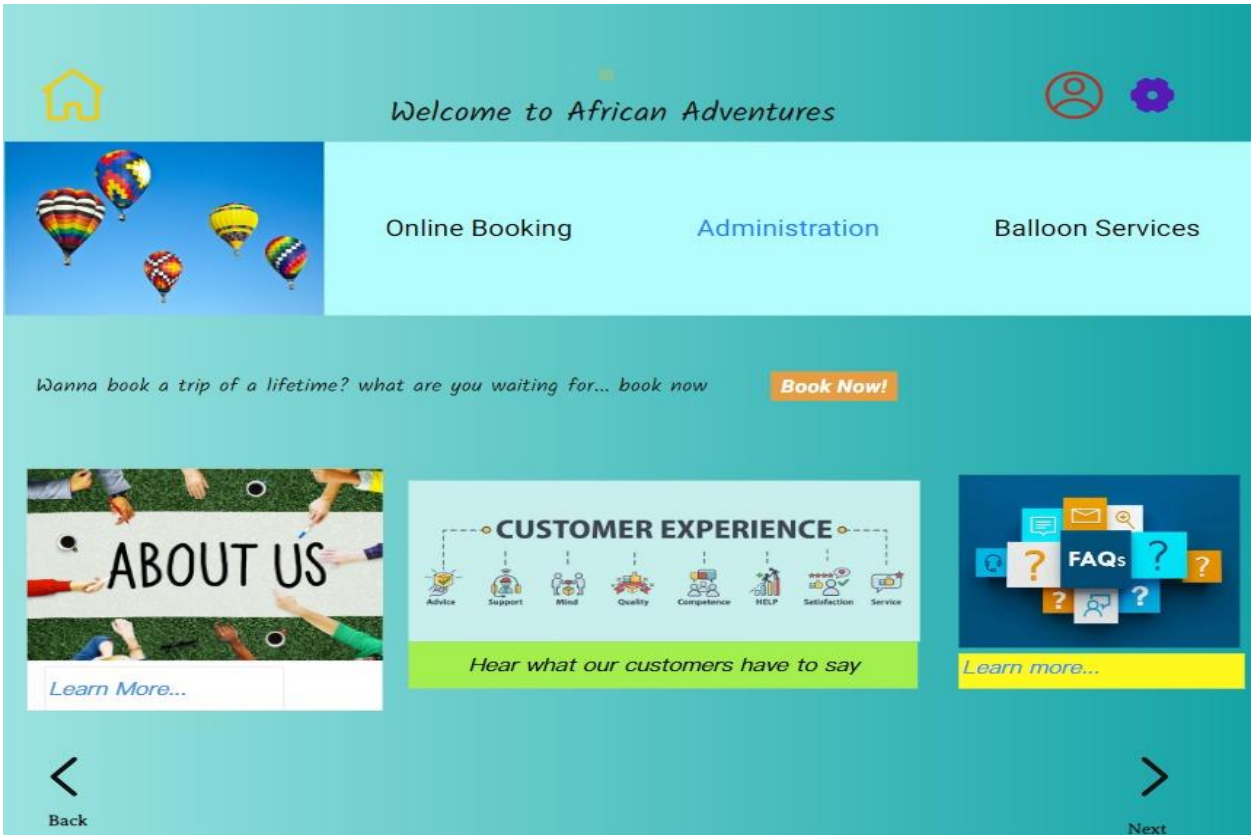

Home
Online Booking
Administration
Ballon Services
Logged-in Employee
Rethabile
3445
Pay Deli
Reconcile Payment
Weekend Schedule
Assign Pilot
Place Order
Order ID
Client ID
Description
Price
20111
10221
Lunch Basket & Champagne
R1000
20112
10222
Lunch Basket
R500
Total
R1500
Place order

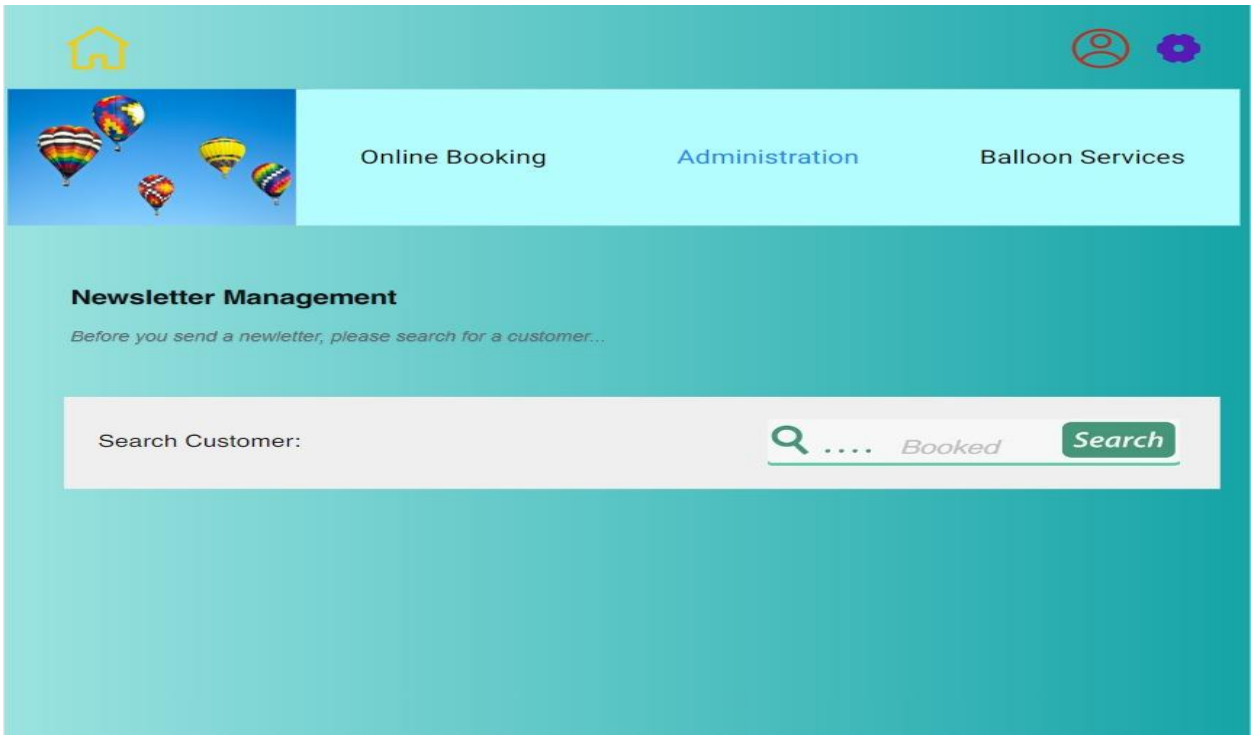

Home
Online Booking
Administration
Ballon Services
Reconcile Payment
Logged-in Employee
Rethabile
3445
Pay Deli
Reconcile Payment
Weekend Schedule
Assign Pilot
Orders placed in September
Order ID
Client ID
Date
Price
20111
10221
09 Sep 2023
R500
20112
10222
10 Sep 2023
R500
20113
10223
17 Sep 2023
R1500
Total
R2500
Print

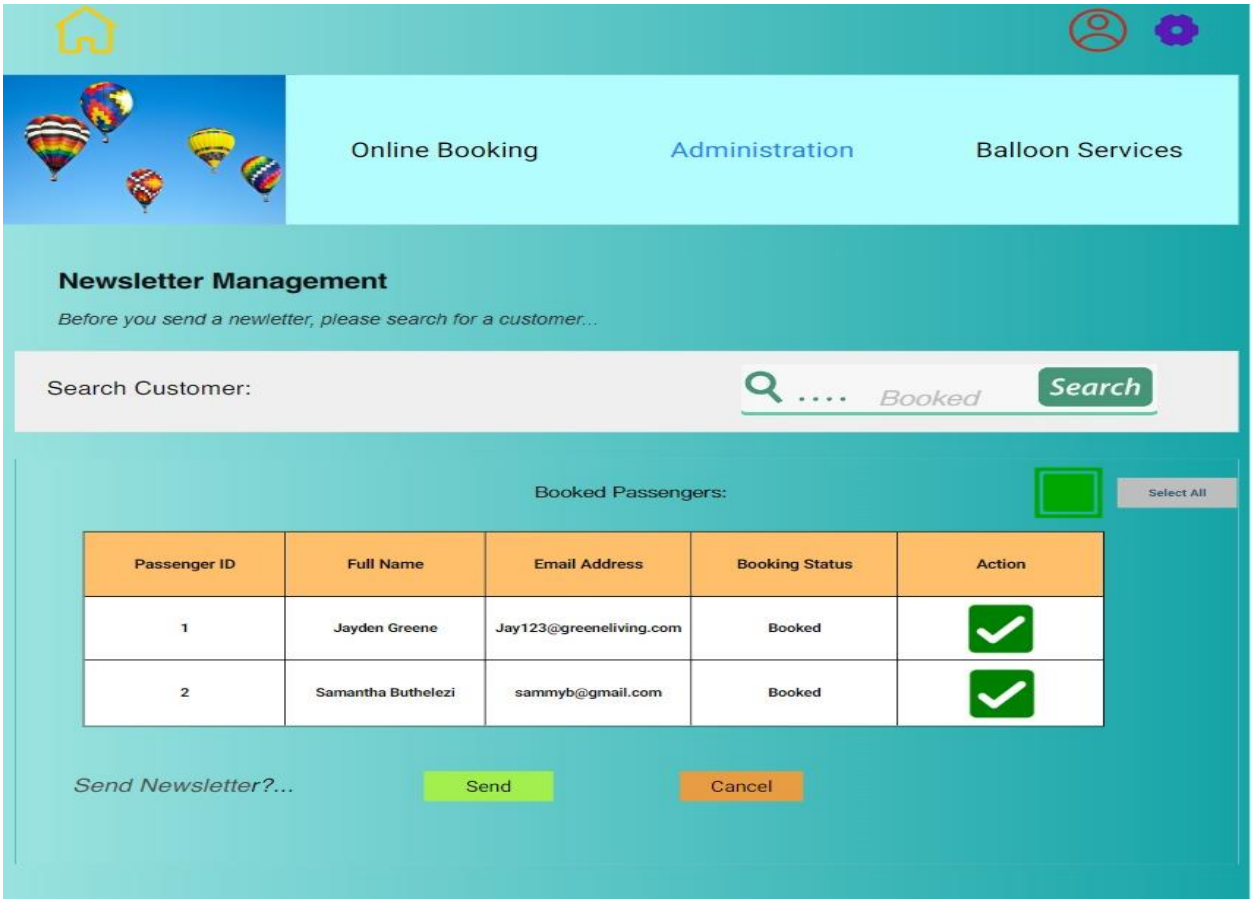

Home
Online Booking
Administration
Ballon Services
Weekend Schedule
Logged-in Employee
Rethabile
3445
Pay Deli
Reconcile Payment
Weekend Schedule
Assign Pilot
Client
Pilot Name
Balloon ID
Lunch Baket
Nick
John
20034
Yes
Michel
James
21300
Yes
James
Paul
21400
No
William
Thomas
20344
Yes
Print

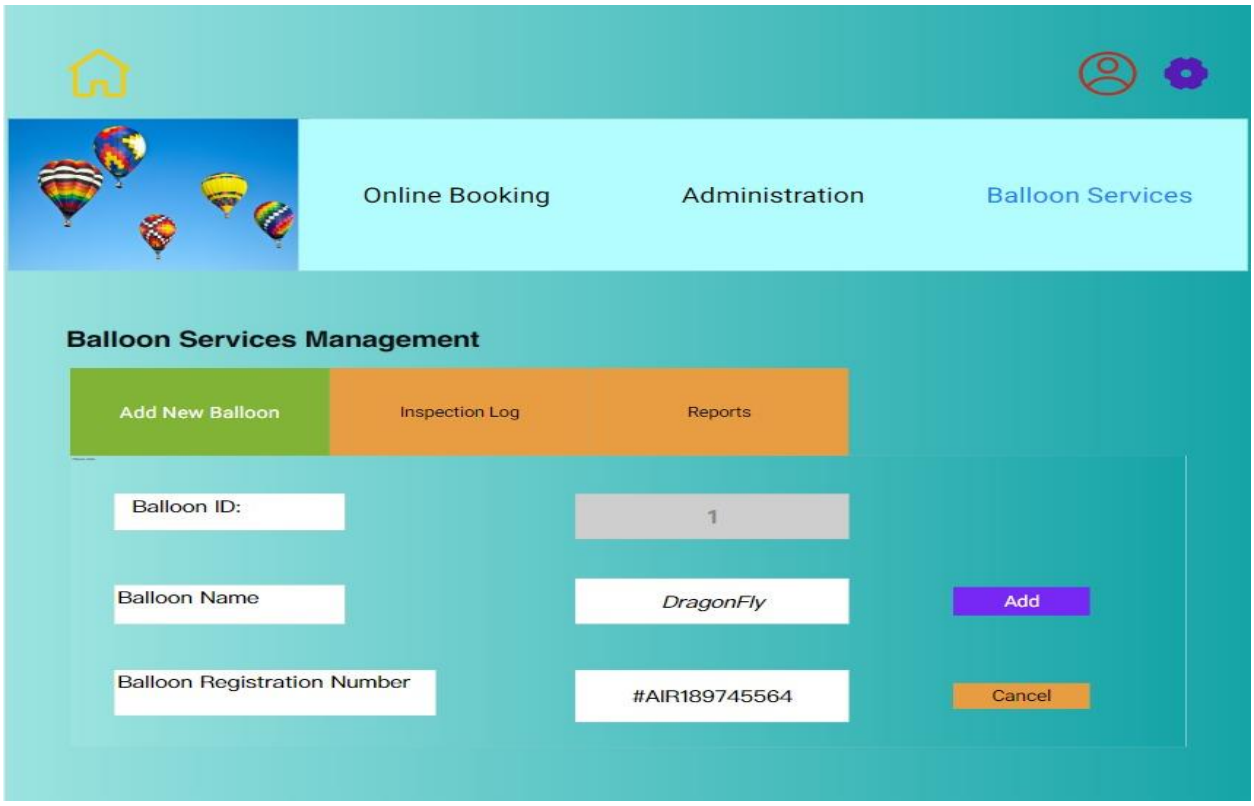


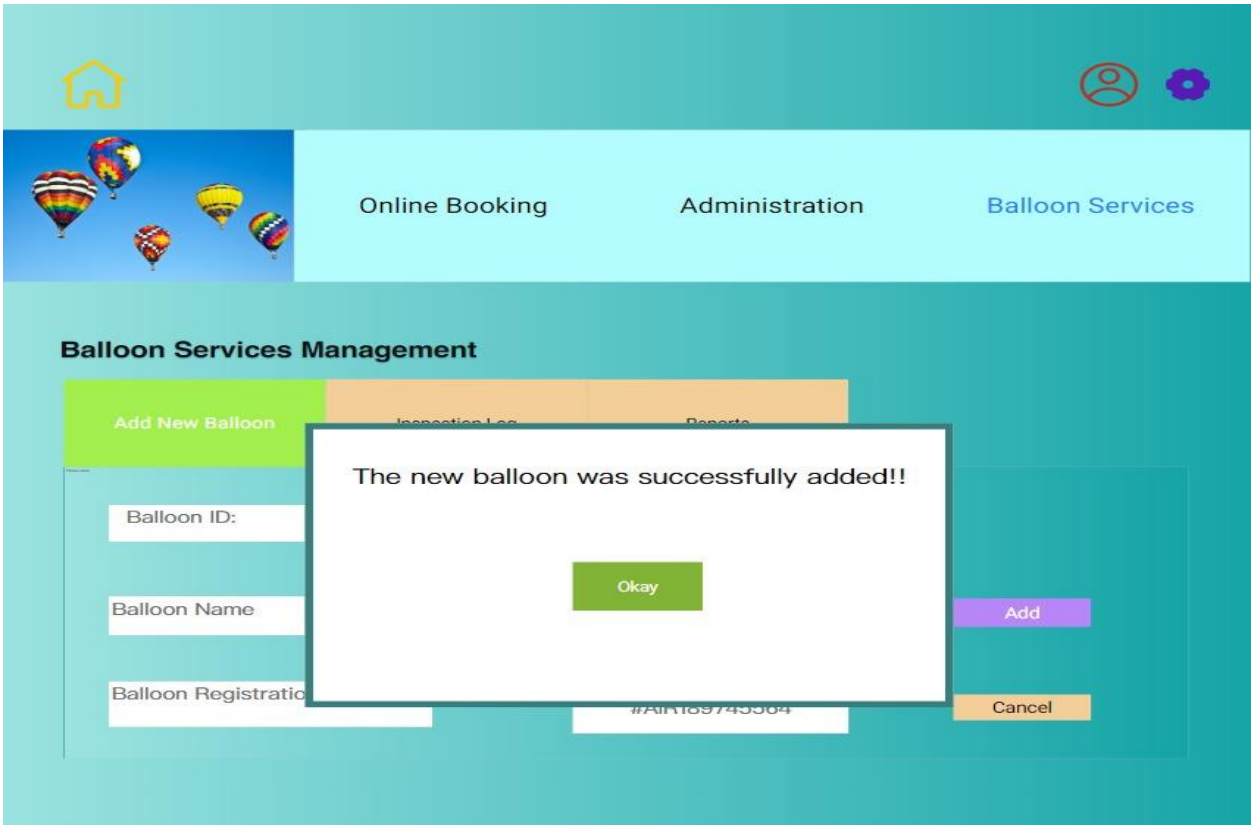
Balloon Service Subsystem

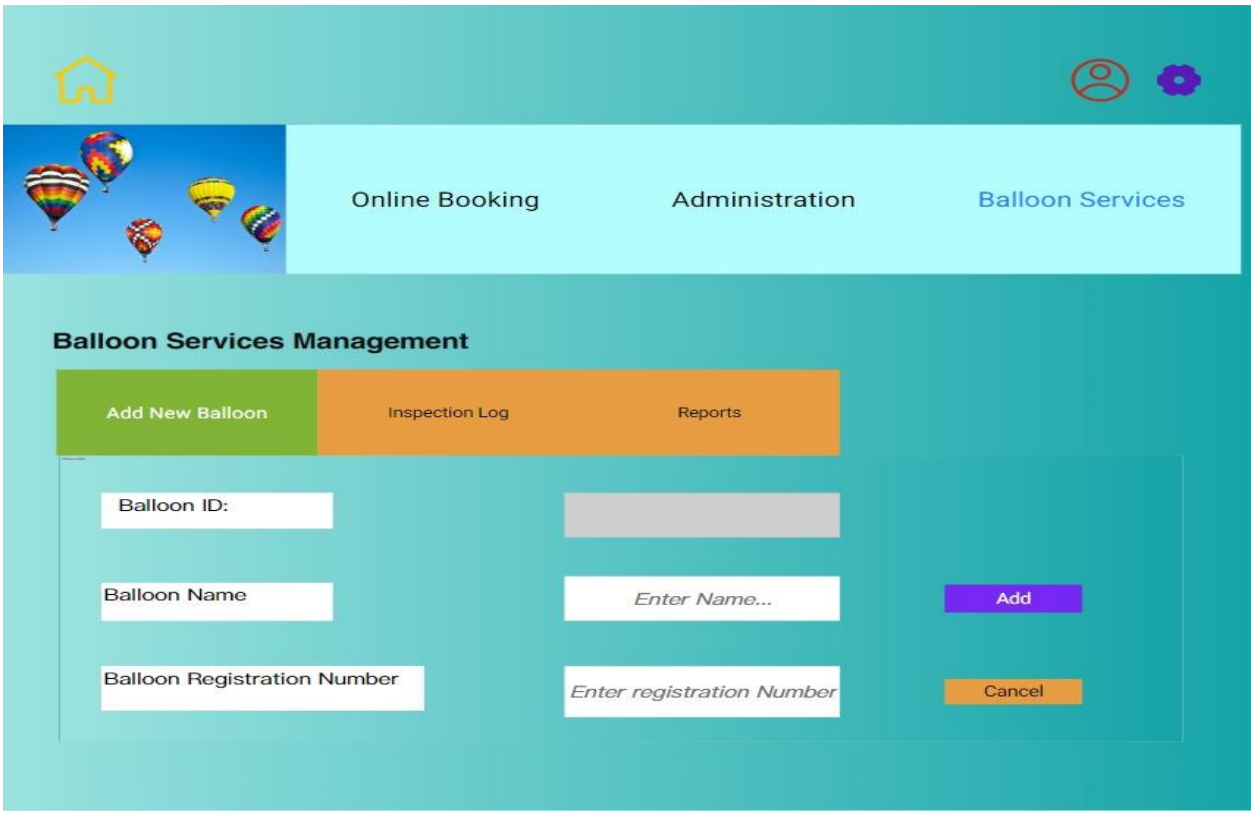


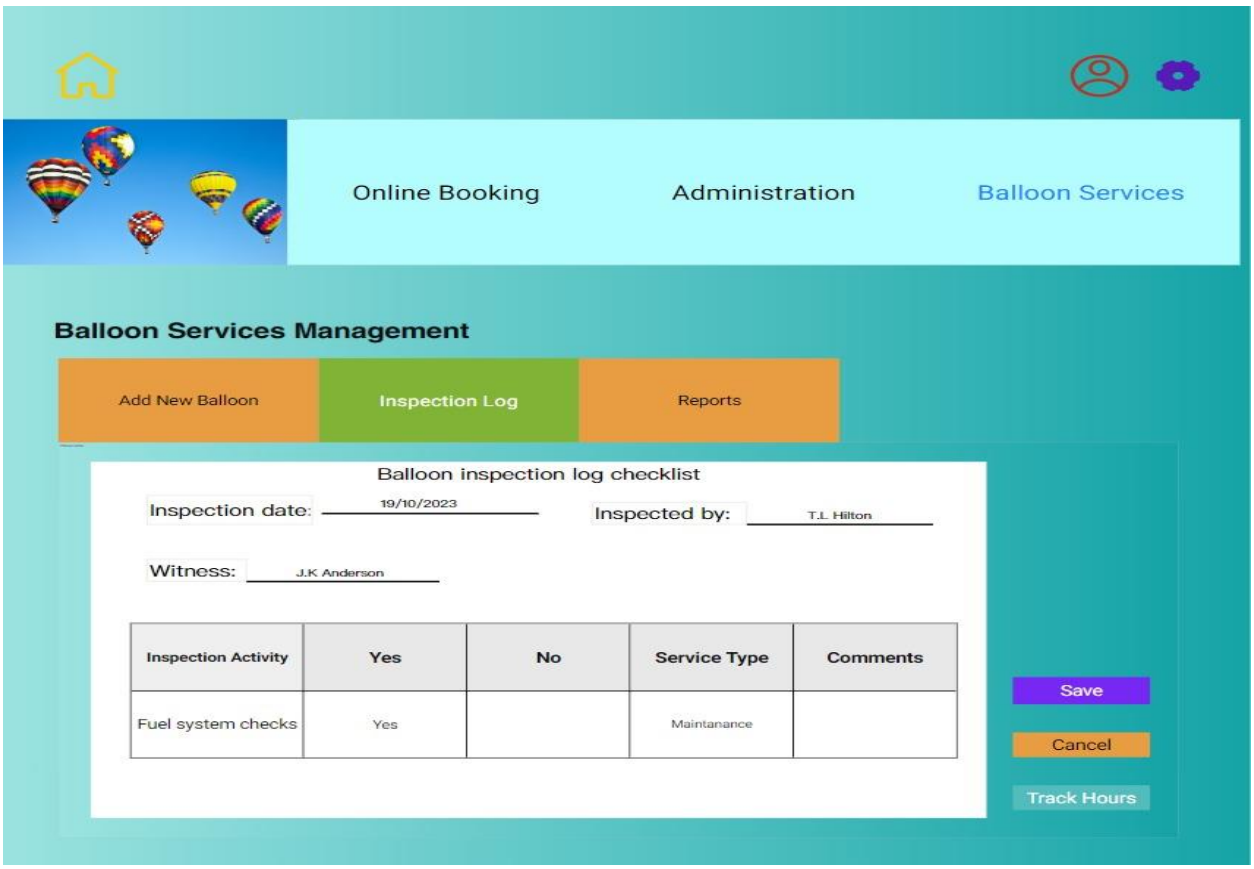

Online Booking
Administration
Balloon Services
Newsletter Management
Before you send a newletter, please search for a customer...
Search Customer:
.... Booked
Search

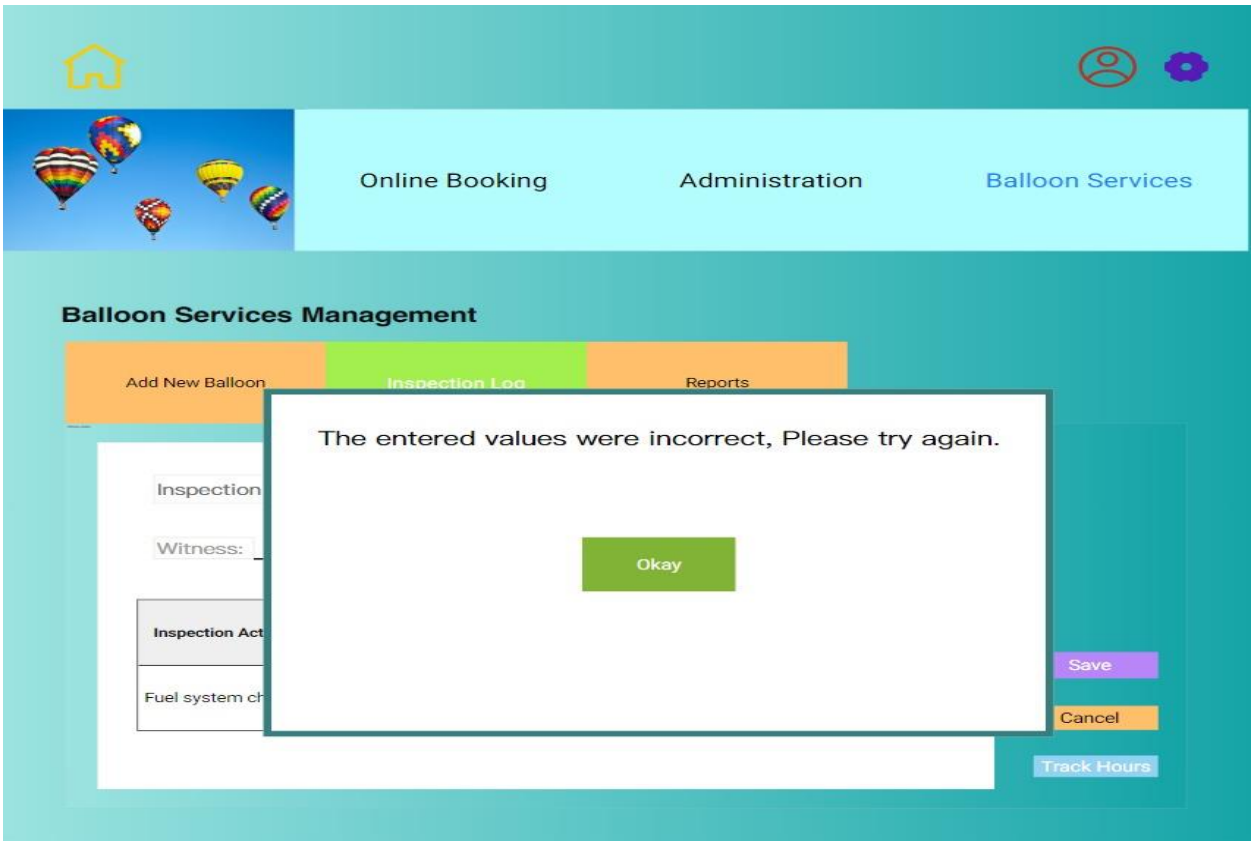

Online Booking
Administration
Balloon Services
Newsletter Management
Before you send a newletter, please search for a customer...
Search Customer:
.... Booked
Search
Booked Passengers:
Select All
Passenger ID
Full Name
Email Address
Booking Status
Action
1
Jayden Greene
Jay123@greeneliving.com
Booked
2
Samantha Buthelezi
sammyb@gmail.com
Booked
Send Newsletter?...
Send
Cancel

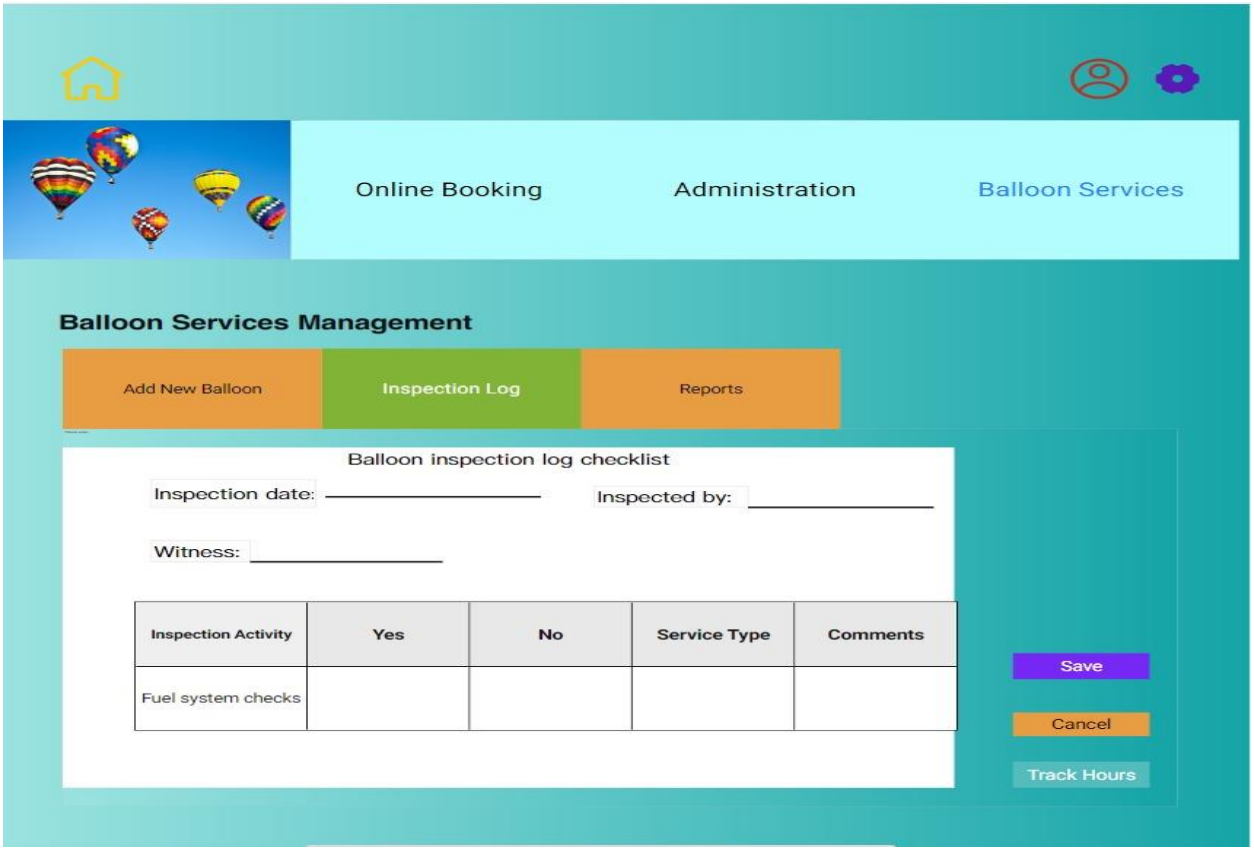

Online Booking
Administration
Balloon Services
Balloon Services Management
Add New Balloon
Inspection Log
Reports
Balloon ID:
1
Balloon Name
DragonFly
Add
Balloon Registration Number
#AIR189745564
Cancel

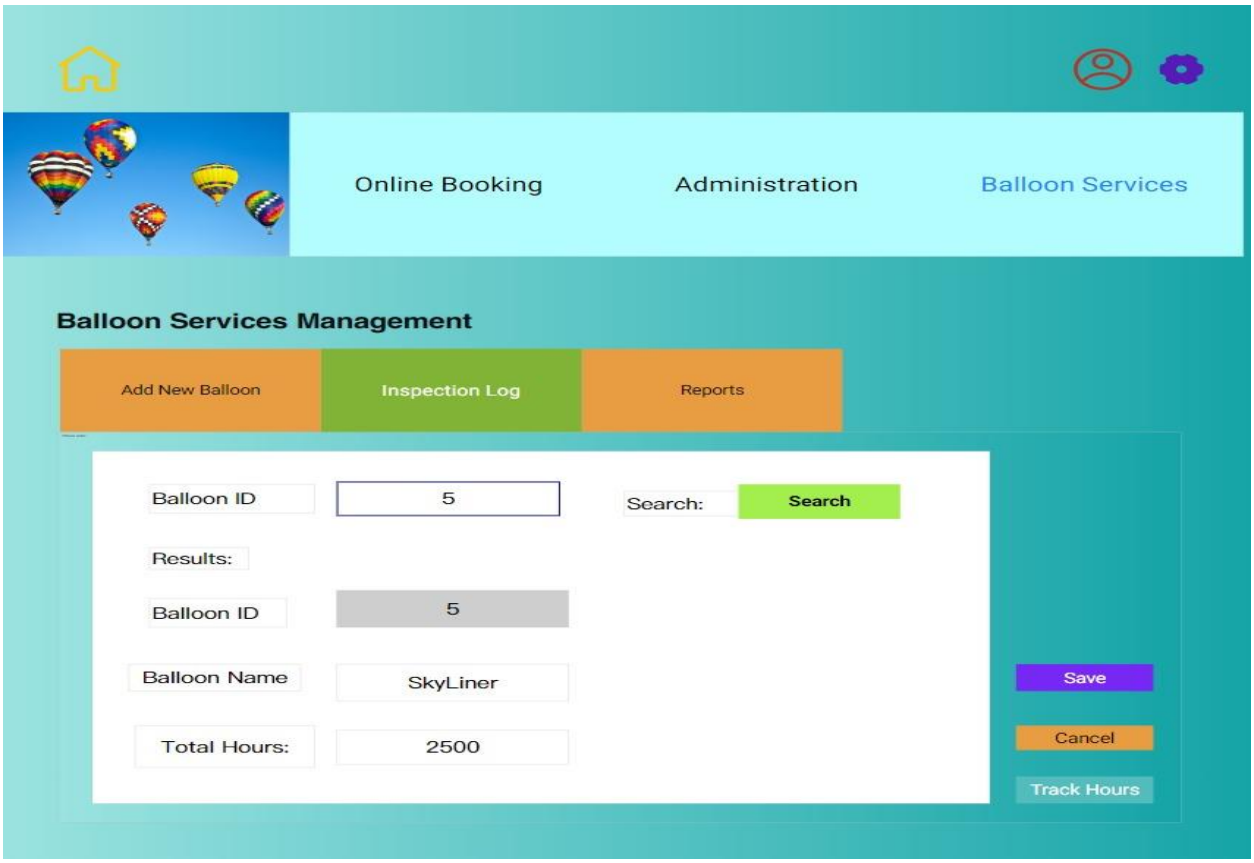

Online Booking
Administration
Balloon Services
Balloon Services Management
Add New Balloon
The new balloon was successfully added!!
Okay
Balloon ID:
Balloon Name
Add
Cancel

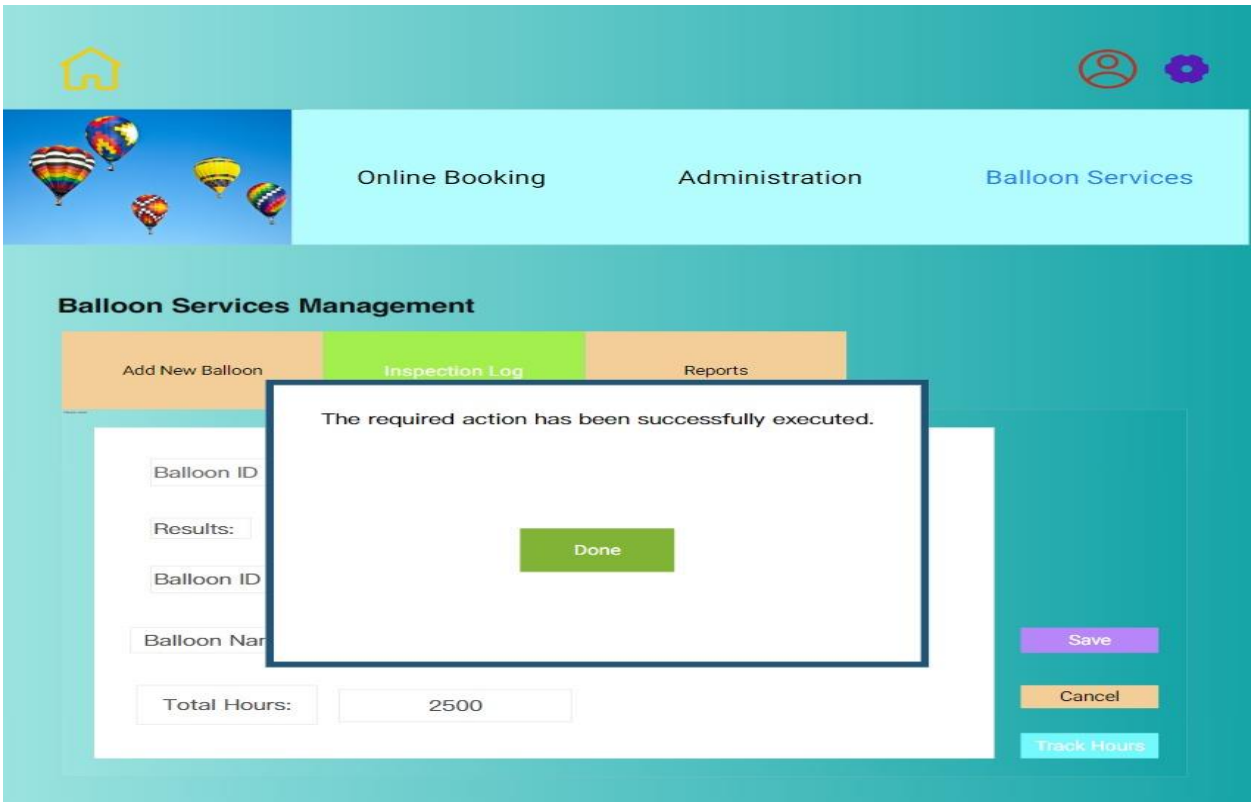

Online Booking
Administration
Balloon Services
Balloon Services Management
Add New Balloon
Inspection Log
Reports
Balloon ID:
Balloon Name
Enter Name...
Add
Balloon Registration Number
Enter registration Number
Cancel

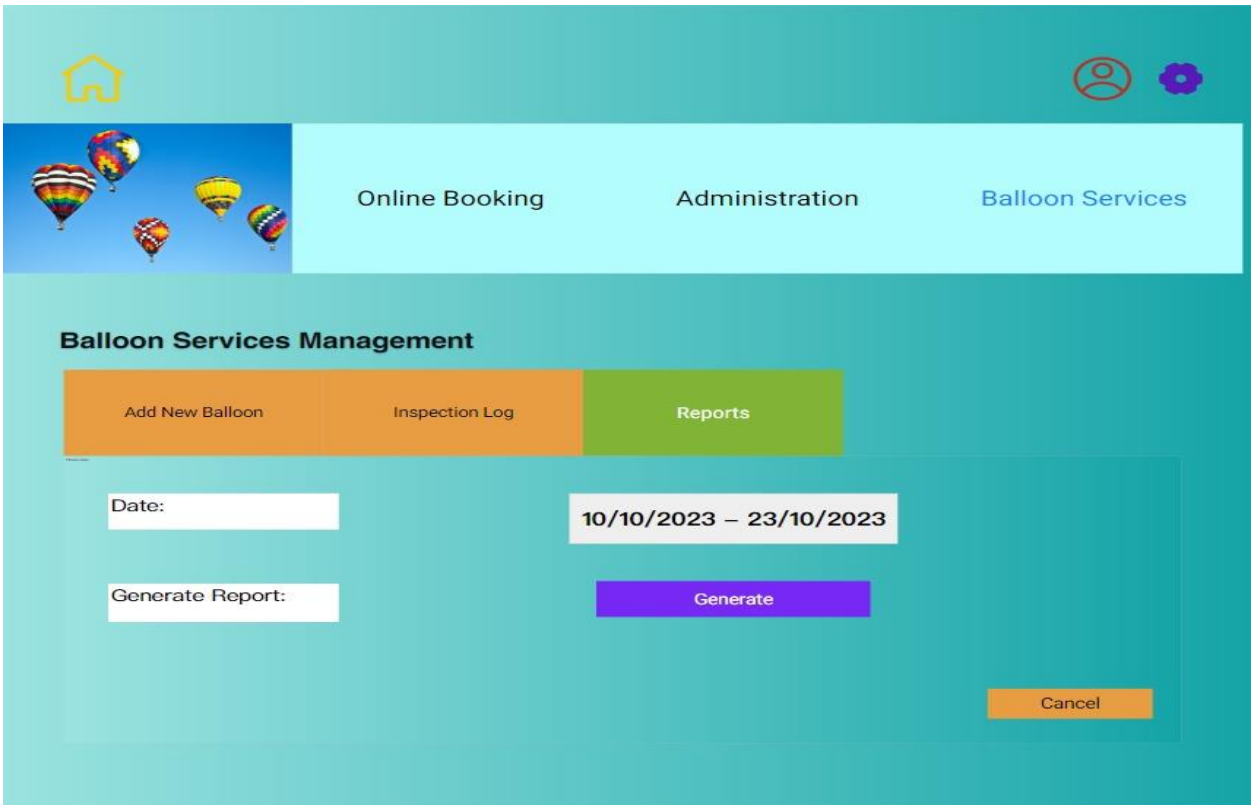

Online Booking
Administration
Balloon Services
Balloon Services Management
Add New Balloon
Inspection Log
Reports
Balloon inspection log checklist
Inspection date: 19/10/2023
Inspected by: T.L Hilton
Witness: J.K Anderson
Inspection Activity
Yes
No
Service Type
Comments
Fuel system checks
Yes
Maintanance
Save
Cancel
Track Hours

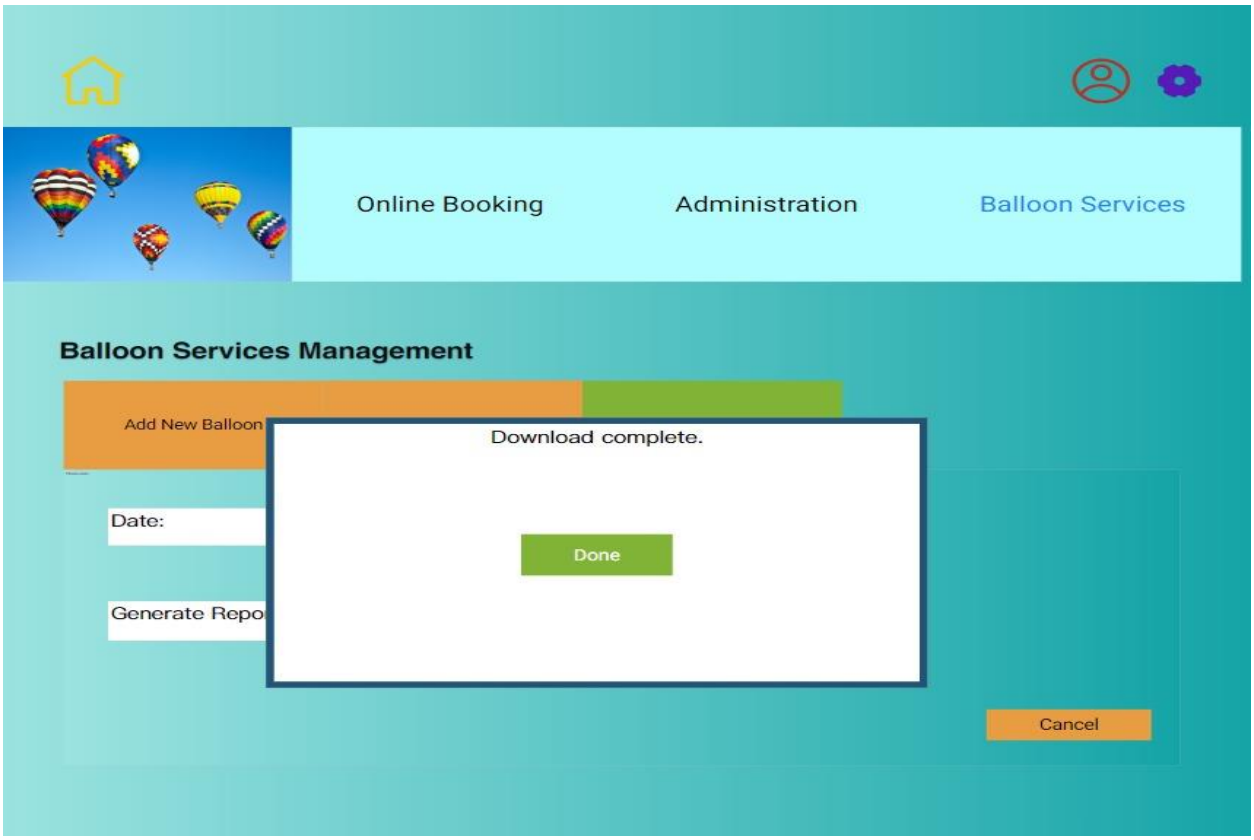

Online Booking
Administration
Balloon Services
Balloon Services Management
Add New Balloon
Inspection Log
Reports
The entered values were incorrect, Please try again.
Okay
Inspection
Witness:
Inspection Act
Fuel system ch
Save
Cancel
Track Hours


Online Booking
Administration
Balloon Services
Balloon Services Management
Add New Balloon
Inspection Log
Reports
Balloon inspection log checklist
Inspection date:
Inspected by:
Witness:
Inspection Activity
Yes
No
Service Type
Comments
Fuel system checks
Save
Cancel
Track Hours


Online Booking
Administration
Balloon Services
Balloon Services Management
Add New Balloon
Inspection Log
Reports
Balloon ID
5
Search:
Search
Results:
Balloon ID
5
Balloon Name
SkyLiner
Total Hours:
2500
Save
Cancel
Track Hours


Online Booking
Administration
Balloon Services
Balloon Services Management
Add New Balloon
Inspection Log
Reports
The required action has been successfully executed.
Done
Balloon ID
Results:
Balloon ID
Balloon Nar
Total Hours:
2500
Save
Cancel
Track Hours

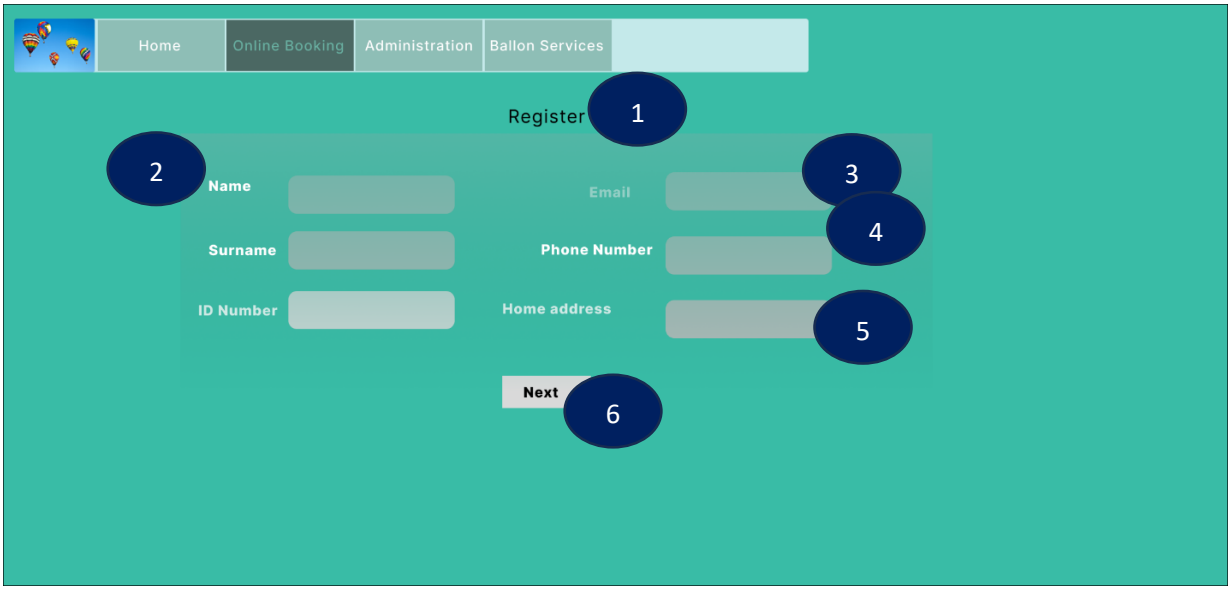

Online Booking
Administration
Balloon Services
Balloon Services Management
Add New Balloon
Inspection Log
Reports
Date:
10/10/2023 – 23/10/2023
Generate Report:
Generate
Cancel


Online Booking
Administration
Balloon Services
Balloon Services Management
Add New Balloon
Download complete.
Done
Date:
Generate Repo
Cancel

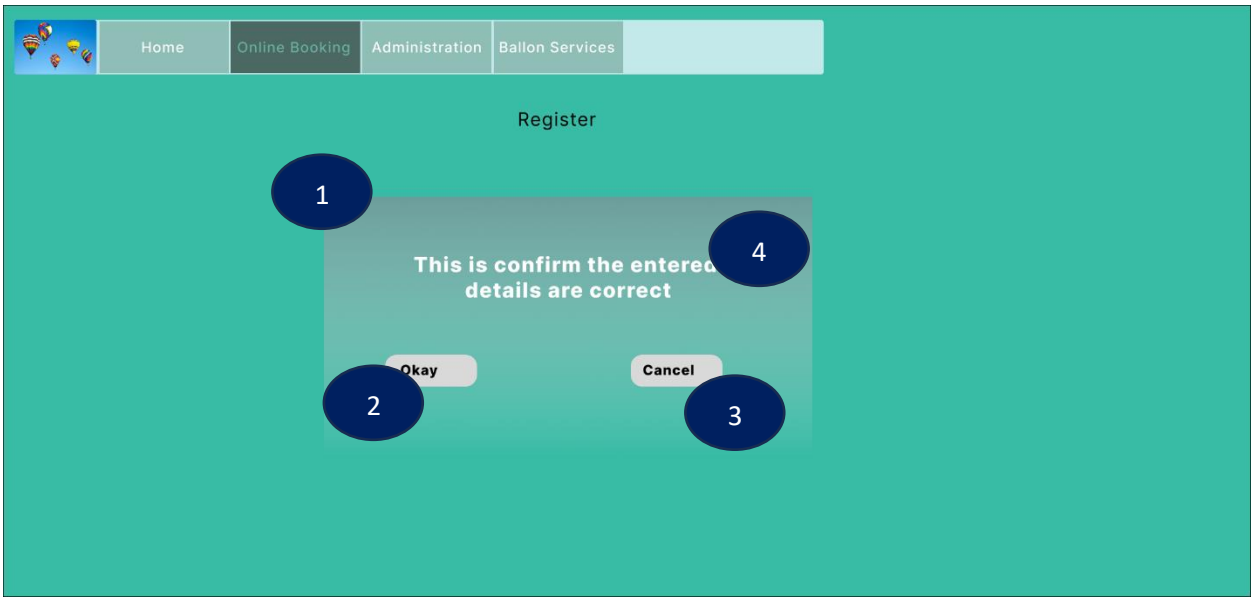
2.Input Design

2.1 Introduction

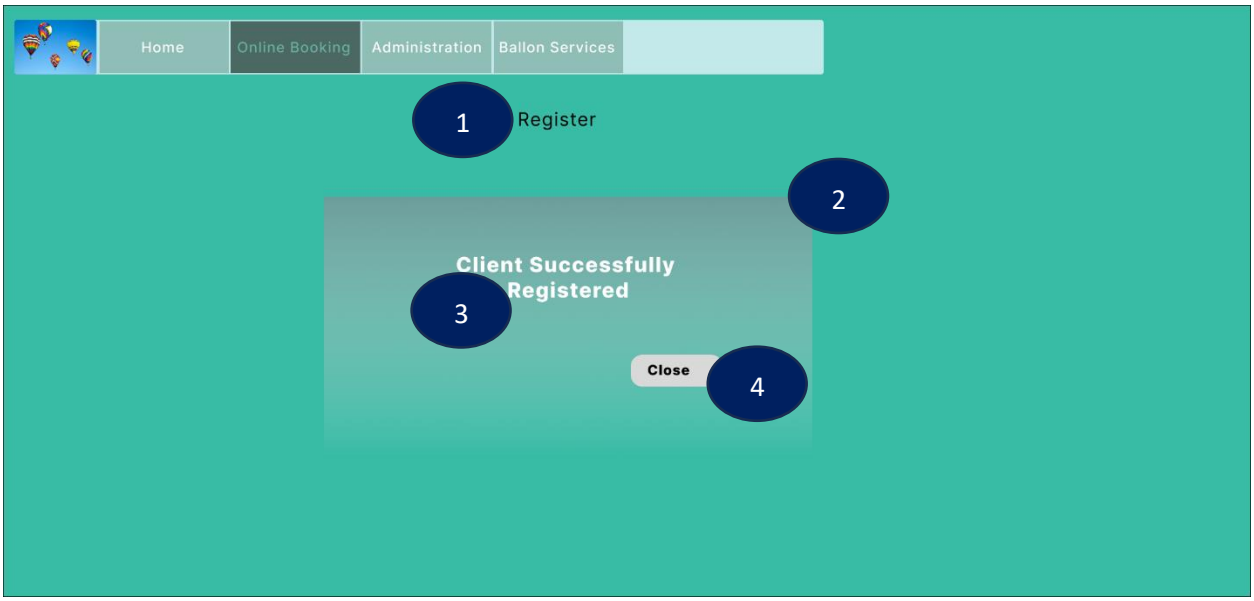
Screen specification offers a layout for visual and interactive elements of digital products, influencing user satisfaction and engagement. They remove misunderstandings between different stakeholders, such as designers and engineers they provided effective communication. They help identify critical issues before they can impact the product.



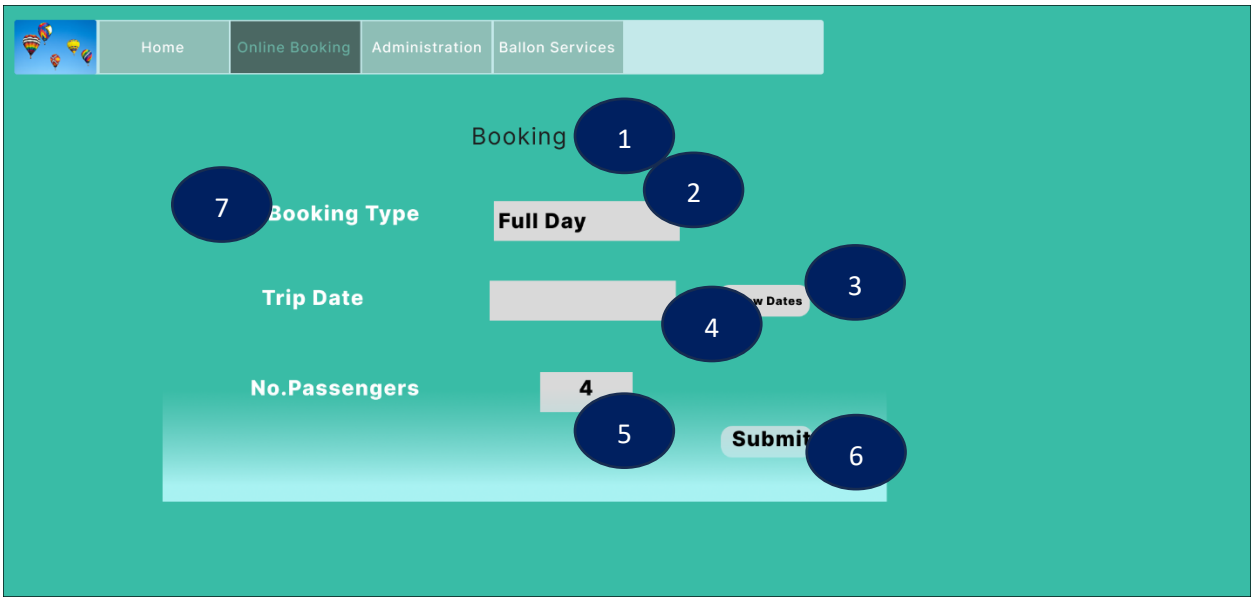
#	Item Name	Item Description
1	Register Heading	Heading serves as screen description
2	Name label	Makes it easier for user to enter correct information in relevant textboxes
3	Email textbox	Saves client email address to customer table
4	Phone Number	Captures client phone number
5	Home address	Captures client home address
6	Next button	Takes customer to detail confirmation screen



#	Item Name	Item Description
1	Confirm Details Message box	Holds confirmation message and two buttons
2	Okay button	Clicking button means client confirms detail entered is correct
3	Cancel Button	Will redirect client to register screen with all textboxes re-enable for correction purposes
4	Confirmation label	Serves as notification to user asking them to confirm the detail entered before proceeding

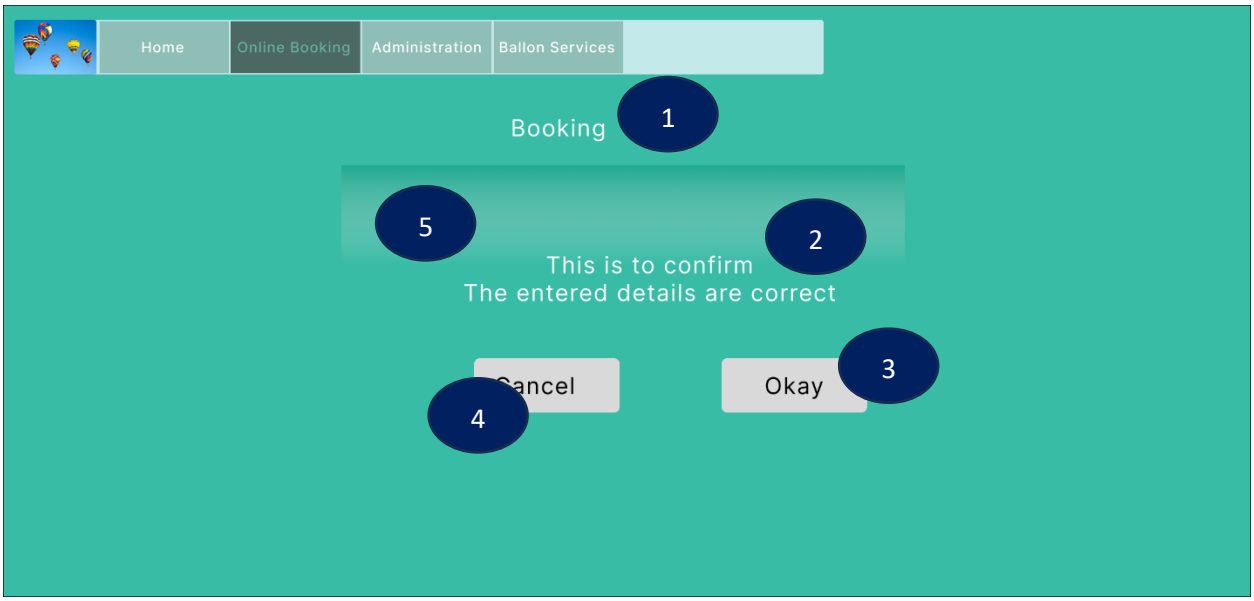


#	Item Name	Item Description
1	Register Heading	Heading serves as screen description
2	Message box	Holds successful registration message and one buttons
3	Registration label	Serves as notification to user to inform them that they have been registered on the African Adventures System
4	Close Button	Will redirect client to home screen



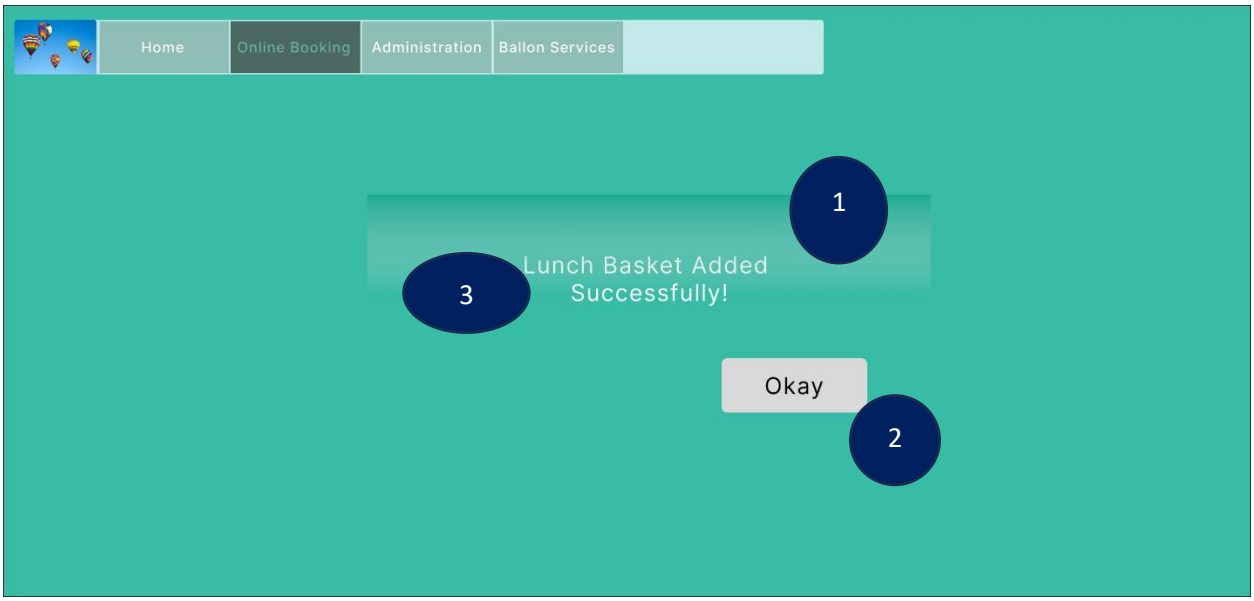
#	Item Name	Item Description
1	Booking Label	Serves as heading for the screen
2	Booking dropdown	Allows client to select either half/full day trip
3	View Dates button	Allows Client to view available flights
4	Trip date Textbox	Auto fills after client has viewed and selected an available trip date
5	No. Passenger numeric up down	Allows user to enter whole number of people to attend trip
6	Submit Button	Allows client to submit capture data to book a trip
7	Booking type label	Labels booking type textbox also making it easier for client to enter correct information

#	Item Name	Item Description
1	Available Flight Label	Screen description
2	Available flights table	Shows array of remaining open time slots for hot air balloon trip
3	Time slots label	Part oof table and show available times
4	Back button	Redirects client to Booking page

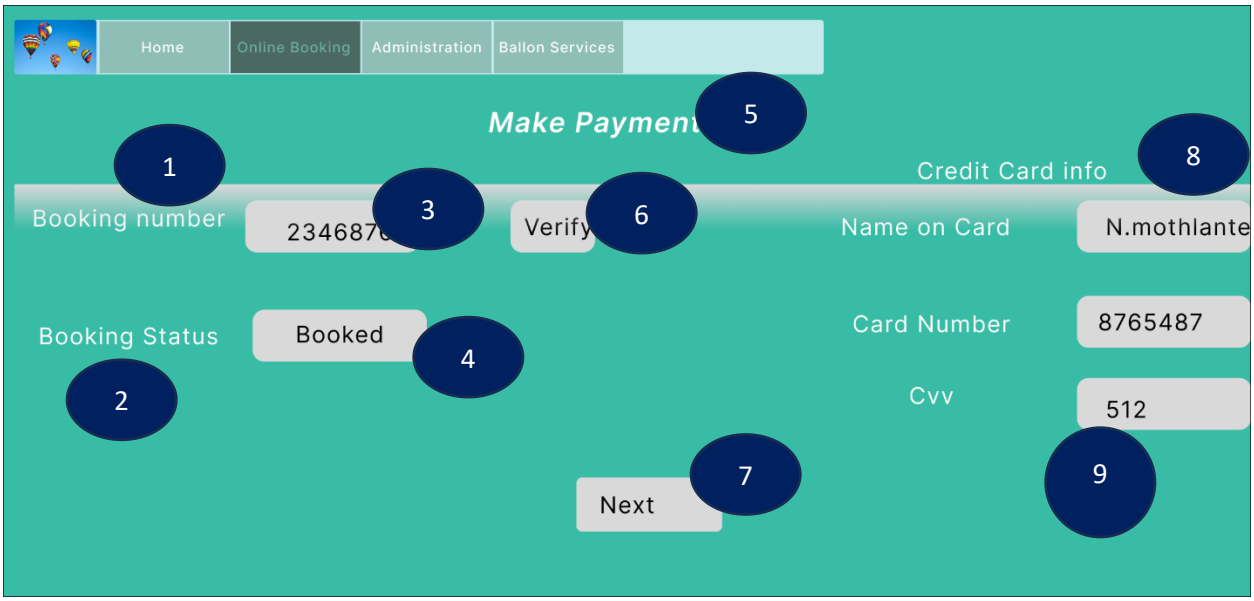


#	Item Name	Item Description
1	Booking label	Header for screen
2	Confirm info Label	Message prompting client to confirm entered trip detail
3	Okay button	Okay button creates booking in system redirects user to home screen
4	Cancel button	Redirects user to bookings page, where corrections to entered information can be made
5	Message box	Holds confirmation message, and two buttons

#	Item Name	Item Description
1	Lunch Basket Label	Serves as heading for the screen
2	Menu Item	Picture to add to aesthetic of screen
3	Add lunch basket label	Screen subheading
4	Check box	Check box for client to add lunch basket

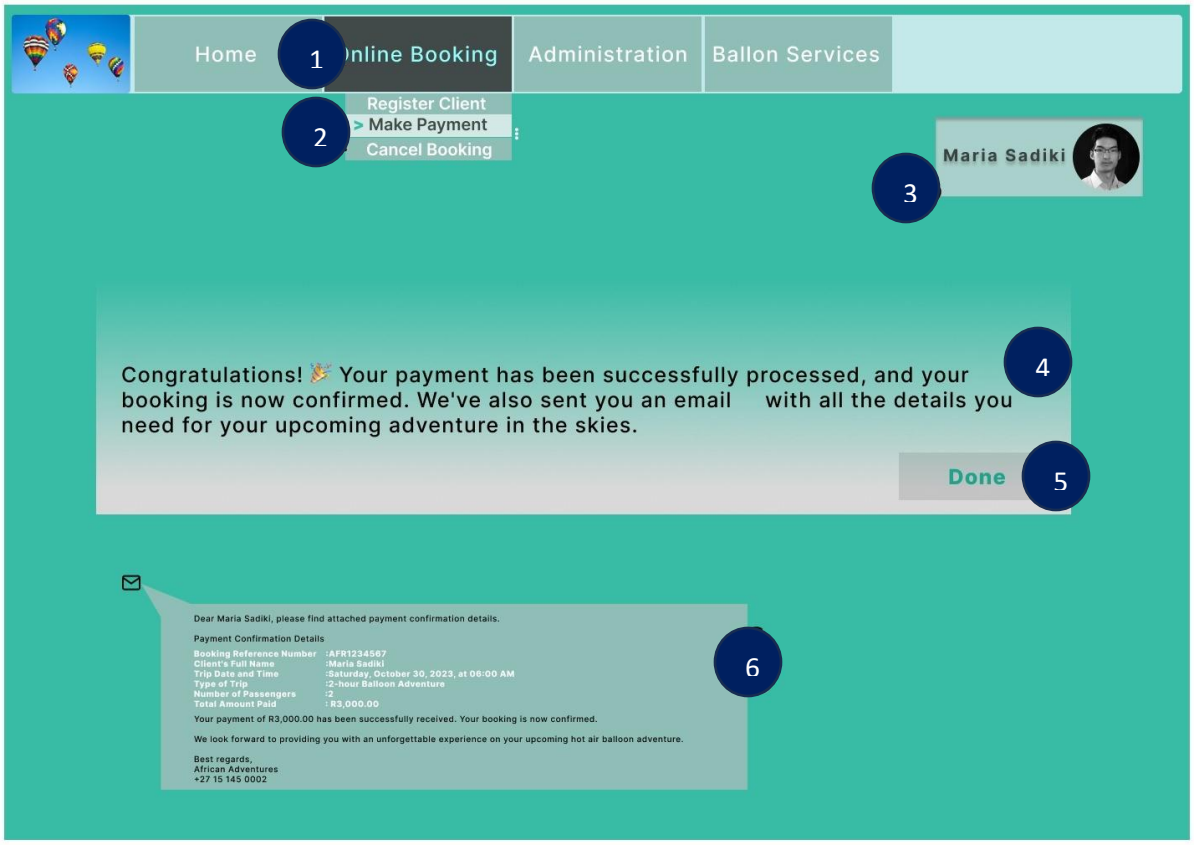


#	Item Name	Item Description
1	Message box	Holds label and 1 button
2	Okay button	Redirects user to bookings page
3	Lunch basket Label	Notification message of successful addition of lunch basket to trip



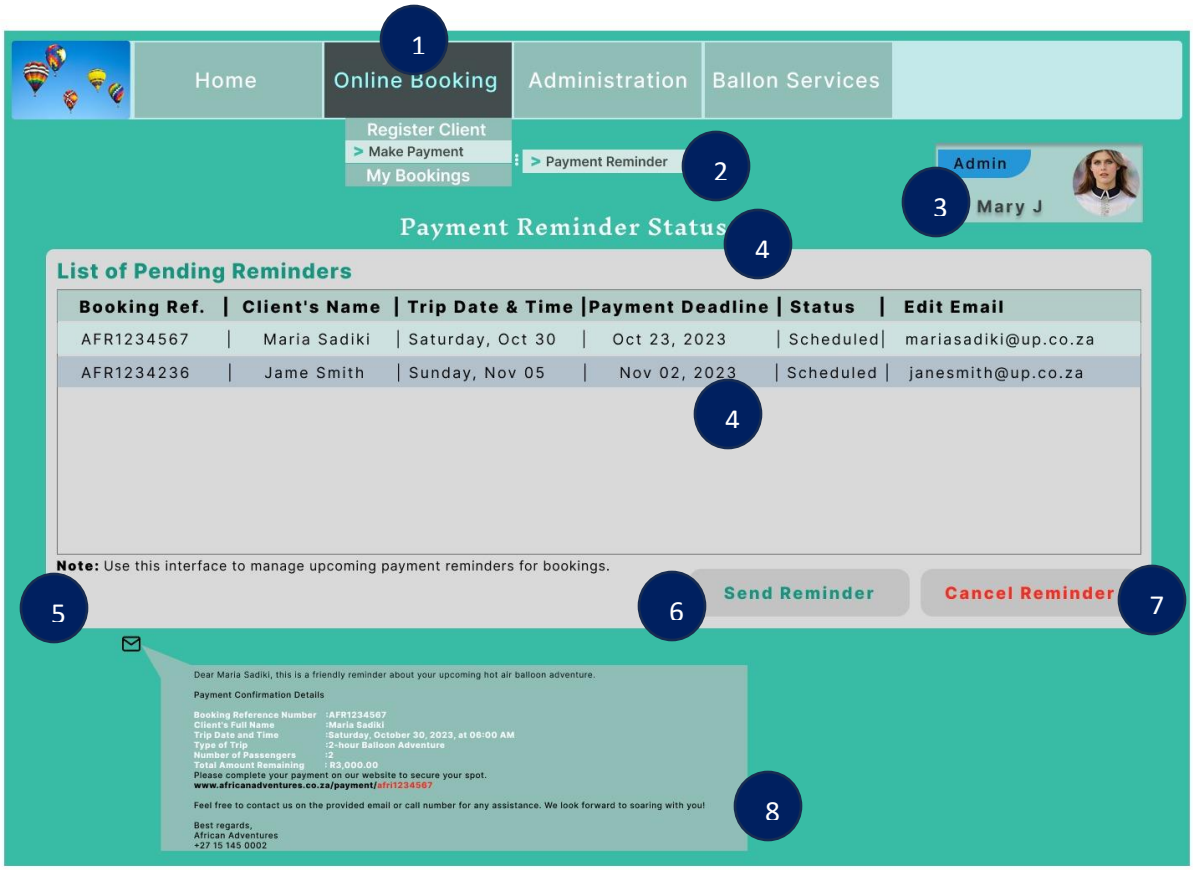
#	Item Name	Item Description
1	Booking number label	Labels booking number textbox ensuring user inputs correct information
2	Booking status label	Label booking status label which will autofill on retrieval
3	Booking number textbox	Captures booking reference
4	Booking status textbox	Autofill's with booking status
5	Make payment heading	Heading for the screen
6	Verify Button	Retrieves booking status from database
7	Next button	Processes payment
8	Credit card info label	Labels credit card information section
9	Cvv numeric up down	Numeric up down for cvv number entries

1.6Send payment confirmation screen



Item number	Payment reminder notification	This notification is sent to the client warning them that they haven't fully paid their booking amount and that if they fail to finalize the payment, they will forfeit their 50% deposit amount.
1	Online booking menu (Function)	This menu is part of the African adventures navigation bar it shows the user which tab they are currently on.
2	Make payment task	It is used to show the task that the use is about to perform in this case it is going to send payment notification.
3	Logged in user profile	It shows the user currently performing the task
4	Message box	It is used to inform the user that the task they were doing is completed
5	Done button	When clicked the system will go back to "Home" screen
6	Email notification	It's the payment confirmation notification which was sent to the client right after making their payment.

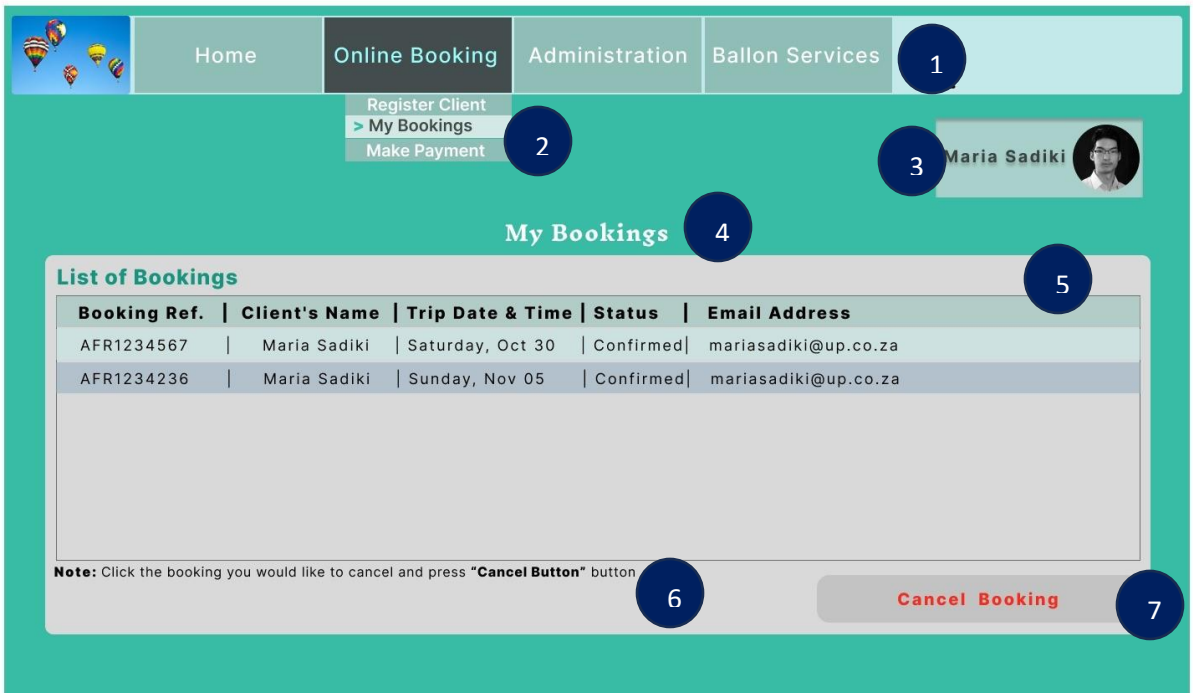
91.7 Send payment reminder screen.

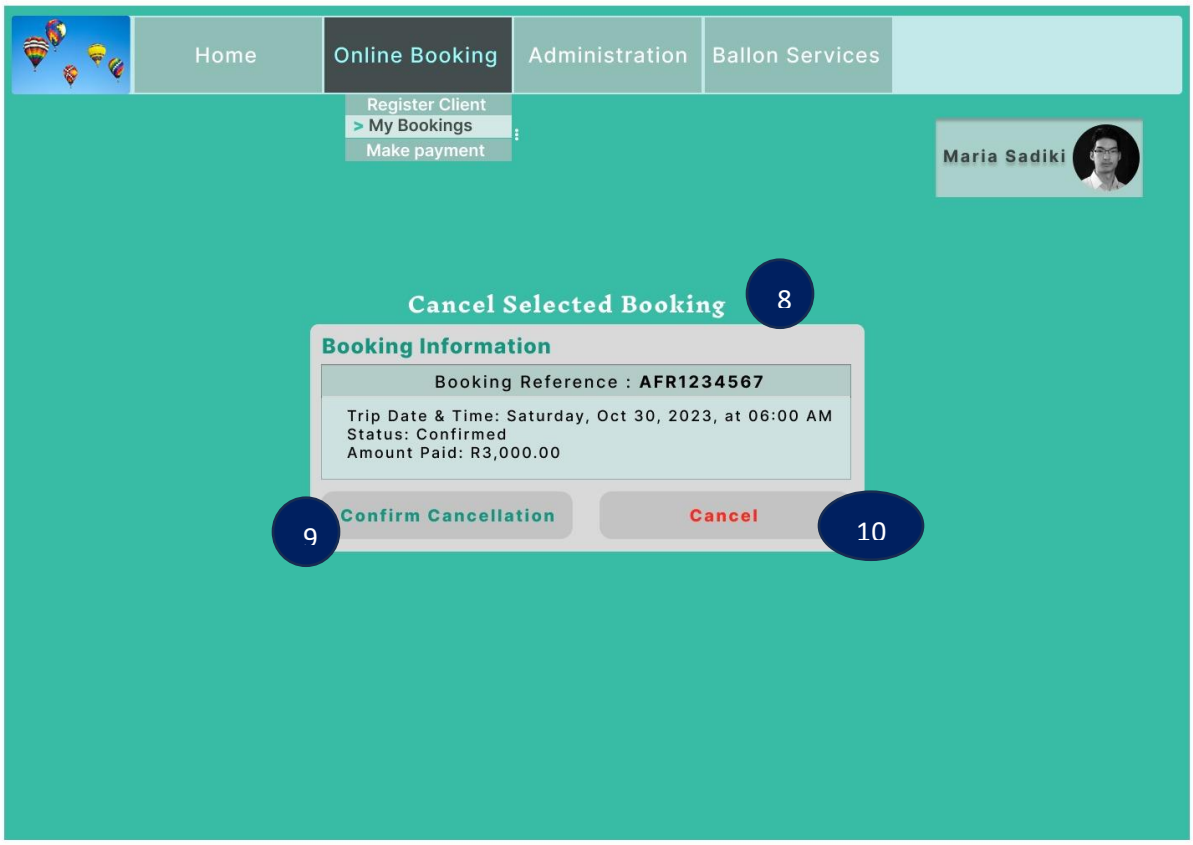


Item number	Item name	Item description
1	Navigation bar	Online booking menu item which when clicked, it takes the user to all tasks under the online booking function
2	Option menu	It's the option menu from the make payment dropdown which takes the user to the scheduled payment reminder's page.
3	Logged in user profile	It shows the user currently performing the task
4	Payment reminder status label	Used to indicate the name and function of the screen containing a list of pending reminders to be sent to clients who did not pay the full amount for the trip.
5	Tooltip note	It guides/inform the user about the contents or function of the screen or the action they are about to perform
6	Send reminder button	When clicked it sends the scheduled payment reminders to the clients.

7	Cancel reminder button	It is used if the client decided to pay the money after the reminder was already scheduled.
8	Payment reminder notification	This notification is sent to the client warning them that they haven't fully paid their booking amount and that if they fail to finalize the payment, they will forfeit their 50% deposit amount.

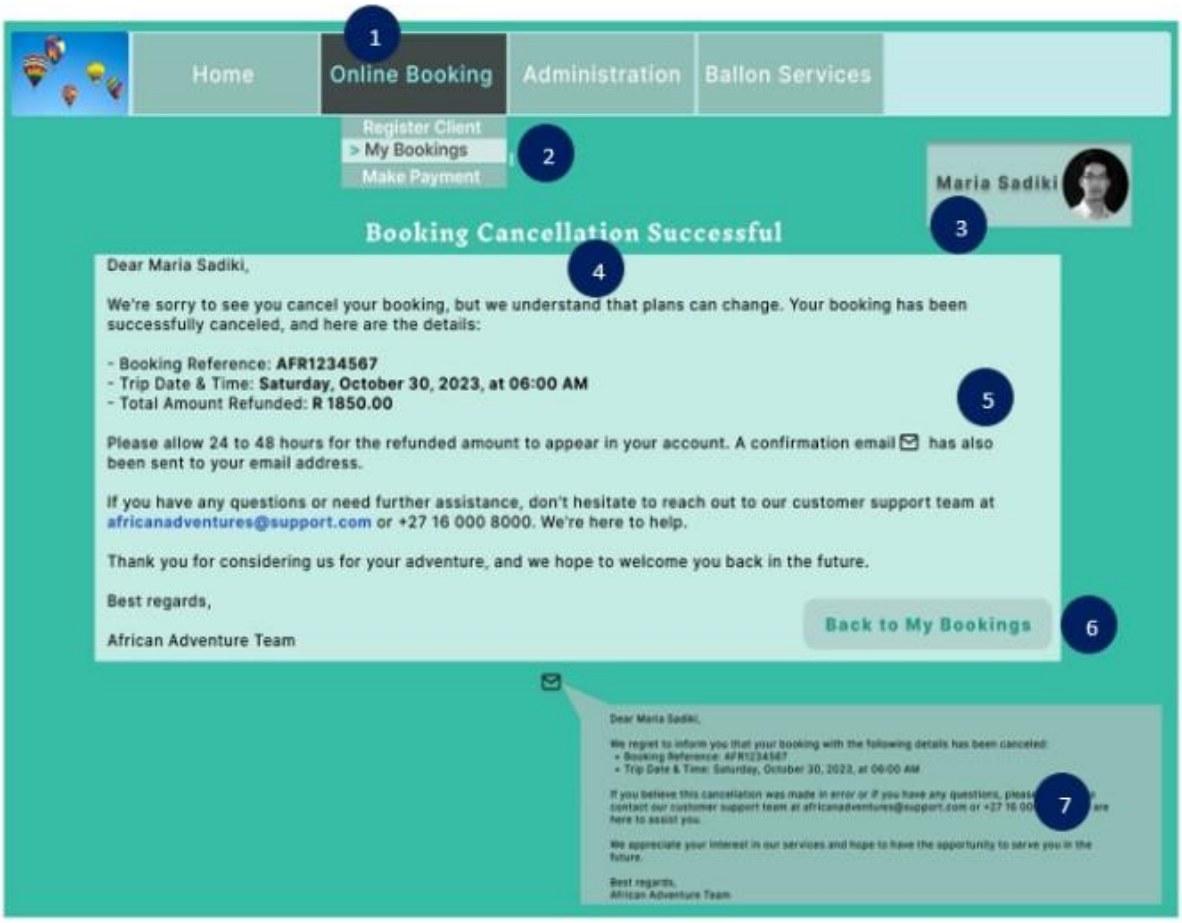
1.8Cancel booking screen




Home
Online Booking
Administration
Ballon Services
Register Client
> My Bookings
Make payment
Maria Sadiki
Cancel Selected Booking
8
Booking Information
Booking Reference : AFR1234567
Trip Date & Time: Saturday, Oct 30, 2023, at 06:00 AM
Status: Confirmed
Amount Paid: R3,000.00
Confirm Cancellation
Cancel
9
10

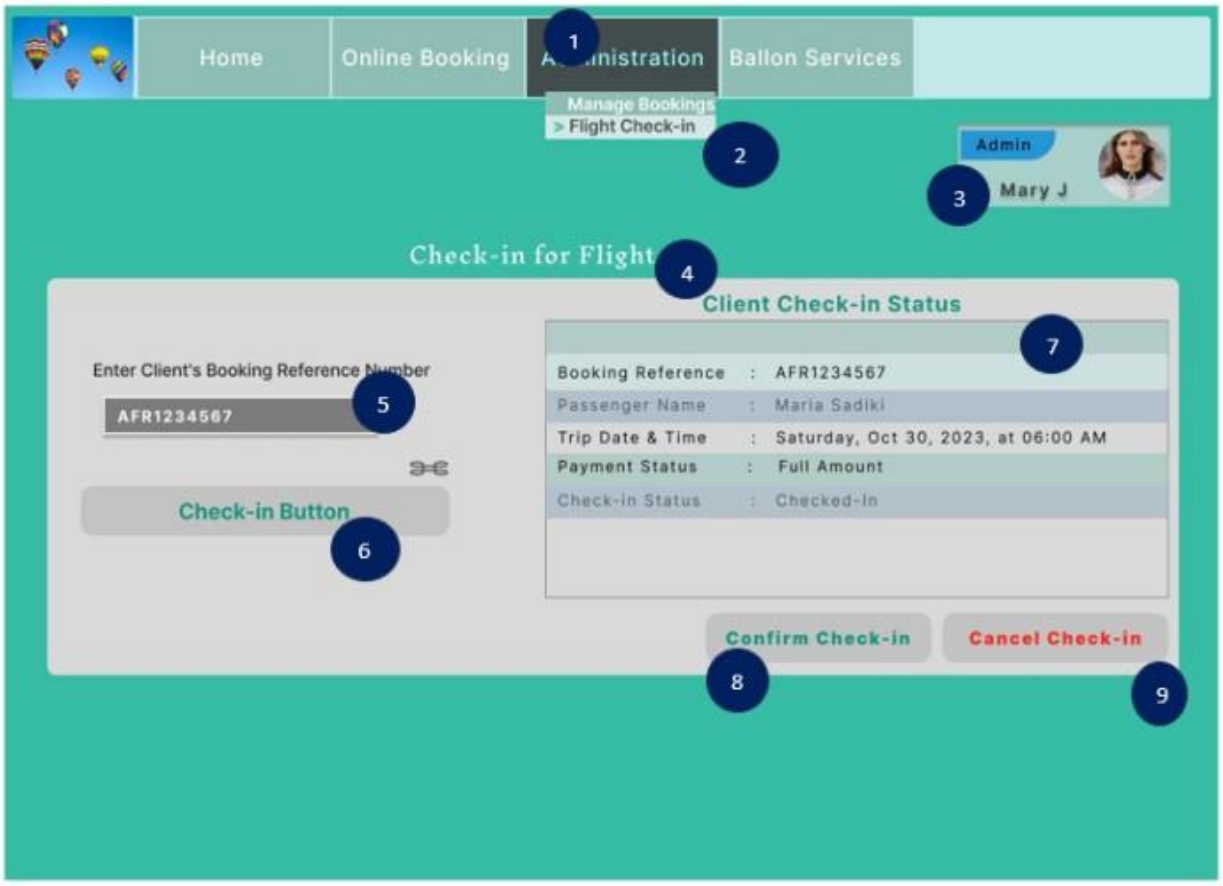
Item number	Item name	Item description
1	Navigation bar	Online booking menu item which when clicked, it takes the user to all tasks under the online booking function in this case cancel booking
2	Option menu	It's the option menu from my bookings dropdown which takes the user to their active bookings
3	User profile icon	It shows the user currently logged in or interacting with the system in this case it's the client
4	My bookings label	Used to indicate the name and function of the screen
5	Rich textbox	It's used to show all the client bookings from which they will select the booking they would like to cancel
6	Tooltip note	It guides/inform the user about the contents or function of the screen or the action they are about to perform
7	Cancel booking button	When clicked it brings a modal pop-up (8)
8	Cancel booking modal pop-up	It's triggered by clicking cancel booking button, it contains summarised booking information of the client booking they are wishing to cancel
9	Confirm cancelation button	When clicked it cancels the selected booking completely from the database
10	Cancel button	It takes the client to the "my bookings" screen when clicked

2.1Check-in for flight screen



Item number	Item name	Item description
1	Online Booking menu	Online booking menu which when clicked takes the user to all tasks under the online booking function
2	My bookings menu item (send cancellation confirmation)	It's the option item from my bookings dropdown which takes the user to their bookings for generation of the cancellation email.
3	User profile icon	It shows the user currently logged in or interacting with the system in this case it's the client
4	Booking cancellation successful label	Used to indicate the name and function of the screen
5	Modal popup	It displays a copy of the cancellation confirmation email
6	Back to my booking button	It takes the user back to my bookings main page when clicked
7	Confirmation email	Snippet of the email sent to the client after successful cancelation of booking

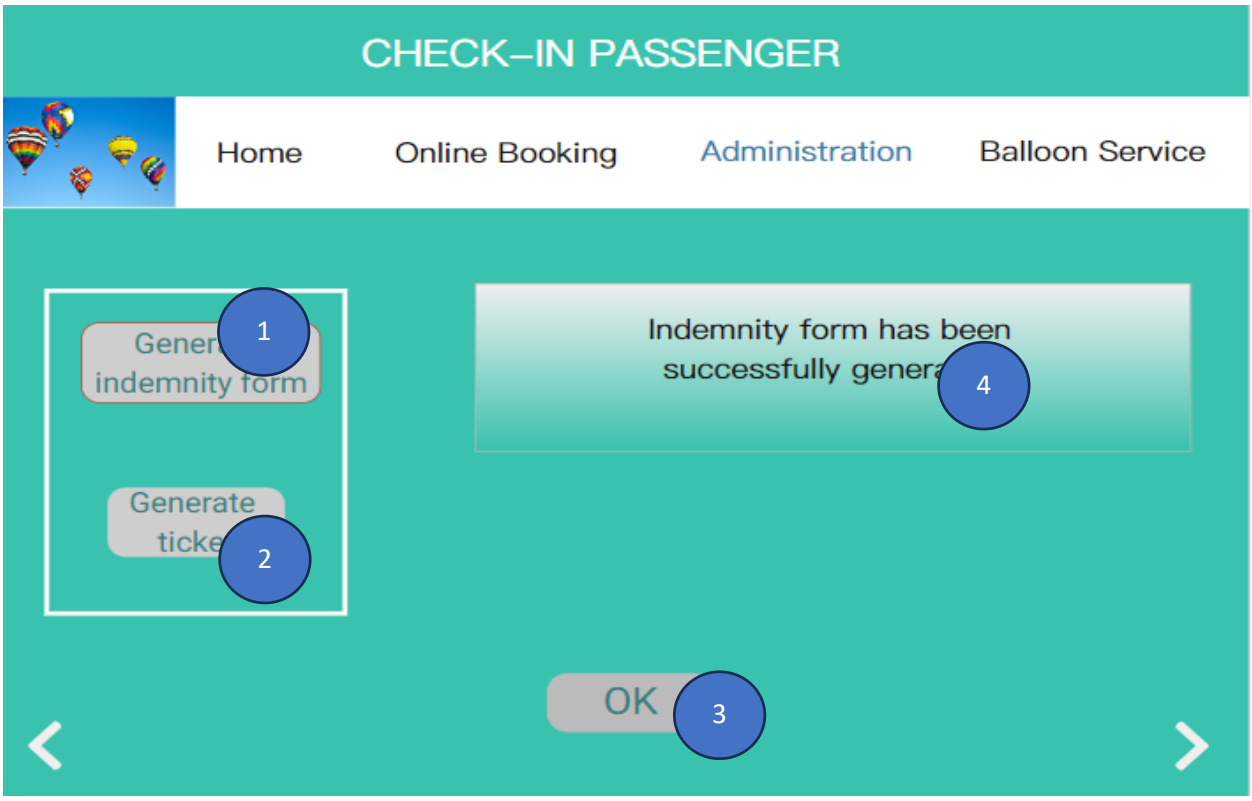
2.1Check-in for flight screen



Item number	Item name	Item description
1	Administration menu	Administration menu which when clicked takes the user to all tasks under the administration booking function
2	Flight check-in menu item	When clicked it takes the user to the check-in main page where the admin can check-in the passenger successfully
3	Admin profile icon	It shows that the admin is logged in and she is the one performing the check-in task
4	Check-in for flight label	It describes the function of the screen which is check-in passenger for flight
5	Reference number textbox	It is used to enter the input (passenger reference number for booking) for processing
6	Check-in button	It is used to processes the entered input when clicked

7	Check-in status modal popup	It is a result from the clicked check-in button, it contains the client booking information and the check-in status
8	Confirm check-in button	When clicked it shows the check in confirmation message
9	Cancel check-in button	When clicked it takes the user to the main check-in page

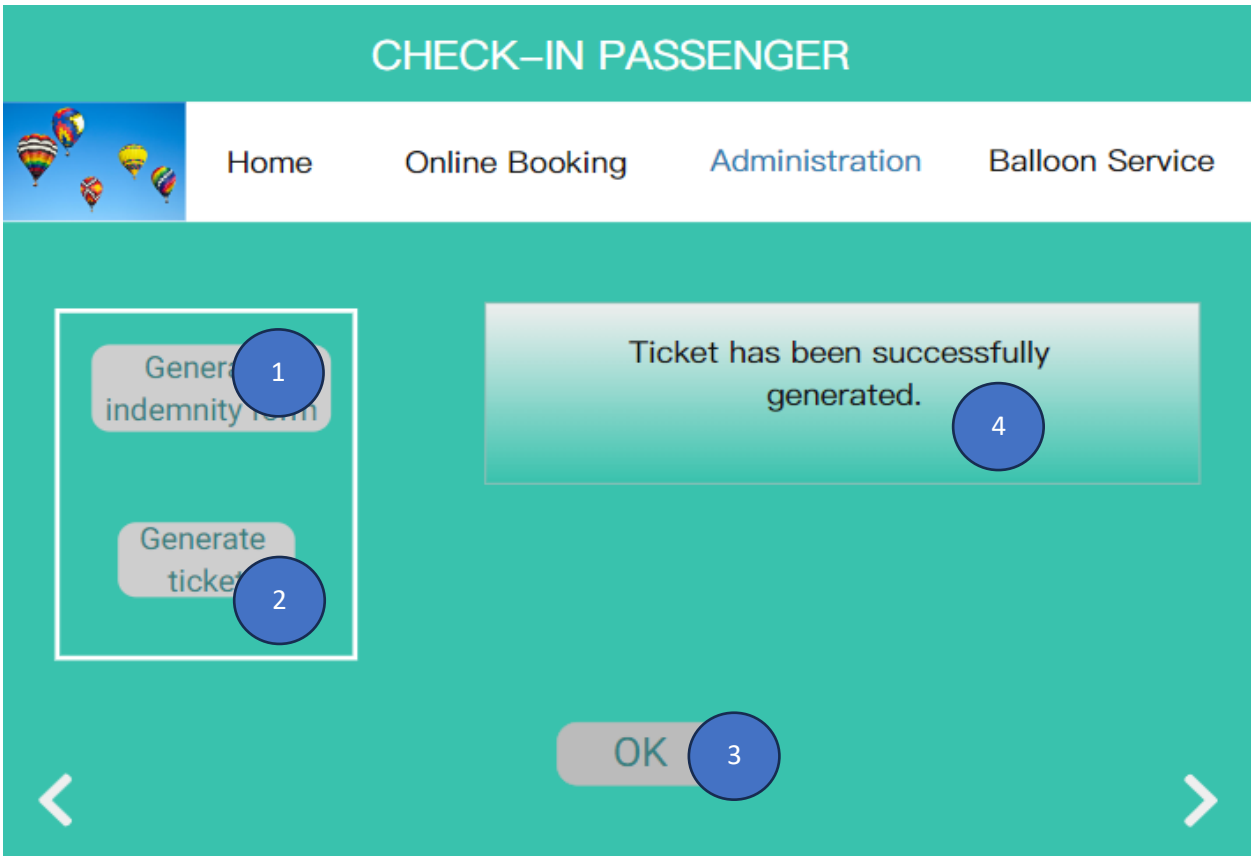
2.2 Generate indemnity form



Number	Control	Explanation
1	Generate indemnity form button	For generating the indemnity form
2	Generate ticket button	For generating ticket
3	OK button	This button directs the user back to the main check in screen.
4	Messagebox	Whenever the generate indemnity form or generate ticket a message box must be displayed to inform the user that they have successfully

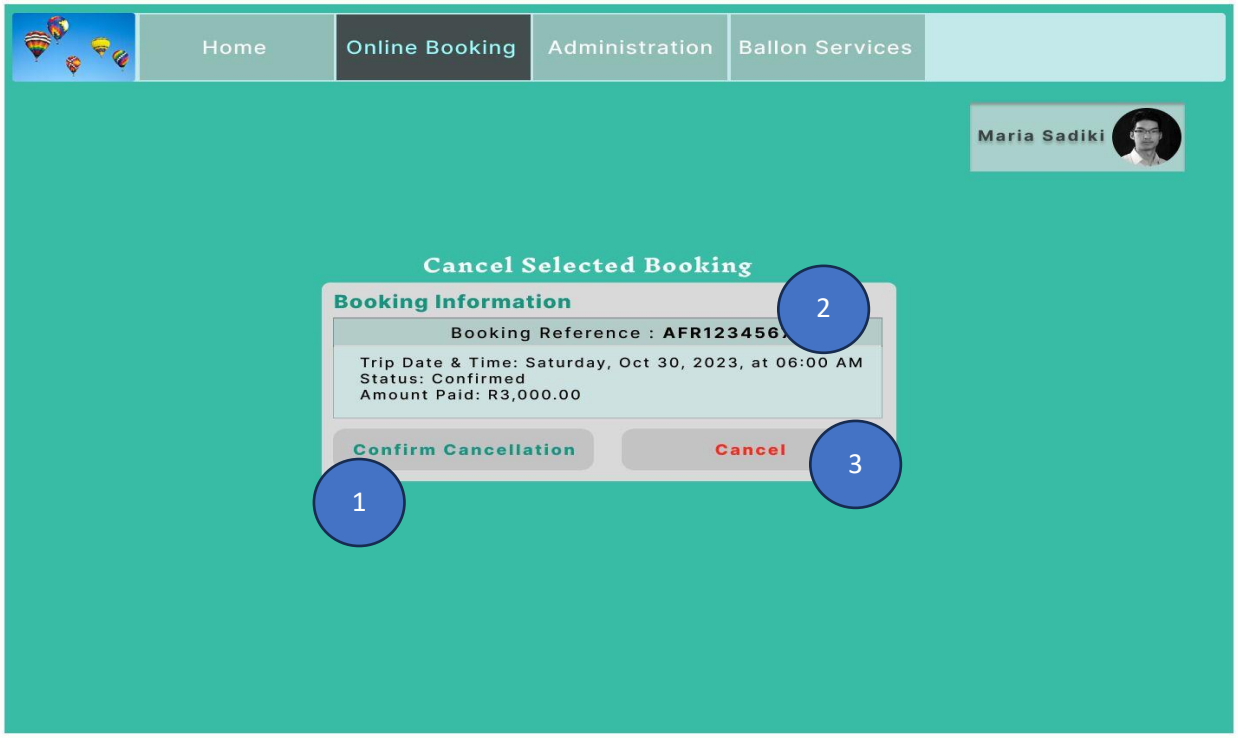
		generated the indemnity form or ticket.

2.3 Generate ticket

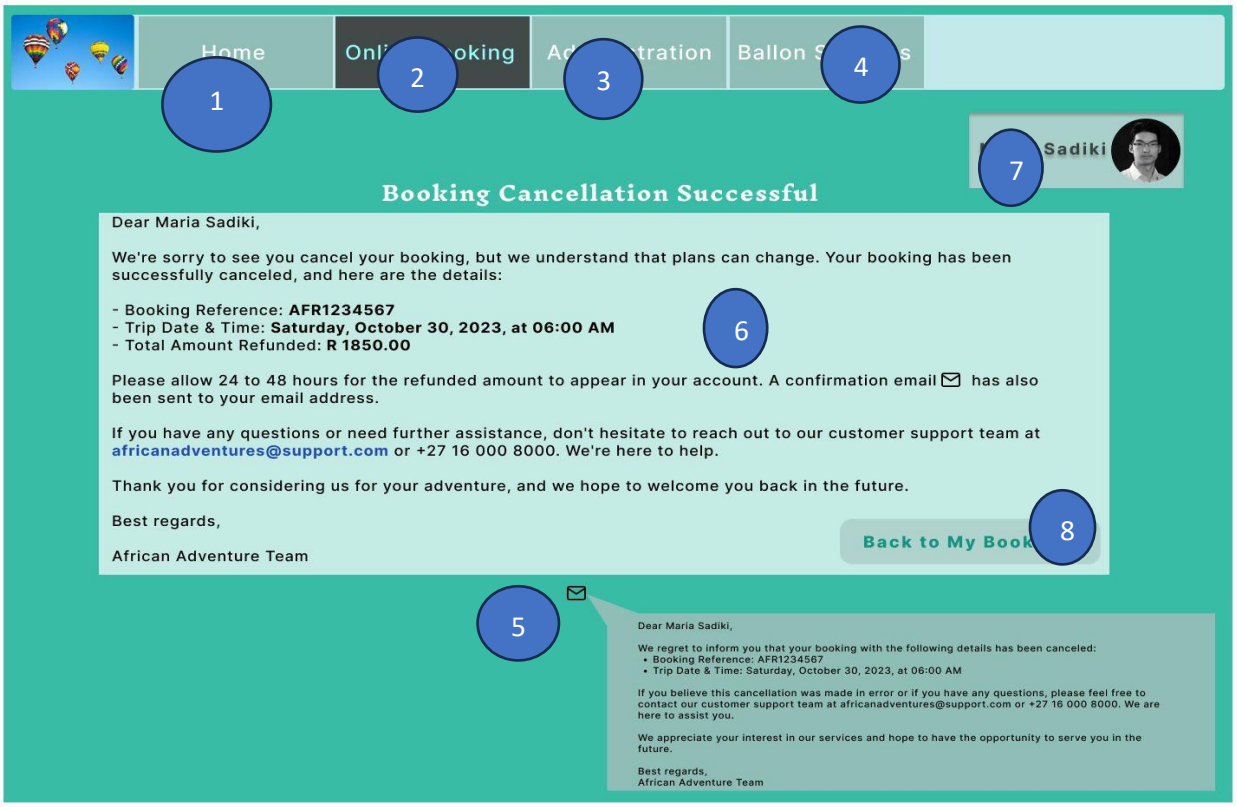


Number	Control	Explanation
1	Generate indemnity form button	For generating the indemnity form
2	Generate ticket button	For generating ticket
3	OK button	This button directs the user back to the main check in screen.
4	Messagebox	Whenever the generate indemnity form or generate ticket a message box must be displayed to inform the user that they have successfully generated the indemnity form or ticket.

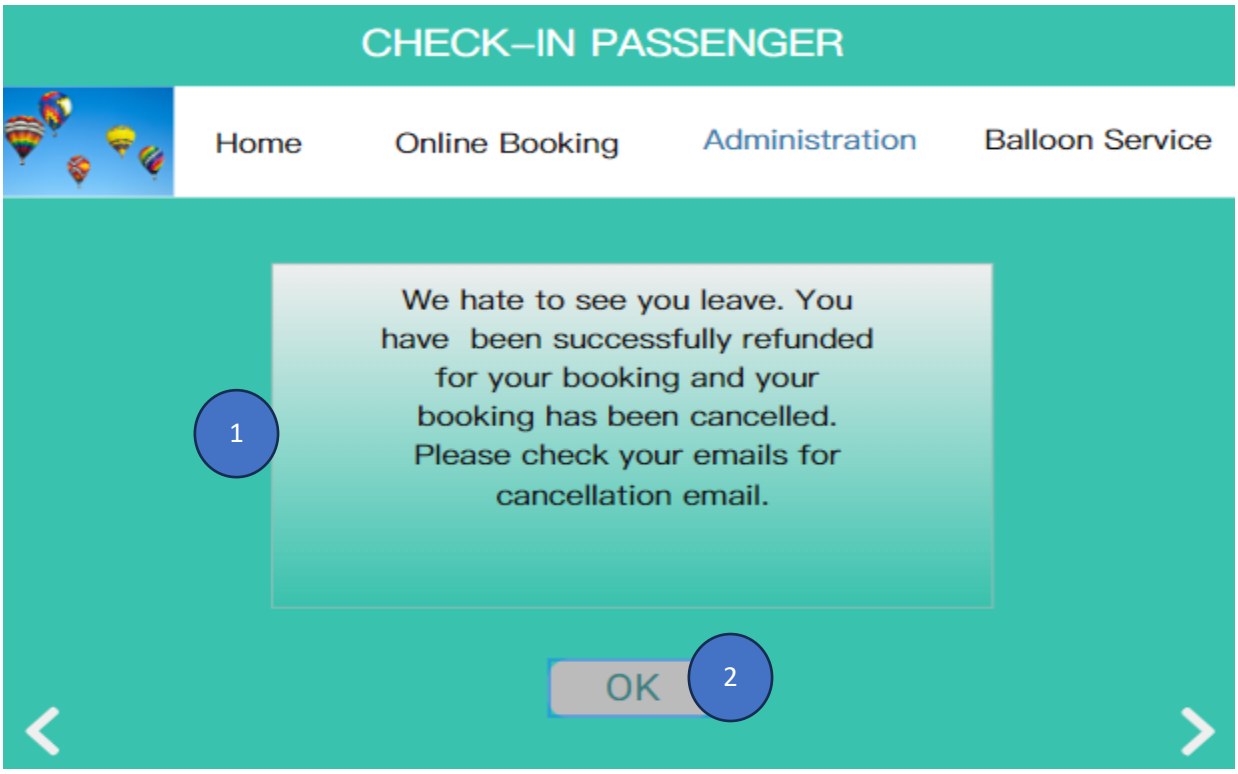
Refund Booking



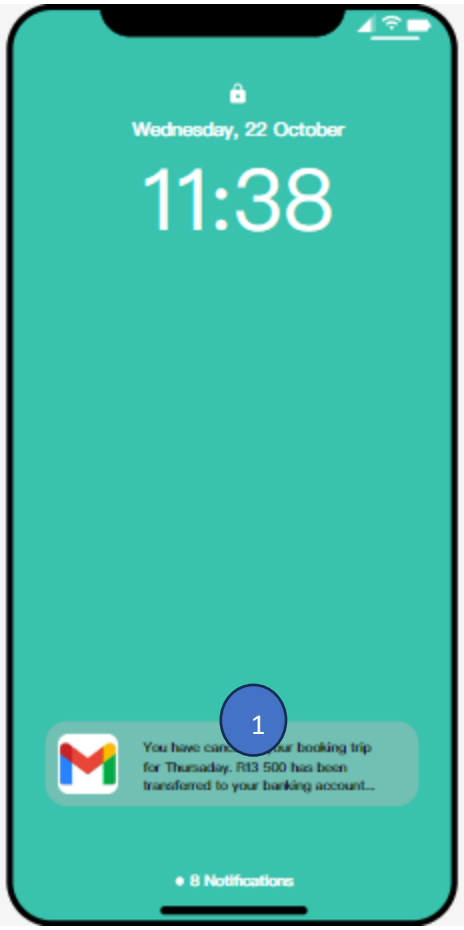
Number	Control	Explanation
1	Confirm cancellation button	It is a button for confirming the cancellation of a booking
2	Selected booking table	Display the retrieved booking details to be cancelled.
3	Cancel button	This button is for cancelling the action of cancelling a booking



No	Controls	Explanation
1	Home button	Takes the user to the home page
2	Online booking button	Directs the user to the online booking page.
3	Administration button	Directs the user to the administration main screen.
4	Balloon Services button	Directs the user to the balloon services page.
5	Email button	The button displays a small message box notifying the user about their booking cancellation information via email.
6	Message box	The message box contains a message that notifies the user that they have successfully cancelled their booking.
7	Signed in button	This button shows that the user is logged onto the system.
8	Back to my booking list	Directs the user to the main booking screen

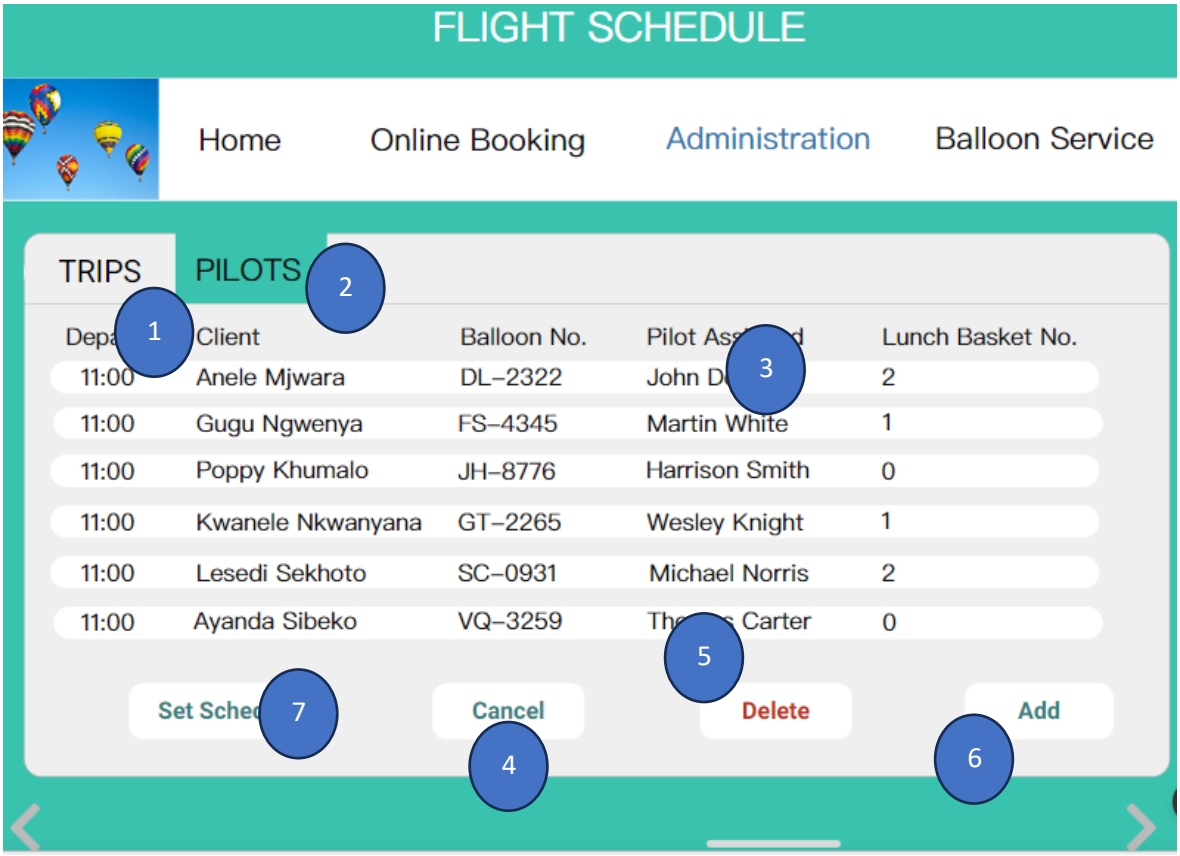


Number	Control	Explanation
1	OK button	Directs the user to the main check in screen
2	Messagebox	For displaying a message notifying the user to check their email for the refund.

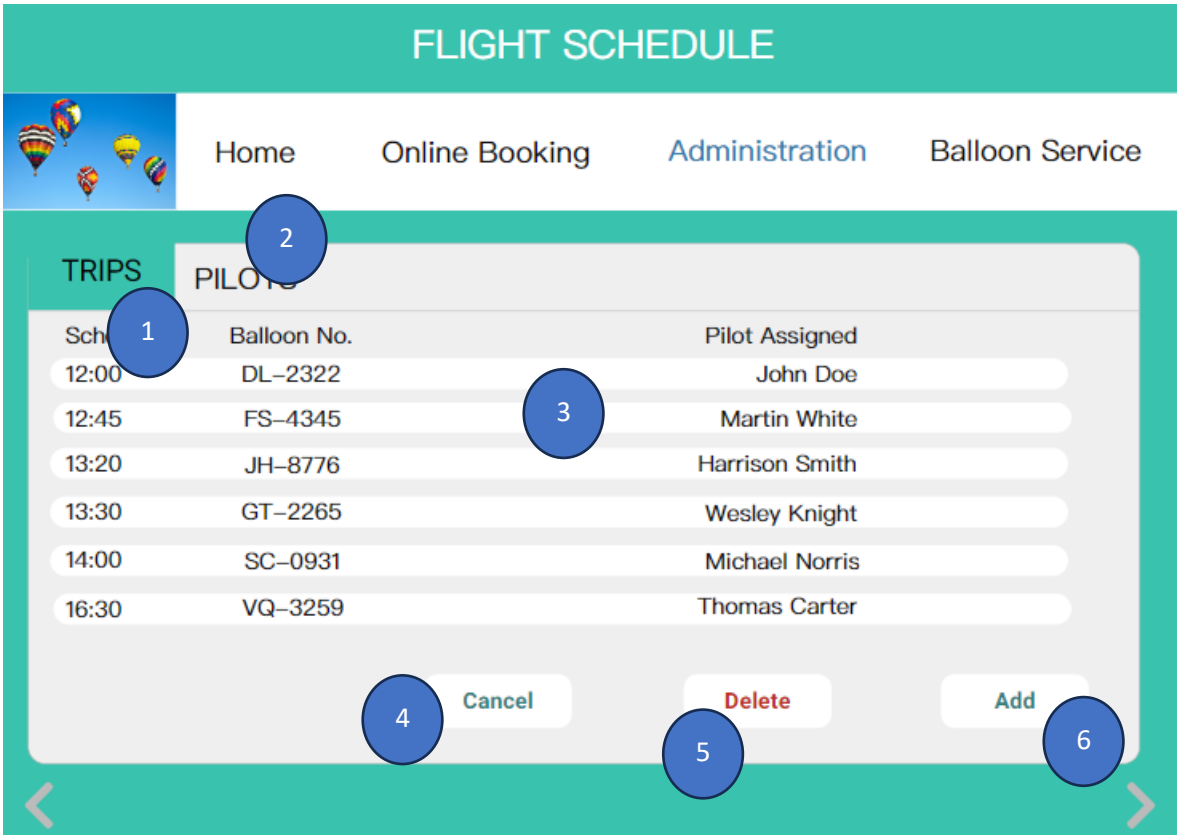


No	Item Name	Item Description
1	Booking cancellation email	The user receives an email on their phone notifying them about the successful refund for their cancelled booking

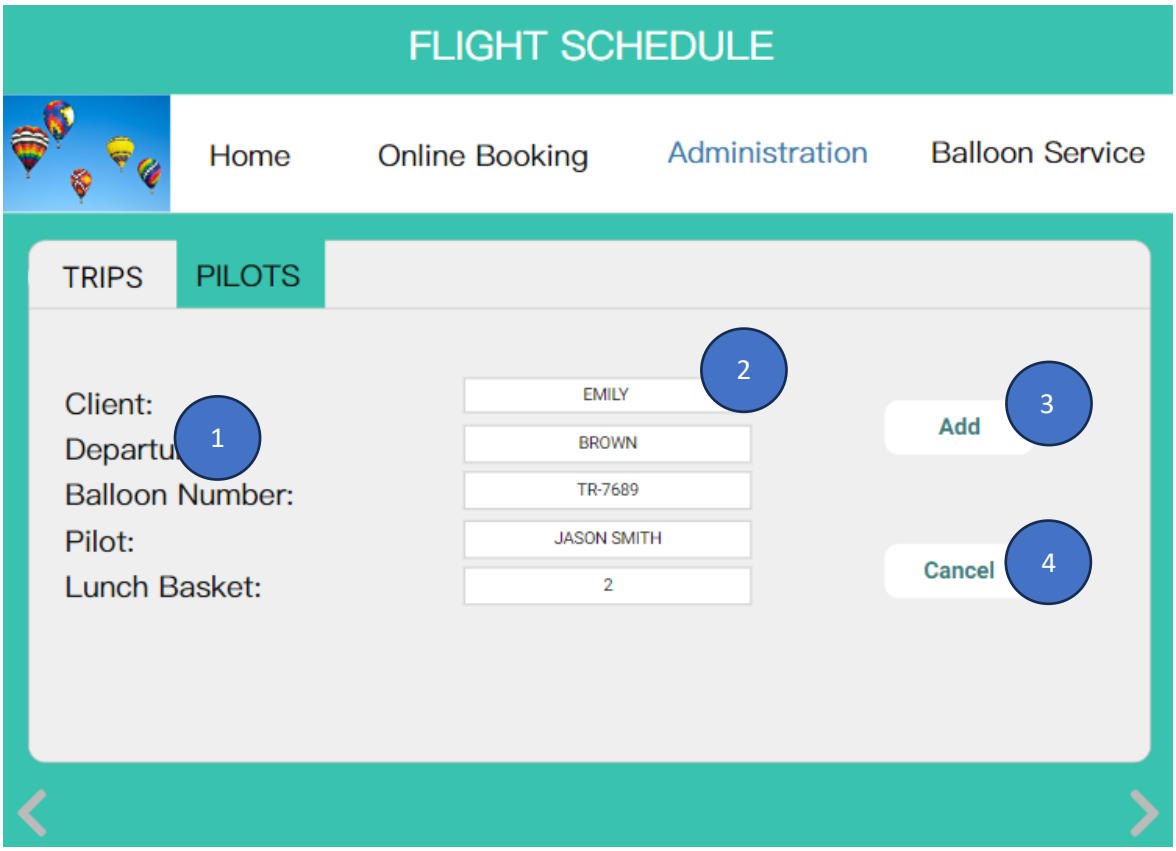
2.5 Set up flight schedule



Number	Item Name	Item Description
1	Trips tab	Displays the flight schedule with trips booked.
2	Pilots tabs	Displays flight schedule with pilots assigned to trips.
3	Pilot table	Displays the flight schedule with clients, balloons for the trips clients booked for and the pilots assigned to trips.
4	Cancel button	If the user changes their mind and do not want to proceed with setting up a flight schedule.
5	Delete button	It is for deleting a specific row from the trips booked.
6	Add button	For adding a new timeslot with client that booked for it.
7	Set up schedule button	A button for setting up a new flight schedule.

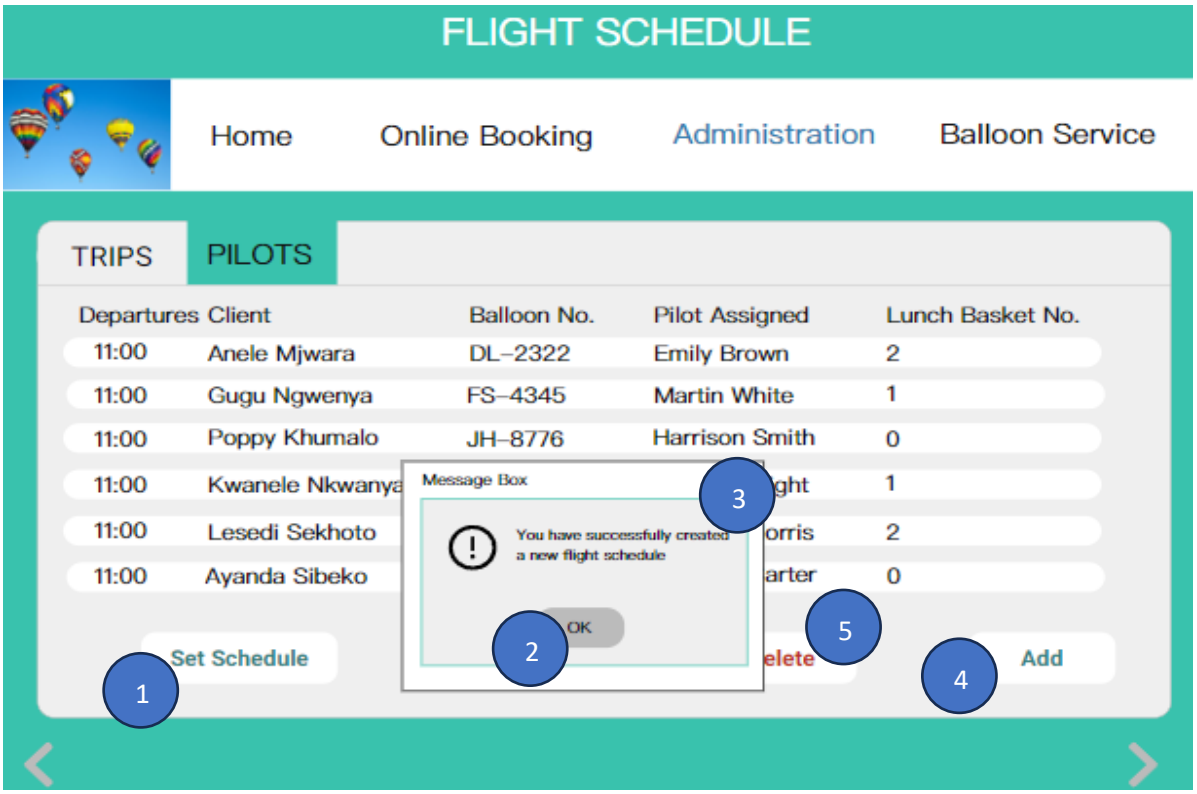


Number	Item Name	Item Description
1	Trips tab	Displays the flight schedule with trips booked.
2	Pilots tabs	Displays flight schedule with pilots assigned to trips.
3	Pilot table	Displays the flight schedule with clients, balloons for the trips clients booked for and the pilots assigned to trips.
4	Cancel button	If the user changes their mind and do not want to proceed with setting up a flight schedule.
5	Delete button	It is for deleting a specific row from the trips booked.
6	Add button	For adding a new timeslot with client that booked for it.



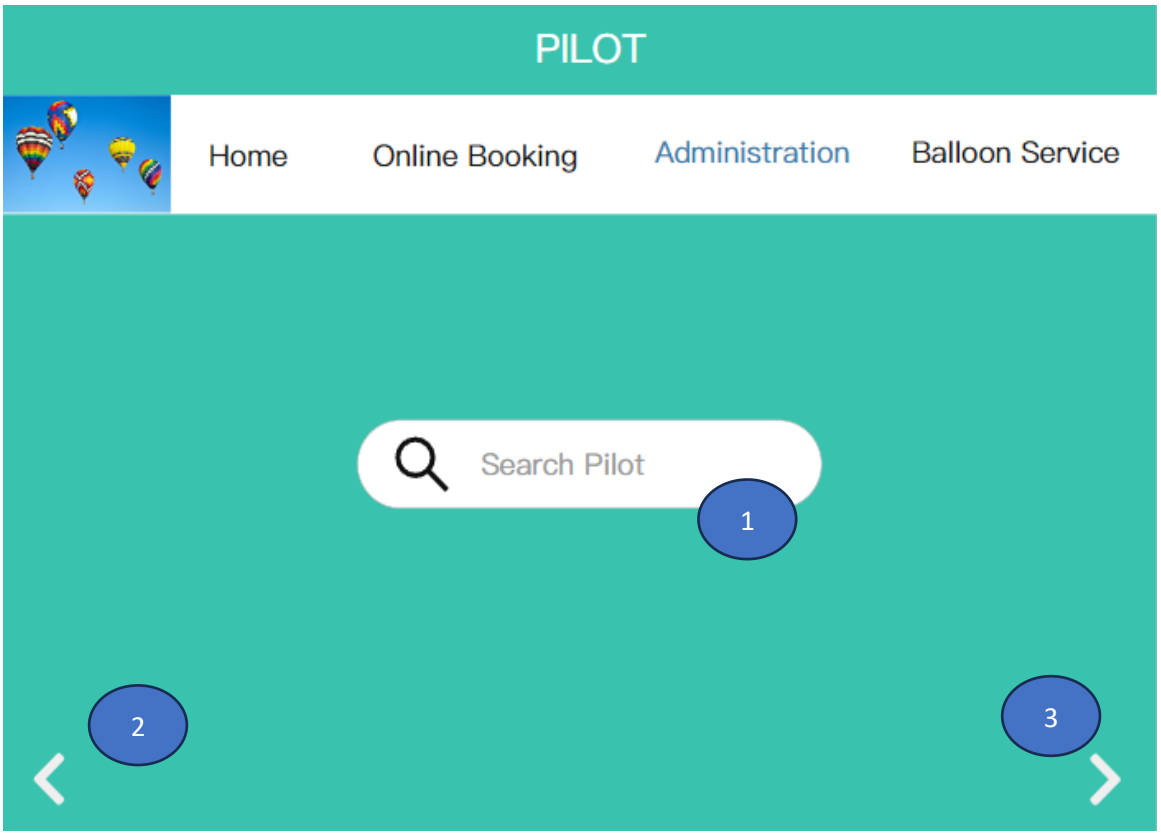
Number	Item Name	Item Description
1	Labels: • Client • Departure • Balloon number • Pilot • Lunch basket	Labels for the following information input: • Client • Departure • Balloon number • Pilot • Lunch basket
2	Textboxes for the following labels • Client • Departure • Balloon number • Pilot • Lunch basket	Textboxes for the following information • Client • Departure • Balloon number • Pilot Lunch basket
3	Add button	For adding a new flight timeslot onto the flight schedule.

4	Cancel button	For cancelling the setting up of a new flight schedule.

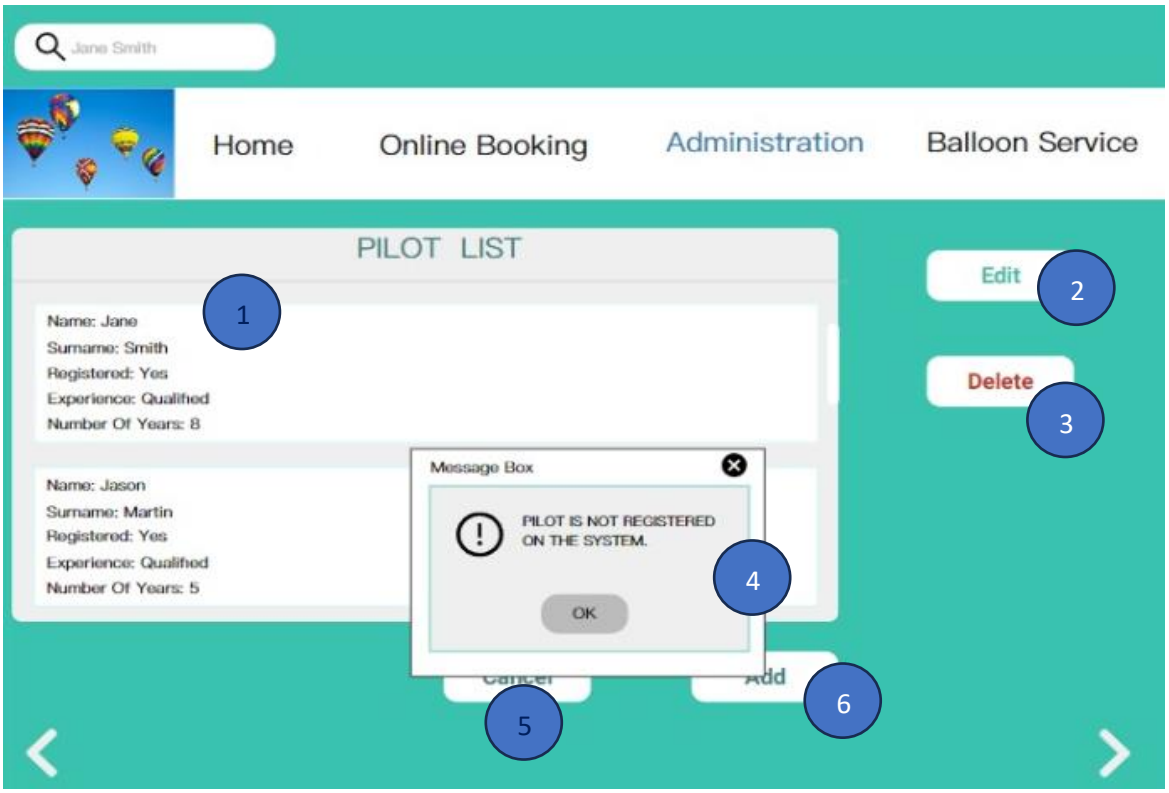


#	Item Name	Item Description
1	Set up flight schedule button	For setting up new flight schedule
2	OK button	For removing the message box and agreeing to the message.
3	Message box	It notifies the user that they have successfully created a new flight schedule.
4	Add button	For adding a pilot to the trip table
5	Delete button	For deleting a trip from the trip table.

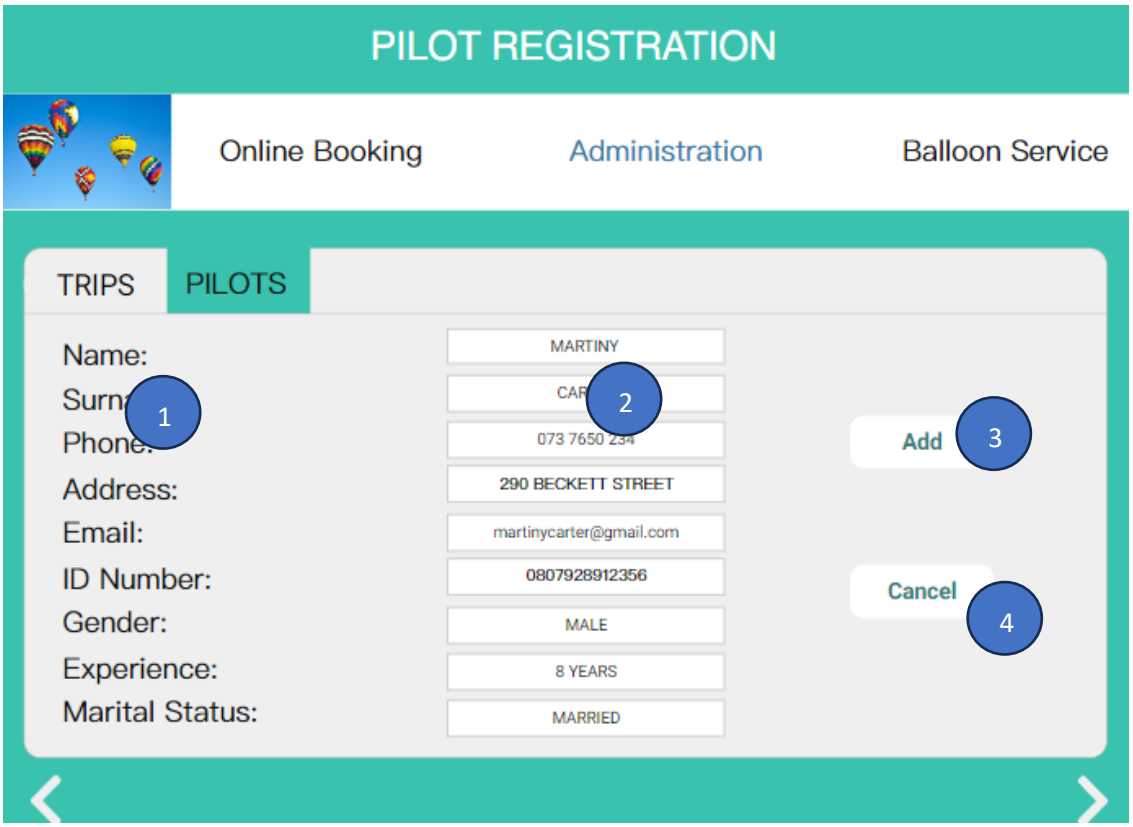
2.6 Register Pilot



#	Item Name	Item Description
1	Search pilot feature	A bar for searching a pilot.
2	Previous button	Takes the user back to the previous page
3	Next button	Takes the user back to the next page

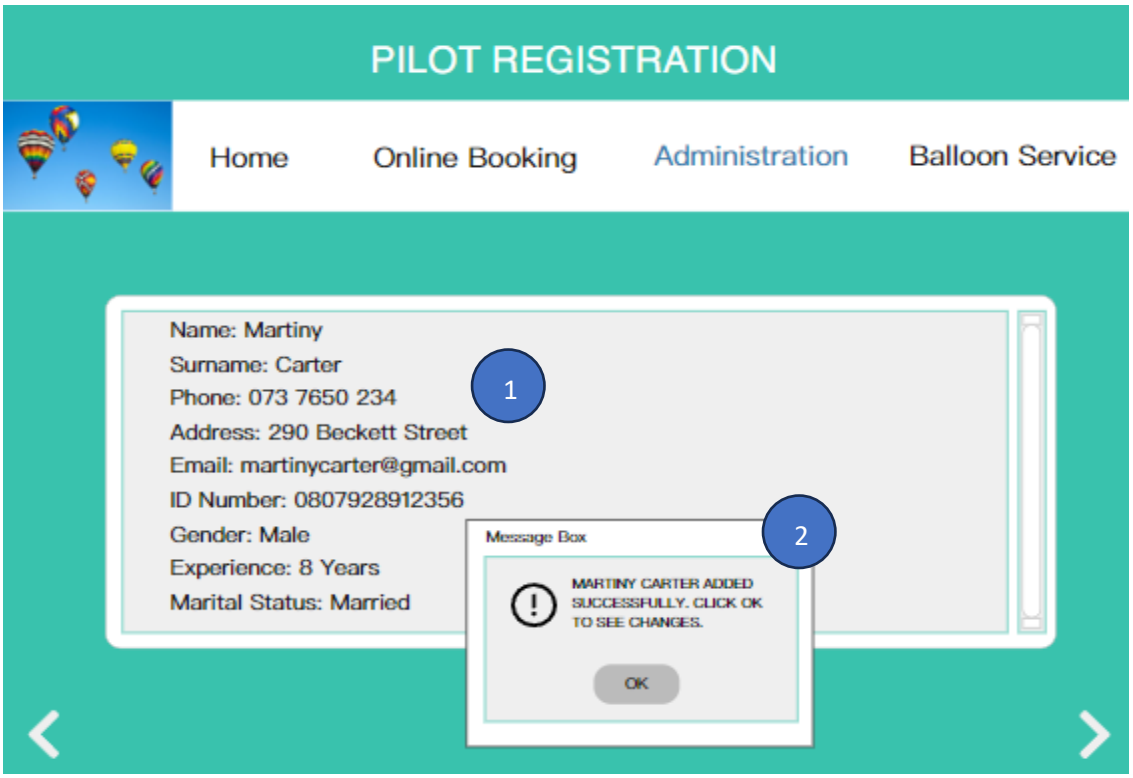


#	Item Name	Item Description
1	Pilot list results	The pilot list is displays for all the pilots registered on the system.
2	Edit button	For editing the pilot's name entered on the search bar
3	Delete button	This button is for removing a pilot from the system.
4	Message box with an OK button	Displays a message notifying the user the pilot does not exist on the system.
5	Cancel button	For the pilot registration cancellation.
6	Add button	A button for registering the pilot onto the system.



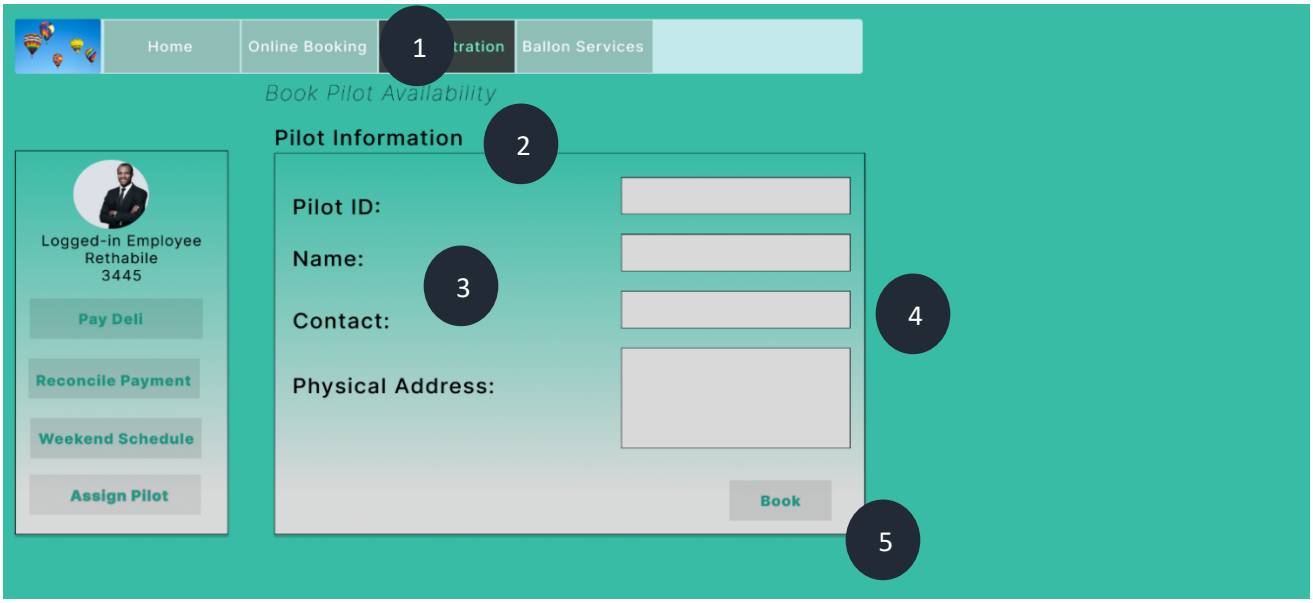
#	Item Name	Item Description
1	Labels: • Name • Surname • Phone • Address • Email • ID Number • Gender • Experience • Marital Status	Label for the following information input: • Name • Surname • Phone • Address • Email • ID Number • Gender • Experience • Marital Status
2	Textboxes for the following information: • Name • Surname	Textboxes with the following information input: • Name • Surname

	• Phone • Address • Email • ID Number • Gender • Experience • Marital Status	• Phone • Address • Email • ID Number • Gender • Experience • Marital Status
3	Add button	For adding a new pilot onto the pilot table.
4	Cancel button	For cancelling the registration of a new pilot.

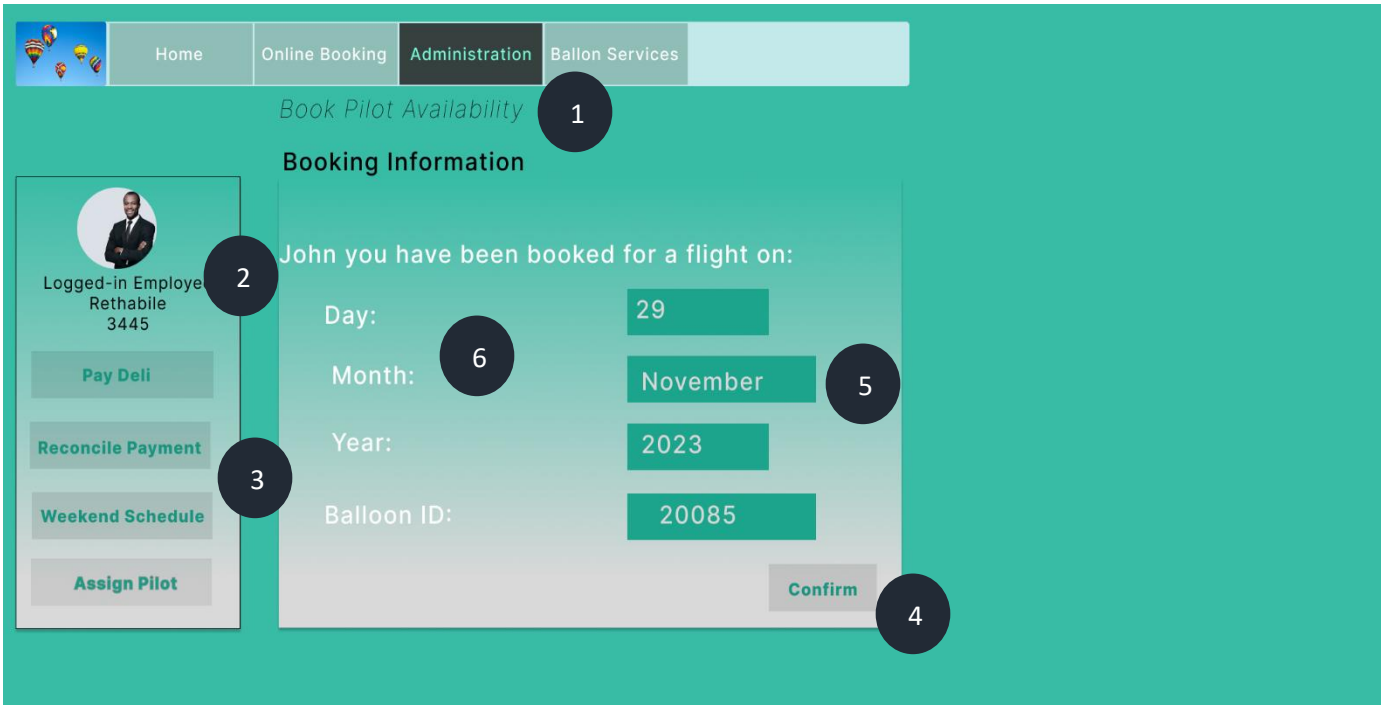


#	Item Name	Item Description
1	Added pilot details	Added pilot details for the newly registered pilot.

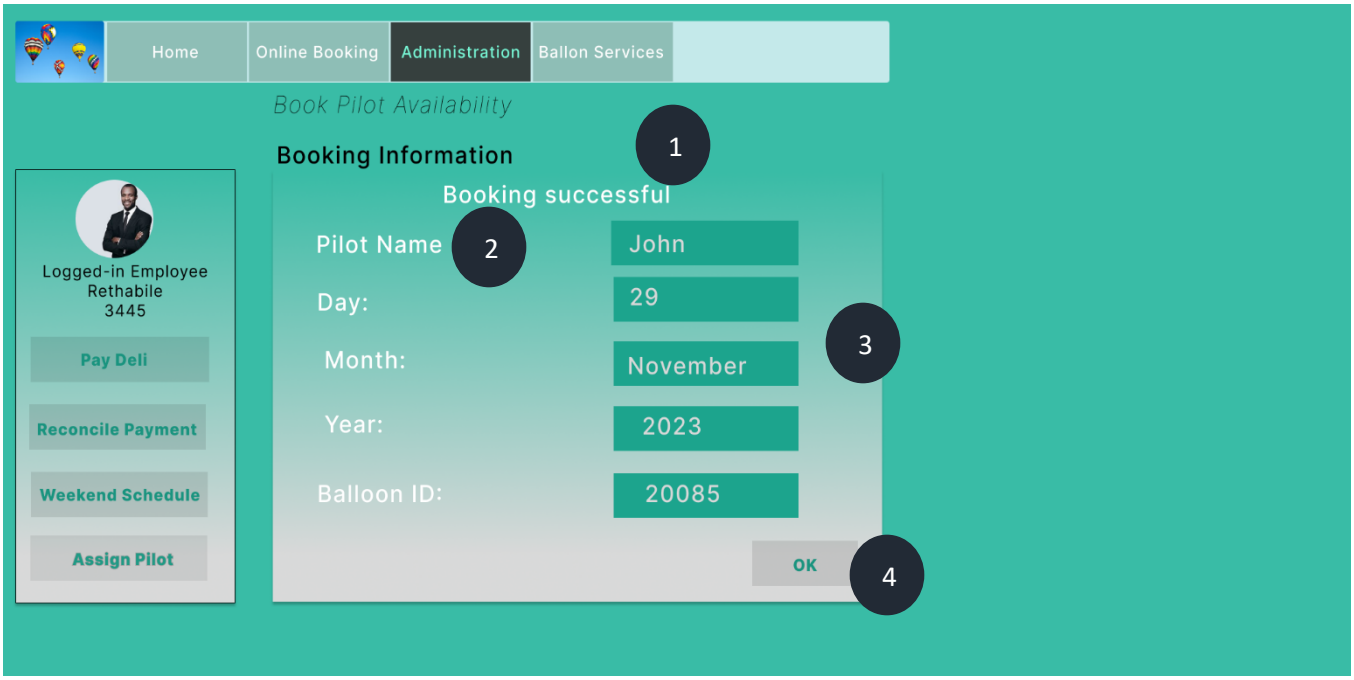
2	Message box with an OK button	Display a message that notifies the user that the new pilot has been successfully registered onto the system.



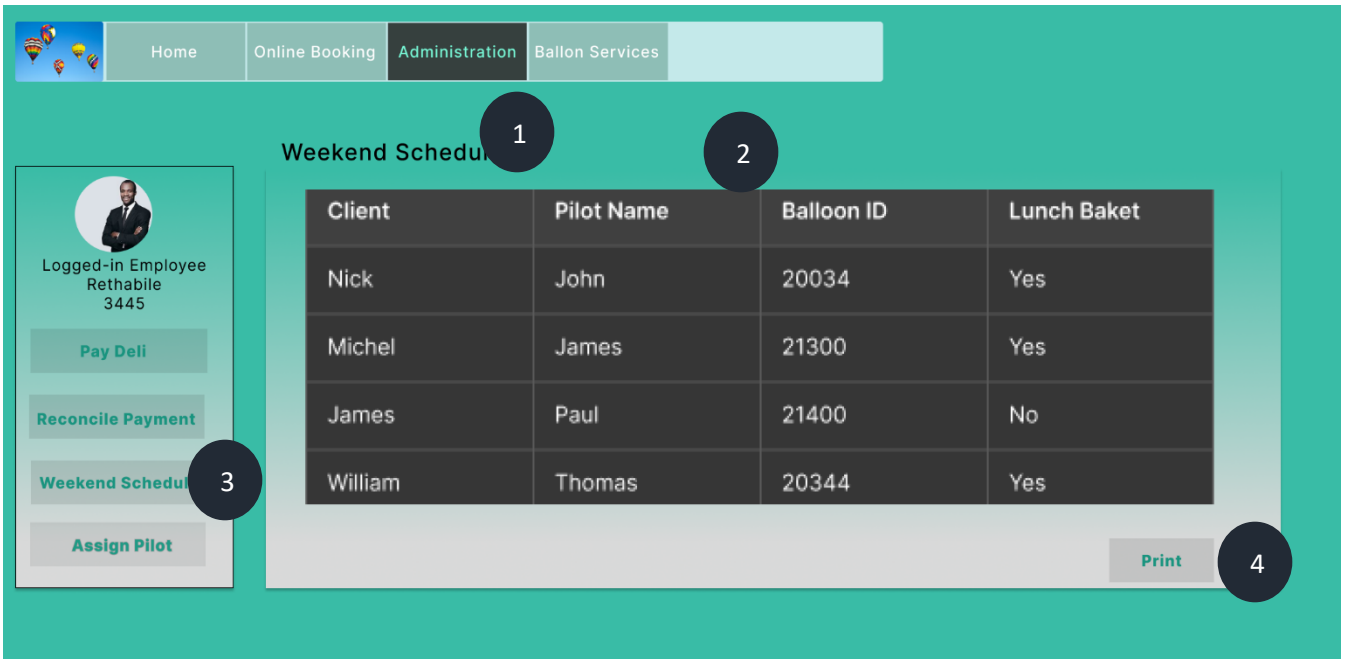
#	Item	Item description
1	Administration navbar	Used to show the active navbar
2	Groupbox name	Used to show the type od data required in the groupbox
3	Labels	Represent the type od data that needs to be filled.
4	Text boxes	Used for input.
5	Book button	Used when every data is filled and to proceed.



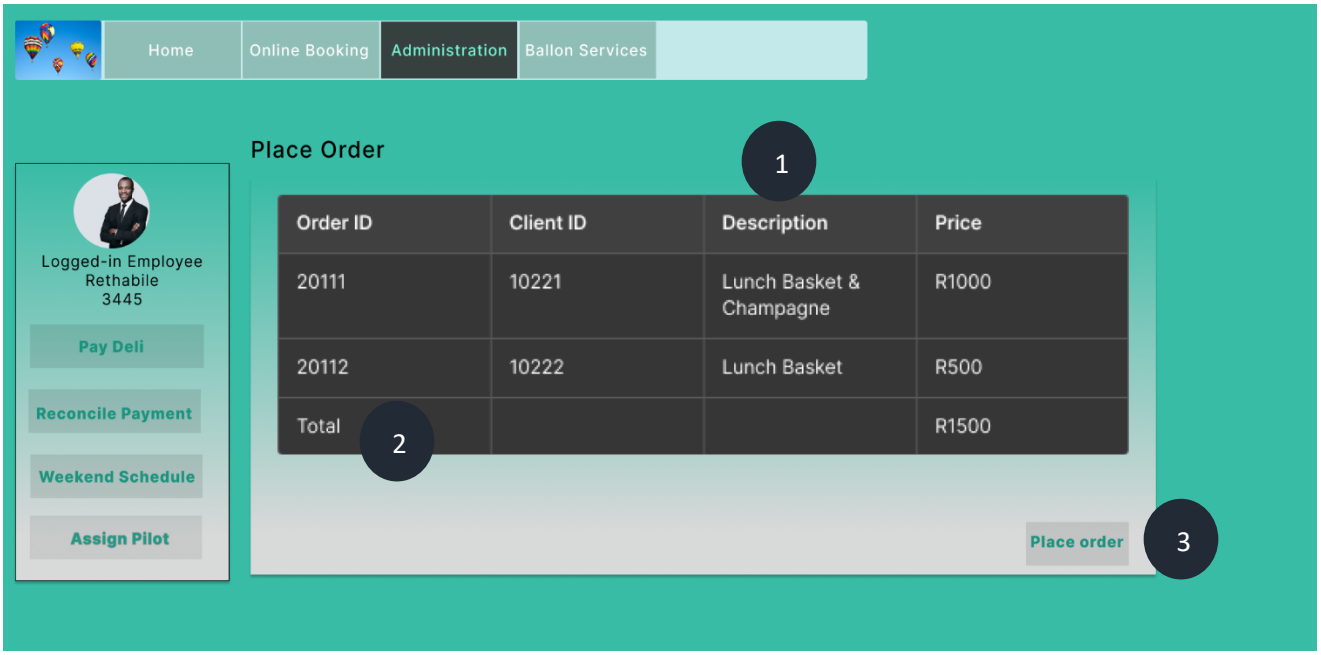
#	Item	Item description
1	Book Pilot Availability heading label	Used to show the page displayed
2	Employee image and details label	Used to show which employee is logged in.
3	Buttons	Used to navigate to another page.
4	Confirm button	Used for confirmation if the displayed information is correct
5	Text boxes	Used to fill up what was filled in the previous page (grayed out)
6	Labels	Describe what's in the textbox.



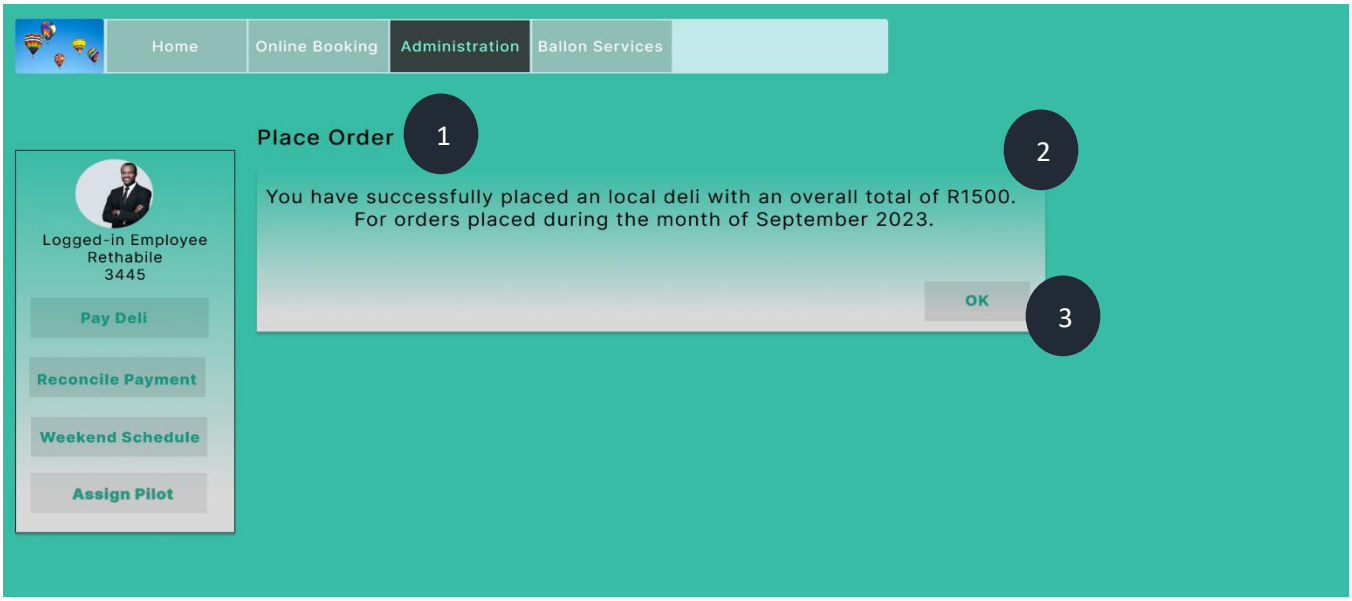
#	Item	Item description
1	Booking Successful label	Used to show that booking is successful.
2	Pilot Name label	Added name to the collection.
3	Labels	Used to show output results
4	Ok button	Used to return to the home page



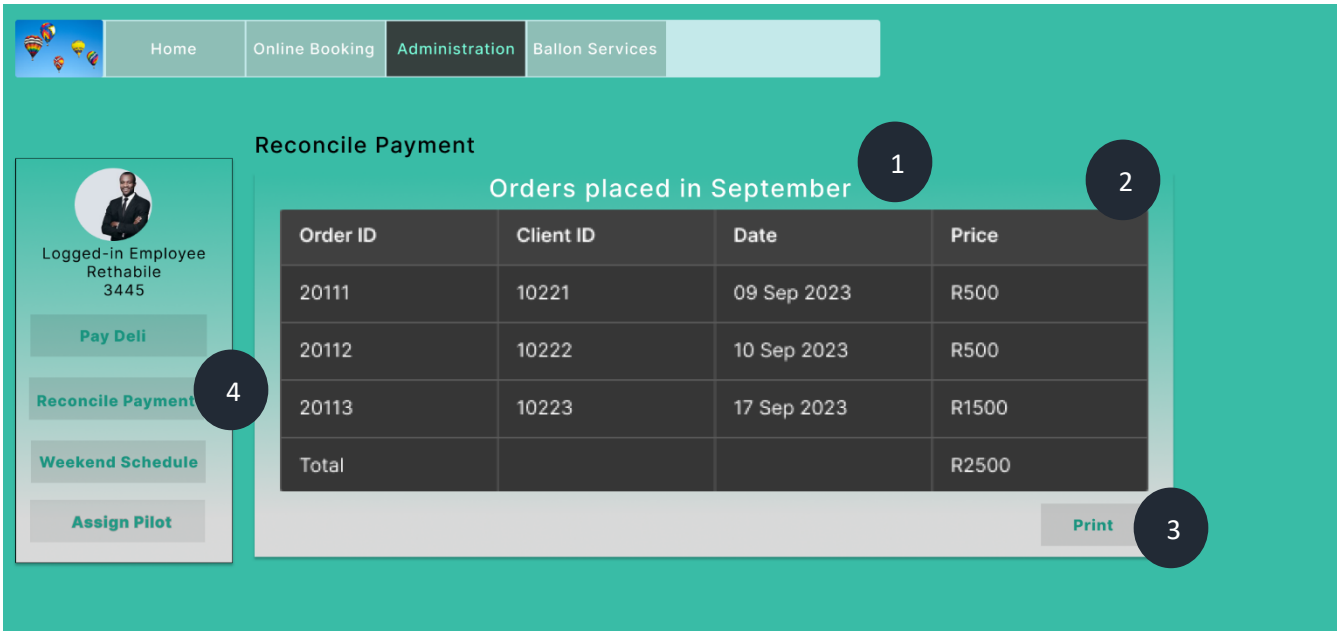
#	Item	Item description
1	Weekend Schedule groupbox name	Used to describe the type of data in the groupbox.
2	Schedule table	Used to group some certain types of data.
3	Weekend schedule button	Active button.
4	Print button	Allows the user to print the data presented.



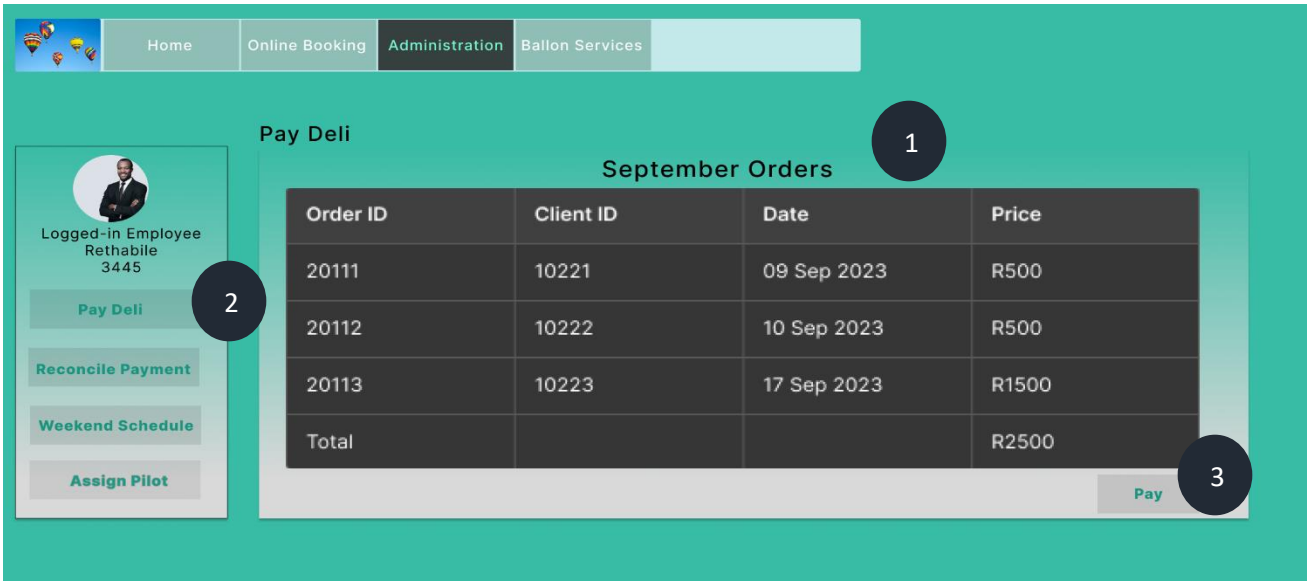
#	Item	Item description
1	Order table	Used for all the orders placed.
2	Total row	Used to calculate the overall total of all the products.
3	Place order button	Confirmation from the use to place the displayed order.



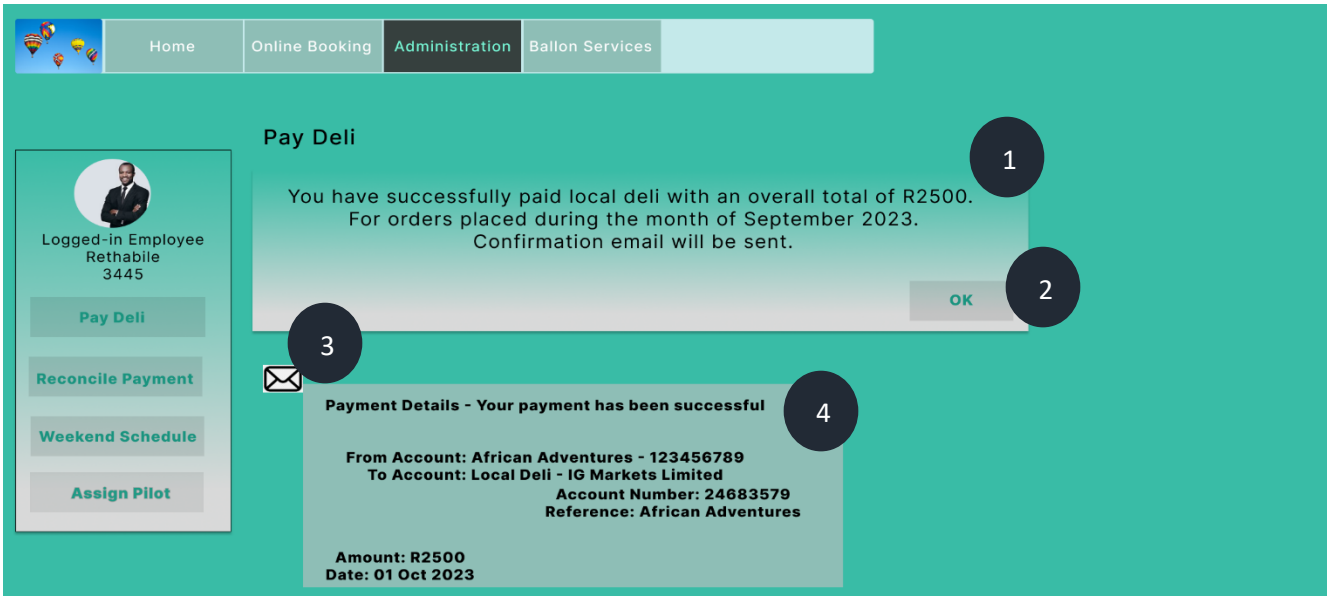
#	Item	Item description
1	Place order groupbox name	Used to describe the type of data in the groupbox.
2	Success message label	Used to inform the user that a particular task was performed.
3	Ok button	Used to take the user back to the home page.



#	Item	Item description
1	Orders month label	Used to show the month the orders were placed in.
2	Orders table	Shows orders data
3	Print button	Allows the user to print the data displayed and exit the page.
4	Reconcile button	Active button for the page.



#	Item	Item description
1	Orders month label	Used to show the month the orders were placed in.
2	Orders table	Shows orders data.
3	Pay button	Used when the user has confirmed everything a pay for orders.

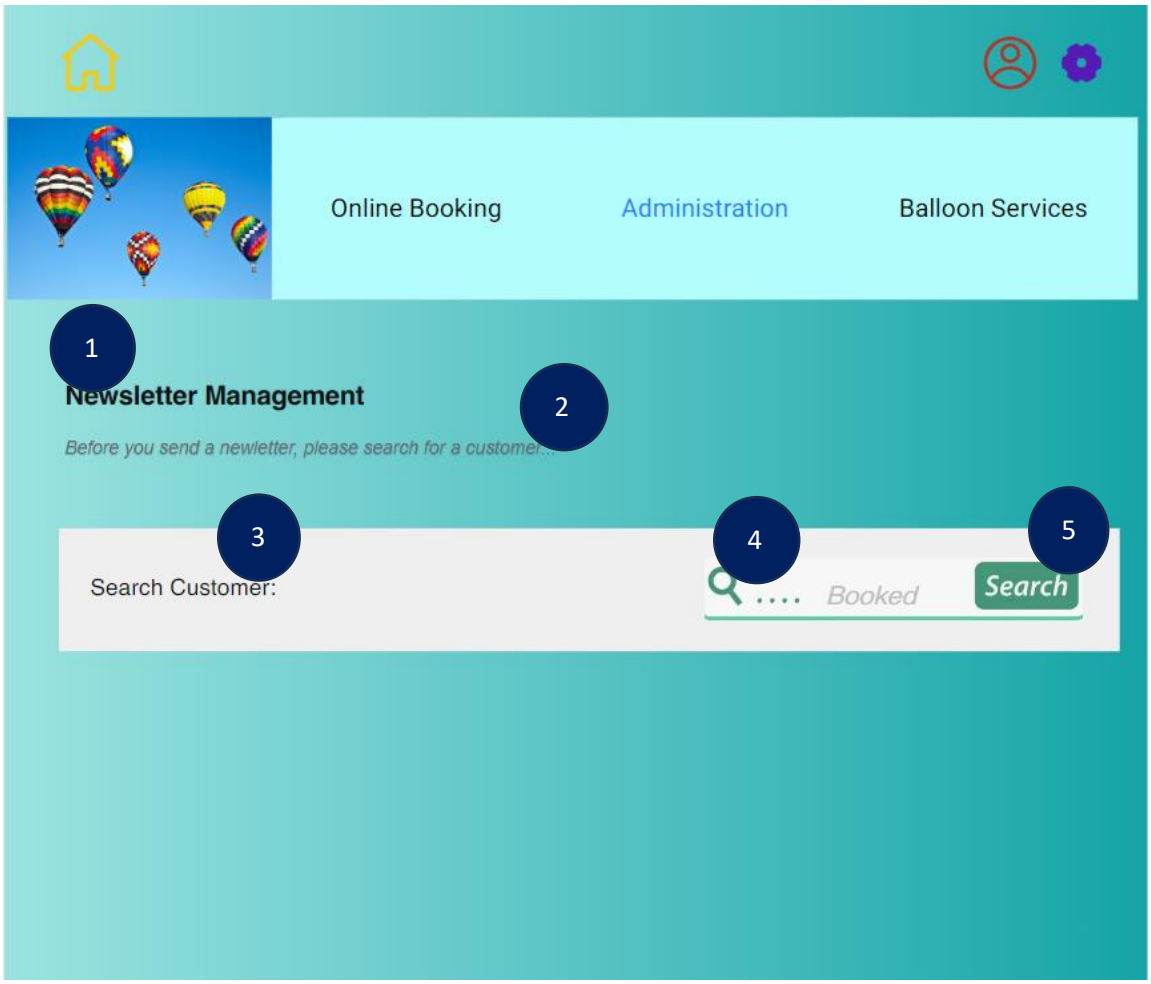


#	Item	Item description
1	Payment successful label	Used to show that a particular action was successful.
2	Ok button	Used to take the user back to the home page.
3	Email sent logo	Represent the email sent to the user.
4	Email message label	All the information contained in the email.

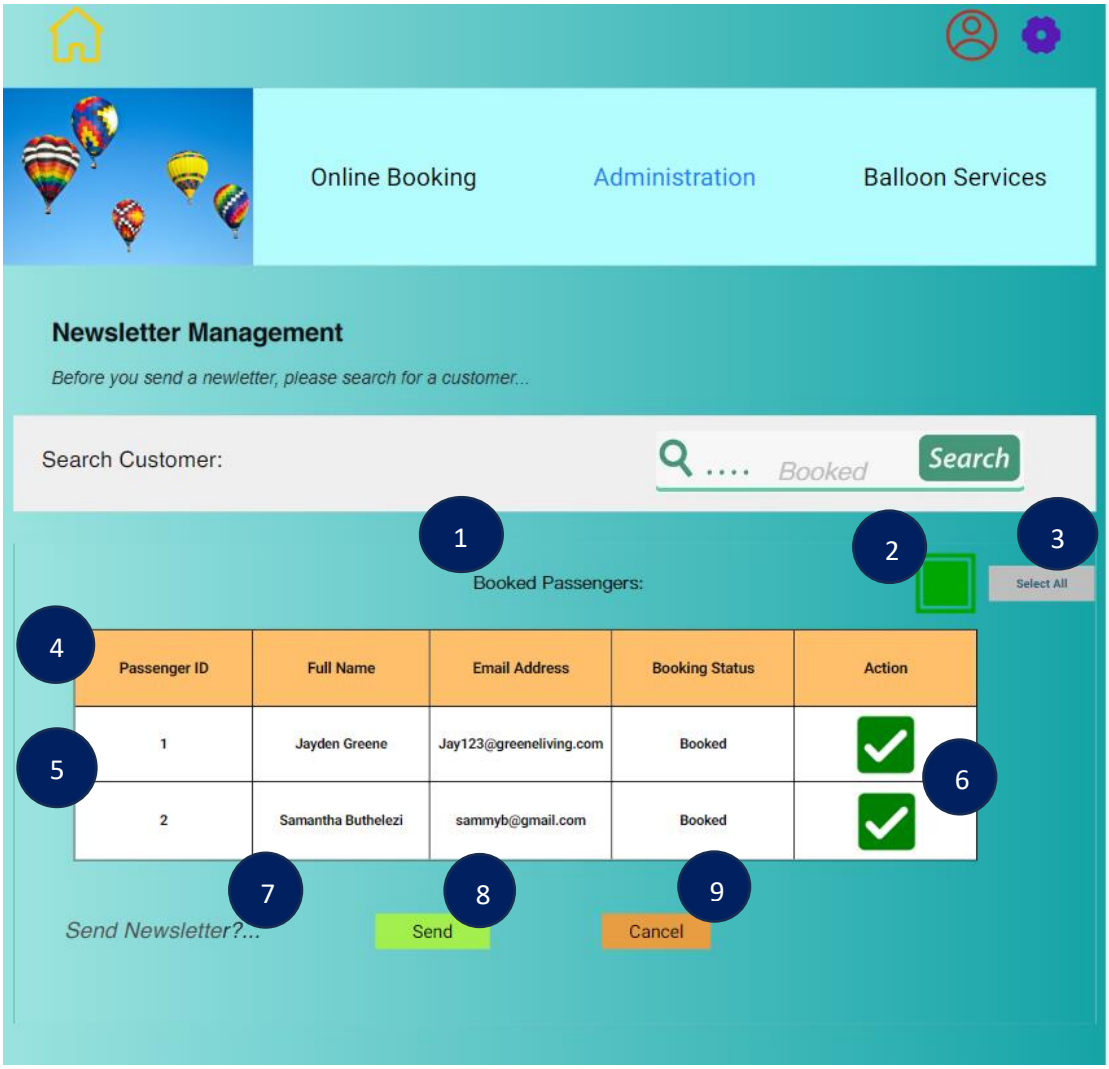
2.12 Send Newsletter

#	Item Name	Item description
1	“Home” icon	When clicked, the icon will always redirect you to the home page (main page).

2	“Welcome” text label	This is an indication that you are on the home page. The first page of the website.
3	“profile” Icon	When clicked, this icon will take the user to their personal profile page that has all their personal details.
4	“Settings” icon	This icon when clicked will take the user to the sites settings where they can see their activity settings or change the theme colour.
5	Navigation bar	This is a collection of links to different section of the website.
6	“Online Booking” tab	This is an indication of a particular section a user is currently accessing. Allows the user to easily access information on the website.
7	“Administration” tab	This is an indication of a particular section a user is can access. Allows the user to easily access information on the website.
8	“Balloon Services” tab	This is an indication of a particular section a user is currently accessing. Allows the user to easily access information on the website.
9	“Book Now” button	When clicked, this button will take the user straight to the “Booking” page
10	Text label	This label indicates to the user that if they want to make a booking, they should
11	“Learn more” link label	When clicked the link directs the customer to the “About us” page, which is an overview of what African Adventures is and how you can stay in touch.
12	“About us” image	This is just an image for the website’s aesthetics.
13	“Customer Experience” image	This is just an image for the website’s aesthetics.
14	“Customer Experience” link label	When clicked the user gets sent to the ‘News and Testimonies’ page where passengers share their ride experiences.
15	“FAQs” image	This is just an image for the website’s aesthetics.
16	“Learn more” link label	When clicked a user is directed to the frequently asked questions page.
17	“Next” icon	When clicked, takes the user to the user to next page.
18	“Back” icon	When clicked, takes the user to the previous page they were accessing.



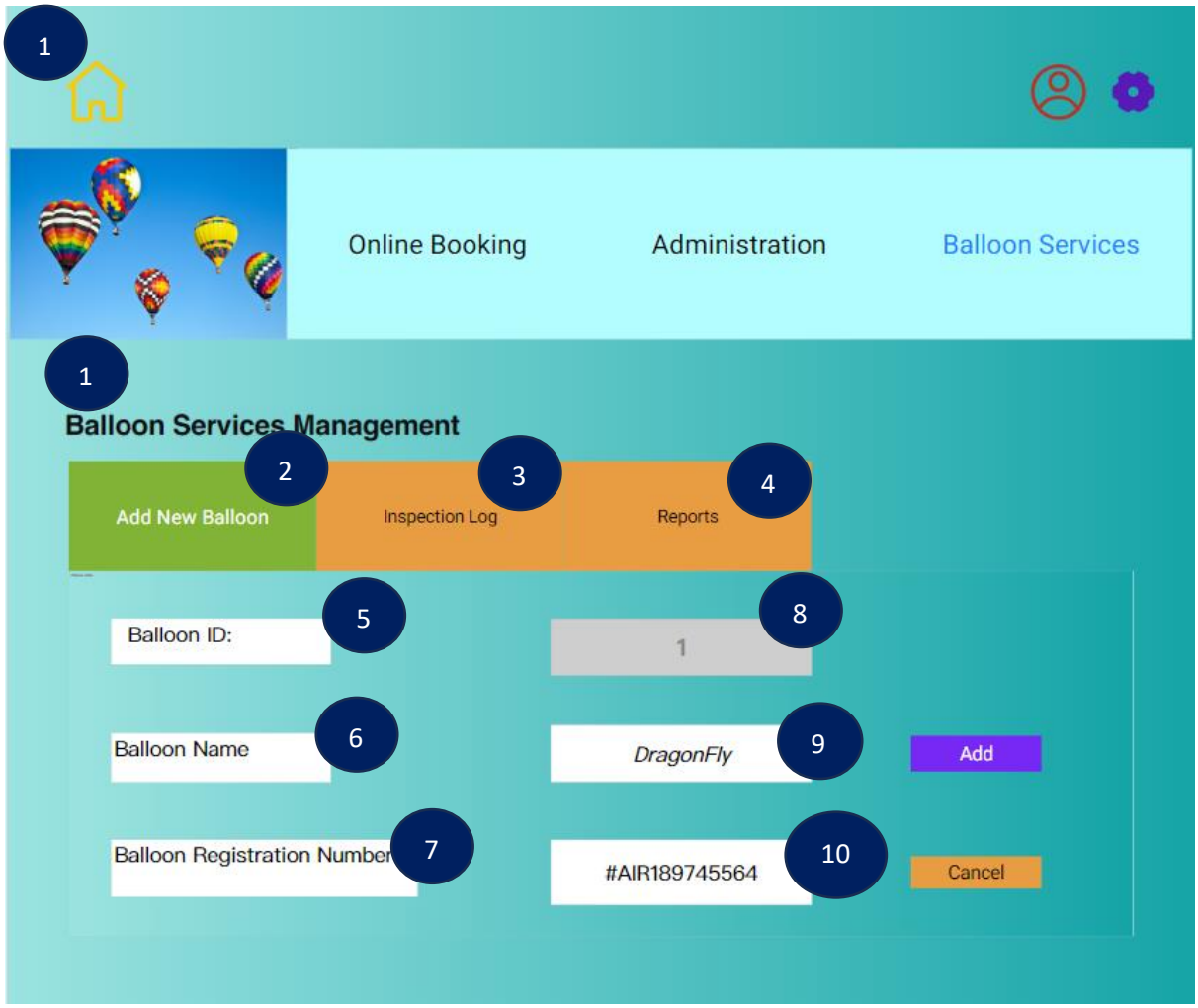
#	Item Name	Item Description
1	“Newsletter Management” text label	This just shows the contents within this particular tab “Administration”.
2	“Before you send a newsletter…” text label	This is a link giving instruction to the user, to take action.
3	“Search Customer” text label	This is a label showing the user that they can search for a customer (they can use the search functionality”.
4	“Magnifying glass” icon	When clicked, this icon will filter the users’ search.
5	“Search” button	When clicked this button will allow the user to search for a customer.



#	Item Name	Item Description
1	“Booked passenger” text label	This label serves the purpose of explaining what is contained within the table.
2	“Unchecked box”	This checkbox control shows that it has not been selected yet.
3	“Select All” button	This is a button that when clicked will select all the rows in the table.
4	“Table headers”	These headers give more detail about what is contained with the table fields (columns).
5	“Table rows”	This is where the actual data records of the passengers are being contained in the table.
6	“Checked box”	This checkbox is an indication that a selection of a record has been made.

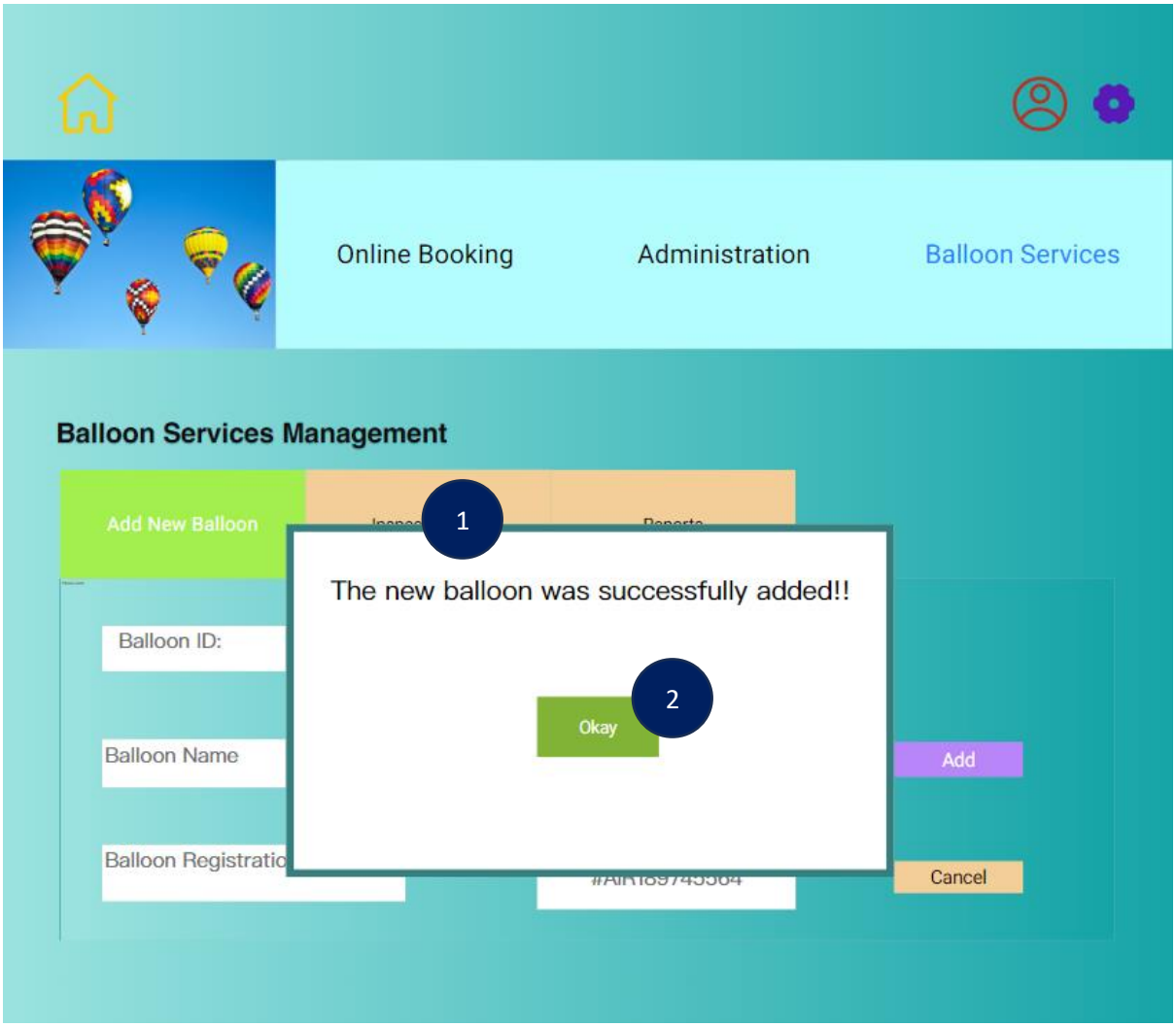
7	“Send newsletter” text label	This labels seeks confirmation from the user and at the same time prompting them to take the an action of sending or not sending a newsletter.
8	“Send” button	This button when clicked will allow the owner to send newsletter to selected previously booked passengers.
9	“Cancel” button	This button allows the user to cancel the sending of newsletter process.

3.1 Add New Balloon



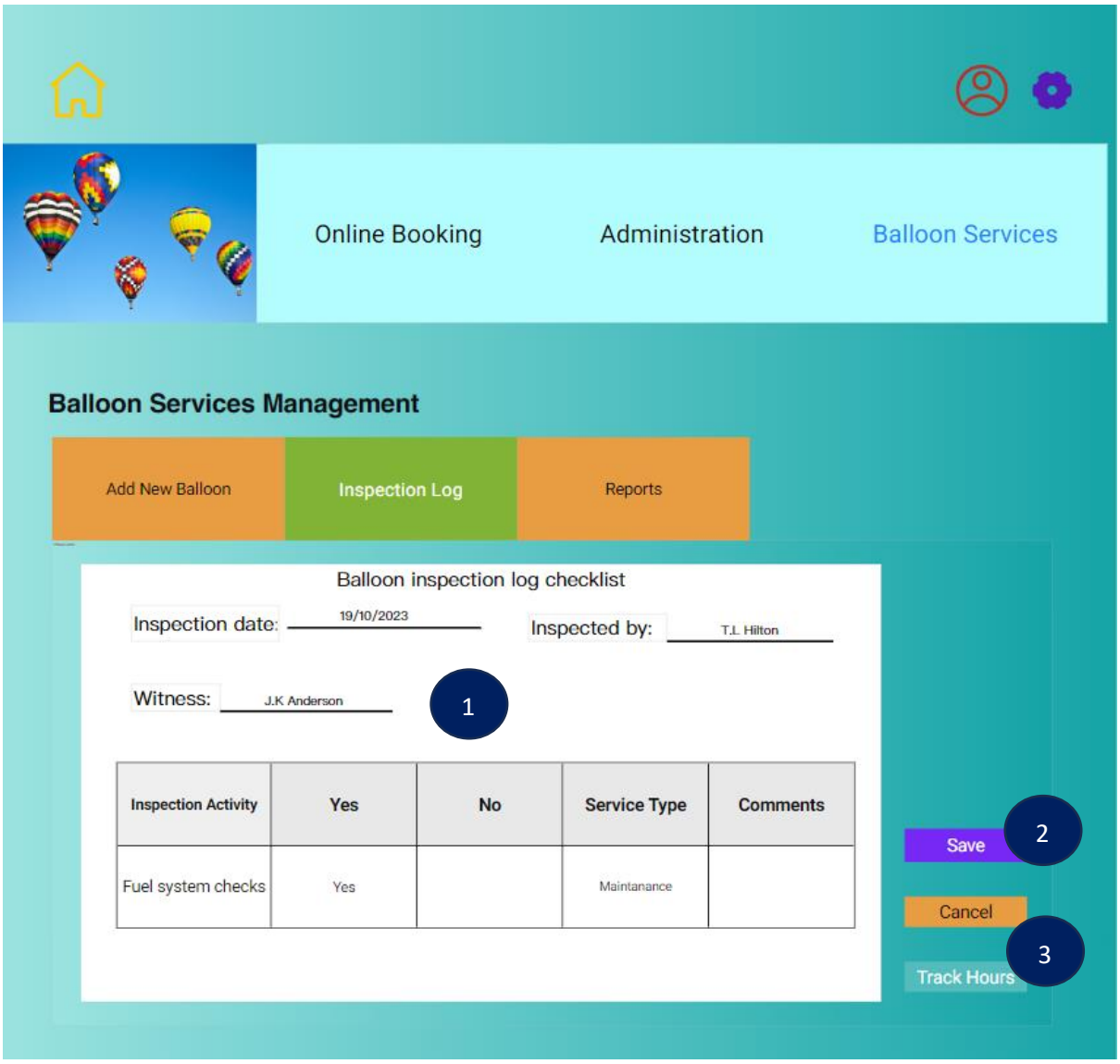
#	Item Name	Item Description
1	“Balloon Services Management” text label	Allows the user to know what is within the contents of the screen.

2	“Add new balloon” tab	This tab contains all the details related relating to adding a new balloon.
3	“Inspection log” tab	Contains all the details related to inspection logs. Can be accessed by a click.
4	“Reports” tab	When clicked the controls and details related to the reports is displayed.
5	“Balloon ID” text label	Represents the entering of a balloon id.
6	“Balloon Name” text label	Represents the entering of a balloon name value.
7	“Balloon Registration Number”	Represents the entering of a balloon registration number value.
8	“Balloon ID” textbox	Represents the input field for the textbox. Which is greyed out becomes it is automatically generated by the system.
9	“Balloon Name” textbox	Represents the input field for Balloon Name textbox.
10	“Balloon Registration Number” textbox	Represents the input field for Balloon registration number textbox.



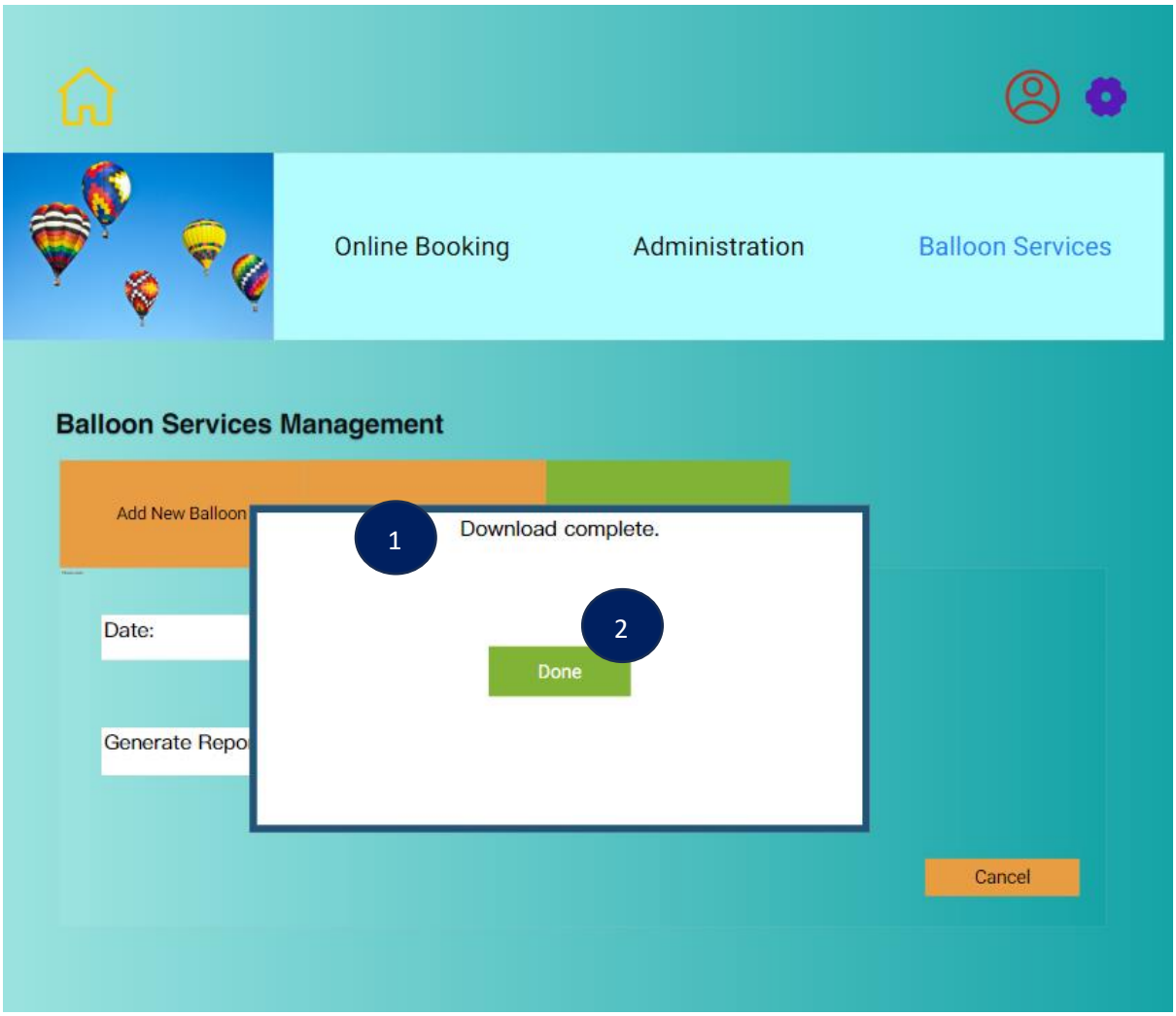
#	Item Name	Item Description
1	Rich Textbox	The rich textbox is a representation of a pop-up message with a text label, confirming successful exaction of a process.
2	“Okay” button	When clicked, this button will redirect the user back to the Add new balloon screen.

3.3 Capture log inspection and balloon services



#	Item Name	Item Description
1	Rich text box	This contains all the information related to the balloon inspection log.
2	“Save” button	This save button saves the inspection log to the database when the user clicks on it.
3	“Track hours” button	This button will automatically calculate the hours the balloon was assigned to a trip and actually made a trip.

3.4 Generate log inspection report



#	Item Name	Item Description
1	Rich text box	Contains a complete execution message after generating and downloading the inspection log report.
2	“Done” button	Will redirect user to Balloon service management when clicked.

Conclusion:

With the provided screen specification, African Adventures can be able to create a wonderful system that will satisfy them and their user. This is also cost effective as avoids costly design issues.

3.Output Design

3.1 Design Principles

Introduction:

Welcome to the heart of our site, where we explore the design principles that shaped this online balloon ride. Like a well-designed hot air balloon, a website must follow key design principles to create a delightful and informative journey for our passengers. This section covers the basic design principles, the look, feel and functionality of the African Adventures' website. Much like the structural components of a hot air balloon, these principles operate in synergy to ensure a seamless and enjoyable experience for our esteemed passengers.

Screen Design Consideration	Design Guideline (Description)
1. Balloon-themed aesthetics	A use of ballooned themed colours through the pages, complimenting the African Adventures' services.
2. High-quality imagery	Showcase the balloons, ballooning sites, and the entire experience with high-resolution photographs and videos. Visuals are extremely important in enticing potential clients.
3. Responsive design	To create a smooth user experience, ensure that the website is adaptable and adjusts to multiple screen sizes, including mobile devices.
4. Clear navigation	Use clear menus and a logical structure to implement easy navigation. Visitors of the site should have little trouble finding information regarding balloon rides, locations, price, and contact information.
5. User-centred content	Create content that addresses potential customers' questions and concerns, such as safety, booking, and frequently asked questions (FAQs).
6. Constraints	Limit the user's behaviour to what is permissible, if a user does not have authorization to access a certain portion of the system, Restrict the user's access to it by "greying out" options on menus or not displaying alternatives to users.
7. Mapping	Allows the user to define the relationship between possible actions and outcomes. users should not be confused about what is expected of them
8. Information hierarchy	Prioritize and organize information logically, such as details about the experience, flight options, security measures and prices.
9. Home page	The organisation's features should be illustrated on the home page. The value of the website and businesses should be easily identifiable when it is only a few minutes for users to spend on its home page. It must be made clear what the website's purpose is.

10. Privacy and Security	Communicate your commitment to user privacy and security, for online bookings and payments.

Conclusion:

As we conclude our discussion on the website's design principles, we hope you now have a better understanding of the careful thought that goes into crafting your balloon ride adventure. Our dedication to design principles like user-centered design, responsive layouts, high-quality imagery, and easy navigation is aimed at ensuring your online experience is as memorable as your actual balloon ride. our website is meticulously structured to provide you with the information you need while capturing the enchantment of ballooning. We appreciate you joining us on this journey and eagerly anticipate welcoming you on board in our real balloons in the near future

3.2 Completeness of reports

Introduction

This section delves into the system's generated reports and outputs, encompassing essential documents such as indemnity forms, tickets, and inspection reports. It offers a comprehensive overview of each report, including the layout design, descriptions of their components, and the intricacies of their design. To enhance clarity, a table has been employed to outline the criteria for these reports, and each report is subjected to a detailed examination, shedding light on the circumstances triggering their generation, their purpose, and the specific information they encapsulate.

INDEMNITY FORM

I, the undersigned, understand that I will be participating in a hot air balloon ride with AFRICAN ADVENTURES on XX/XX/XXXX. In consideration of being allowed to participate in this activity, I hereby acknowledge and agree to the following terms and conditions:

ASSUMPTION OF RISK: I am aware that participating in a hot air balloon ride involves certain inherent risks and dangers. These risks include, but are not limited to, sudden changes in weather, wind conditions, equipment malfunctions, and other unforeseeable events. I acknowledge that I am voluntarily participating in this activity with full knowledge of these risks.

RELEASE OF LIABILITY: I hereby release and discharge AFRICAN ADVENTURES, its agents, representatives, employees, and all associated personnel from any and all claims, liabilities, demands, actions, or causes of action that may arise from or relate to my participation in the hot air balloon ride.

INDEMNITY AGREEMENT: I agree to indemnity and hold harmless AFRICAN ADVENTURES, its agents, representatives, employees, and all associated personnel from any and all liabilities, costs, or expenses, including but not limited to reasonable attorney's fees, that may be incurred as a result of any claims brought by myself or by any third party, including my family members, arising out of my participation in the hot air balloon ride.

MINORS: In the case of a minor participant, I, as the parent or legal guardian, acknowledge and agree to all the terms and conditions on behalf of the minor.

AGREEMENT TO FOLLOW INSTRUCTIONS: I agree to abide by all instructions and safety guidelines provided by the staff of AFRICAN ADVENTURES during the course of the hot air balloon ride.

I have read and understand this Indemnity Form in its entirety, and I am signing it voluntarily. I am aware that by signing this form, I am releasing certain legal rights, including the right to sue AFRICAN ADVENTURES for any injuries or damages resulting from my participation.

Participant's Name (Print): _____
Participant's Signature: _____
Date: _____

If the participant is a minor, please provide the following information:
Parent/Guardian's Name (Print): _____
Parent/Guardian's Signature: _____
Date: _____

This Indemnity Form is valid for the specific hot air balloon ride mentioned on XX/XX/XXXX and for all related activities associated with this ride.
AFRICAN ADVENTURES reserves the right to refuse participation or access to the hot air balloon ride to anyone unwilling to sign this Indemnity Form.

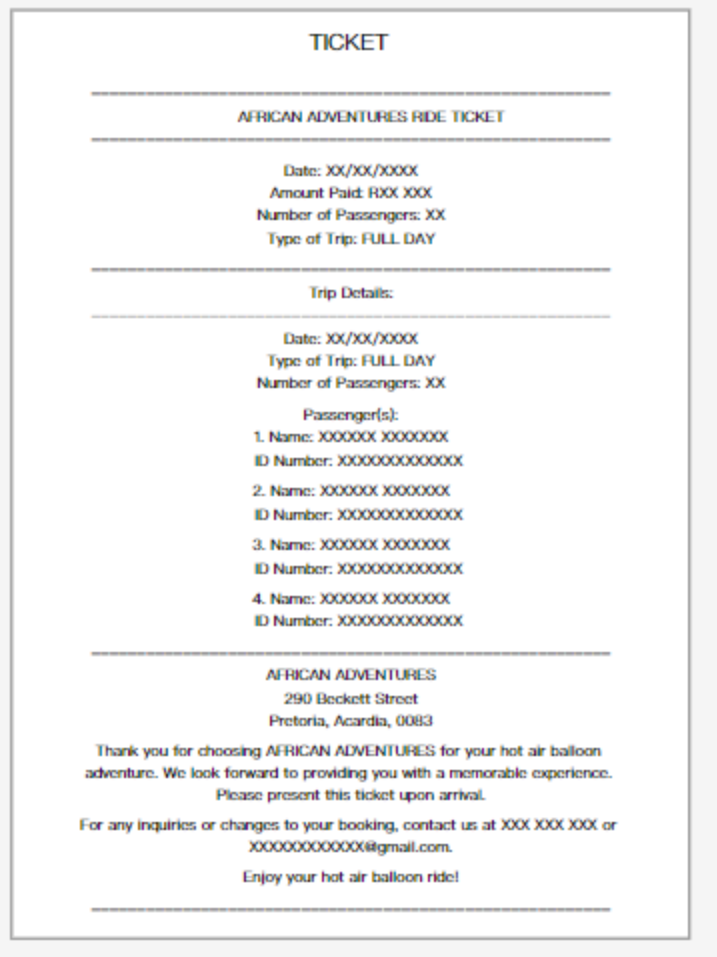
XXXXXX XXXXX
AFRICAN ADVENTURES
By: _____
XXXXXX XXXXX
Date: _____

The indemnity form is generated after the passenger has been checked in for their booking for a trp. Upon checking in the passenger must provide their reference number for their booking. The indemnity form must be signed and completed by the passenger. The passenger must not be given a ticket unless the indemnity form is fully completed and signed by the passenger. The indemnity form has the following layout information to be filled:

- The participants name, date and signature provided that the participant is 18 years and above.
- If the participant is a minor, the participants name, date and signature must be filled in underneath the first details for participants above 18 years old.

The indemnity form is generated along with date the form was generated and it also must contain the name and surname of the office administrator and space for providing their signature and the data the indemnity form was signed and lastly the name and surname of the participant that checked in for the trip must also be generated along with the indemnity form.

Criteria	Default	Type	Values
Passenger details for both minors and adults	Name & Surname	label	Name and Surname of the passenger checking in.
Date	DD/MM/YYYY	Date	The date the passenger checked in
Signature	Signature	Must be provide on blank space	Signature of passenger checking in
Phone Number	10	Number	Any valid existing phone number of the passenger.
Name & Surname of office administrator	Text	label	Name and Surname of office administrator



TICKET

AFRICAN ADVENTURES RIDE TICKET

Date: XX/XX/XXXX
Amount Paid: RXX XXX
Number of Passengers: XX
Type of Trip: FULL DAY

Trip Details:

Date: XX/XX/XXXX
Type of Trip: FULL DAY
Number of Passengers: XX

Passenger(s):
1. Name: XXXXXX XXXXXXX
ID Number: XXXXXXXXXXXXX
2. Name: XXXXXX XXXXXXX
ID Number: XXXXXXXXXXXXX
3. Name: XXXXXX XXXXXXX
ID Number: XXXXXXXXXXXXX
4. Name: XXXXXX XXXXXXX
ID Number: XXXXXXXXXXXXX

AFRICAN ADVENTURES
290 Beckett Street
Pretoria, Acardia, 0083

Thank you for choosing AFRICAN ADVENTURES for your hot air balloon adventure. We look forward to providing you with a memorable experience. Please present this ticket upon arrival.

For any inquiries or changes to your booking, contact us at XXX XXX XXX or XXXXXXXXXXX@gmail.com.

Enjoy your hot air balloon ride!

The ticket is generated after the indemnity form has been signed and submitted by the client to the office administrator. The ticket contains information about the passenger's details such as their name and surname along with other passengers' details. The date the ticket was received, the number of passengers checked in for and the type of trip the passenger checked in for are included in the ticket. Moreover, the phone number and email address for African Adventures is also included in the ticket in case passengers have any queries or complaints regarding trips. The ticket must be generated after the indemnity form has been generated and signed by the passenger checking in.

Criteria	**Default**	**Type**	**Values**
Number of passengers	1	Label	Any valid number
Date	DD/MM/YYYY	Date	Any date the passenger checked in

Type of trip	FULL DAY	Label	• FULL DAY • 2 Hours • 4 Hours • 2 Hours for groups • 4 Hours for groups
Passenger list	4	Text	Name and Surname of each passenger
African Adventure Contact No	10	Number	Any valid existing phone number
African Adventure email	africanadventures@gmail.com	Email Address	Any valid existing email address

001

Inspection Log Report

Date of Inspection: dd/mm/yyyy

Balloon ID: 20

Inspected By: M.L Parker

Pre–Flight Inspection

Inspection Activity	Yes	No	Comment
Envelope: Inspected for damage, wear, or tear.	X	X	XXX
Basket: Checked for structural integrity.	X	X	XXX
Fuel System: Inspected for leaks or damage.	X	X	XXX
Thorough rigging inspection.	X	X	XXX
Instruments: Verified for accuracy.	X	X	XXX

Testing and Inspections of Safety Equipment

– Emergency Landing Equipment: Checked for condition and accessibility. ☑

– Fire Extinguishers: Inspected and verified for proper operation. ☑

– First Aid Kit: Inspected for completeness and expiration dates. ☑

The purpose of the ‘Balloon Inspection Log Report‘ is to provide a comprehensive record of all inspections and maintenance activities related to the hot air balloons operated by African Adventures.

This report serves several essential purposes:

Safety Compliance: It ensures that our hot air balloons meet and adhere to all safety standards and regulations as required by aviation authorities.

Maintenance Documentation: It tracks and documents all maintenance and repair work performed on the balloons to ensure their ongoing airworthiness.

Management Oversight: It offers management and stakeholders a transparent view of the condition and maintenance status of our hot air balloons, aiding in decision-making related to safety and operational readiness.

Generally, the report is executed on the following occasions: Pre-Flight Inspections: These are conducted before each balloon flight to ensure that the equipment is in optimal condition.

Entities Involved: The following entities are typically involved in the generation and utilization of the ‘Balloon Inspection Log Report‘: Balloon Owner: The owner is responsible for initiating and overseeing the inspection process and generating the report.

Maintenance Team: The foreman and the technicians are responsible for any repair work, component replacements, or upkeep recorded in the report.

Layout of the Report: The ‘Balloon Inspection Log Report‘ includes the following key sections: Report Header: Contains the report title, date, and essential identification details of the balloon.

This comprehensive layout ensures that the report provides a detailed and accurate account of the balloon's inspection history, making it a valuable tool for safety compliance and operational readiness.

Report No:	001
Report Name:	inspection log report
Short Description	Generate inspection log report
Frequency:	Weekly

Criteria	**Default**	**Type**	**Values**
Date of inspection	DD/MM/YYYY	Date	Date of inspection
Balloon ID	1+	Label	Automatically generated by system (int)
Inspected by	Name and surname of Inspector	Label	Name and Surname of Inspector

Conclusion:

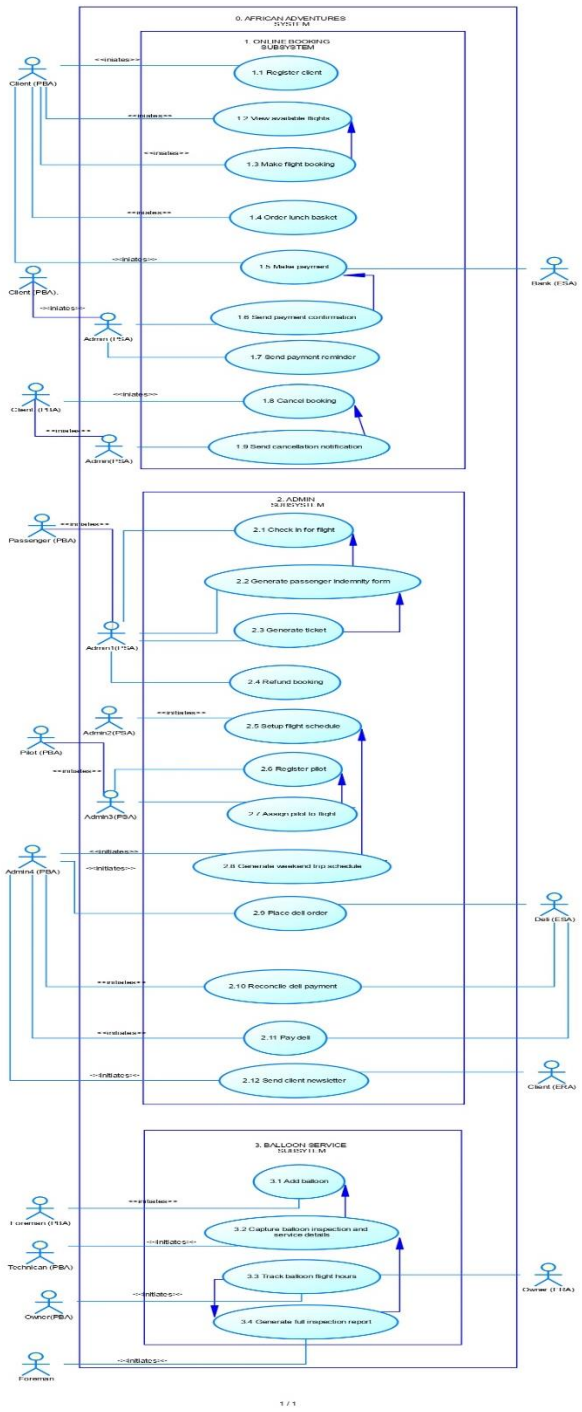
This section centers on the design aspect of the system's generated reports, offering valuable insights into the timing and mechanisms governing their generation. The tabulated layout of these reports not only offers a visual representation but also imparts an understanding of the essential information that must be included, thereby defining a structural framework that enforces necessary content requirements for the reports.ee

C. Technical Specification

1.Use cases

Introduction:

Use case diagrams in the context of a prototype and technical specification serve as a valuable tool for illustrating and clarifying the functional requirements and interactions within a system or software project.



1 / 1

Conclusion:

This use case is particularly useful for identifying potential technical challenges, as they reveal the necessary inputs, outputs, and dependencies for each use case. The use case informs the technical specification, enabling a more accurate and comprehensive development plan.

1.2 Design Use Case Documentation

Technical narrative serves as indispensable tool for conveying complex technical information, facilitating understanding. This bridge the gap between intricate technical concepts and decision making.

{African Adventures}

Author (s): **Phumzile** **Date:** **24/10/2023**

Version: **1**

USE CASE NAME:	Register Client	**USE CASE TYPE**	
USE CASE ID:	1.1	Business Requirements:	☐
PRIORITY:	Hight	**System Analysis:**	☐
SOURCE:	Case Study	System Design:	√
PRIMARY BUSINESS ACTOR	Client		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case describes the fundamental activity to booking a trip with African Adventures which is to register as a client.it takes one through the technical process and conclude with a 'registration successful message'		
PRE-CONDITION:	User logged in to African Adventures website		
TRIGGER:	User wants to register as client		
TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response: Manual Action (PSA)	System Response: Automated Action (System)
	Step 1: User wants to register as client		

	Step 2: User Clicks on the online booking tab		Step 3: Complete Registration page loads with Register as the title Labels for: name, surname, phone number, email,ID and physical address And textboxes for each label A Next Button
			Step 4: System prompts user to enter detail according to form controls
	Step 5: User enters required detail. And clicks on the next button		Step 6: A confirm detail message box loads with an okay and cancel buttons
			Step7: Inset statement SQL query saves detail to client table in database
	Step 8: Users selects ok button [ALT]		Step 9: message box 'Client successfully registered' loads
ALTERNATE COURSES:	Step 8 ALT: User clicks cancel button redirected to register screen		
CONCLUSION:	User registered as client. Return to home screen		
POST-CONDITION:	User detail exists on system		
BUSINESS RULES	•		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	•		
ASSUMPTIONS:	•		
OPEN ISSUES:			

{African Adventures}

Author (s): **Phumzile** **Date:** **24/10/2023**

Version: _____

USE CASE NAME:	View Available Flights	**USE CASE TYPE**	
USE CASE ID:	1.2	Business Requirements:	☐
PRIORITY:	Middle	**System Analysis:**	☐
SOURCE:	Case Study, Use case narrative	System Design:	√
PRIMARY BUSINESS ACTOR	Client		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	The use case occurs as the client is making a booking and wants to view all the available flights so they can select a slot that best suits their schedule. The Client will fill in their information for make booking after which they will click on the view dates button which will result in a schedule of available flights displaying		
`	User is a registered client		
TRIGGER:	Client wants to view available flights		
TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1: Client wants to view available flights to male booking		
	Step 2: Client clicks on the View dates button which is adjacent to the Trip date. textbox		Step 3: An Available Flights page loads with Labels fir Saturday and Sunday, Timeslot

			And labels of the available timeslots at the time of booking And a Back button
			Step 4: An SQL query selects all available slots from the database and that is how the available slots page is displayed
	Step 5: Client clicks on the Back button		Step 6: c# code redirects user to Bookings page
ALTERNATE COURSES:			
CONCLUSION:	Client has viewed the remaining flights for the weekend		
POST-CONDITION:	Redirected to Booking Page		
BUSINESS RULES	•		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	•		
ASSUMPTIONS:	•		
OPEN ISSUES:			

{African Adventures}

Author (s): **Phumzile** **Date:** **24/10/2023**

Version: **1**

USE CASE NAME:	Make Flight Booking	**USE CASE TYPE**	
USE CASE ID:	1.3	Business Requirements:	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	Case Study	System Design:	√
PRIMARY BUSINESS ACTOR	Client		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	The use case begins when the client wishes to make a trip booking, the client will begin by clicking on the online booking tab after which they will direct to the booking page. It is here that the client will fill out all relevant booking information.Afterwhich a prompt to confirm will appear, it is after confirming that the clients booking status will be updated.		
PRE-CONDITION:	Logged in to system		
TRIGGER:	Clients wants to book a trip		
TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1: Client wants to make a trip booking		
	Step2: Client clicks on the online booking tab		Step3: Online booking screen opens with radio buttons for register, make booking, make payment and order lunch basket

	Step4: Client selects make booking radio button:		Step 5: Booking page loads with labels: Booking, booking type, trip date, no. passengers, Buttons: view dates, submit Textboxes: no passenger, trip dates Dropdown: booking type
	Step 6Client fill in booking type info, no. passenger		Step 7: Client clicks on view dates button
			Step 8:sql query retrieves available slots for the weekend
			Step 9: available slots display
	Step 10: adds date to booking		
	Step11: client clicks submit button		Step 12: Confirm booking message box appears with buttons: okay and cancel [ALT]
	Step13; Client Click okay		Step 14: SQL query writes to booking status table to change status to booked
ALTERNATE COURSES:	Step 12 ALT: Client clicks cancel button ,redirected to booking page		
CONCLUSION:	Booking status in database updated		
POST-CONDITION:	Redirected to home Page		
BUSINESS RULES	•		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	•		
ASSUMPTIONS:	•		
OPEN ISSUES:			

{African Adventures}

Author (s): **Phumzile** **Date:** **24/10/2023**

Version: **1**

USE CASE NAME:	Add Lunch Basket	**USE CASE TYPE**	
USE CASE ID:	1.4	Business Requirements:	☐
PRIORITY:	Middle	**System Analysis:**	☐
SOURCE:	Case Study	System Design:	√
PRIMARY BUSINESS ACTOR	Client		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case begins when a user wishes to add lunch basket to their booking. The client will be given the option to choose the quantity of lunch baskets they require after which this information will be submitted and saved to the database		
PRE-CONDITION:	Logged in to system. Making/made a booking		
TRIGGER:	Client wants to add a lunch basket to their booking		

TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1: Client wants to add a lunch basket to their booking		
	Step 2: Click clicks on the Online Booking Tab		Step 3 the online booking screen loads with 4 radio buttons. -register client

			-book a trip -add lunch basket -make payment
	Step 4: Client selects lunch basket radio button		Step 5: Add lunch basket screen loads with labels: lunch basket, add lunch basket, And check box for add lunch basket and next button
	Step 6:Client checks add box		Step 7:Inser SQL query written into order table of database
			Step 8:Lunch basket added successfully message box appears with ok button
ALTERNATE COURSES:			
CONCLUSION:	Lunch basket added to client trip in Database		
POST-CONDITION:	Redirected to home page		
BUSINESS RULES	•		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	•		
ASSUMPTIONS:	•		
OPEN ISSUES:			

{African Adventures}

Author (s): **Phumzile** **Date:** **24/10/2023**

Version: **1**

USE CASE NAME:	Make Payment	**USE CASE TYPE**	
USE CASE ID:	1.5	Business Requirements:	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	Case Study	System Design:	√
PRIMARY BUSINESS ACTOR	Client		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• Bank		
DESCRIPTION:			
PRE-CONDITION:	Client logged in to system Client has made a booking		
TRIGGER:	Client wants to make payment		

TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1: Client wants to make payment		Step2: Make Payment page loads with labels. : Booking reference, Booking Status, Name on card and card number. Textboxes for: Booking Status, Name on card and card number.

			Buttons: Verify and Next
	Step 3: Client enters booking number		Step 4: System SQL query retrieves corresponding booking status and autofill's booking status textbox
			Step 5: Booking Status retrieved message box
	Step6: Client Clicks ok button		
	Step 7: Client fills in remaining credit card info		
	Step7: Click next button		Step 8:System alongside bank server verify credit card info
			Step 9: Transaction successful notification [AALt]
ALTERNATE COURSES:	Step 9 Alt Payment declined /incorrect info notification		
CONCLUSION:	Tranaction processed successfully		
POST-CONDITION:	Client redirected to home page		
BUSINESS RULES	•		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	•		
ASSUMPTIONS:	•		
OPEN ISSUES:			

{AFRICAN ADVENTURES}

Author (s): **M. SADIKI** **Date:** **22/10/2023**

Version: _____

USE CASE NAME:	Send payment confirmation	**USE CASE TYPE**	
USE CASE ID:	1.6	Business Requirements:	☐
PRIORITY:	Medium	**System Analysis:**	☐
SOURCE:	African Adventures case study	System Design:	√
PRIMARY BUSINESS ACTOR	Time		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• Client		
DESCRIPTION:	This use case describes an event in sending a payment confirmation to a client after their payment has been successfully made and processed.		
PRE-CONDITION:	The payment for the booking has been successfully processed.		
TRIGGER:	Payment processing is completed successfully.		

TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1: Client successfully made the booking payment [ALT]		**Step 2**: Upon successful payment processing, the system captures payment details and identifies the client associated with the payment.
			Step 3: The system retrieves the client's contact information, including their email address, from the booking record where it

			was saved during registration (Invoke use case 1.1Register client)
			Step 4: Using the client's email address, the system generates a payment confirmation email which includes relevant details such as the booking reference number, payment amount, and trip details and a thank you message
			Step 5: The system sends the payment confirmation email to the client's email address.
ALTERNATE COURSES:	ALT-STEP 1: If the payment processing encounters errors, the system sends an error notification to the client instead of a payment confirmation email.		
CONCLUSION:	The use case concludes when the payment confirmation email has been sent to the client's email address, providing them with a record of the payment and details about their booking.		
POST-CONDITION:	The payment confirmation email has been sent to the client, and the client received the confirmation.		
BUSINESS RULES	• Only one email is sent to the client after each successful transaction. • The payment confirmation email should be sent immediately upon successful payment processing.		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• NONE		
ASSUMPTIONS:	• NONE		
OPEN ISSUES:	NONE		

{AFRICAN ADVENTURE}

Author (s): **M.SADIKI** **Date:** **22/10/2023**

Version: _____

USE CASE NAME:	Send payment reminder	**USE CASE TYPE** Business Requirements: ☐ **System Analysis:** ☐ System Design: √
USE CASE ID:	1.7	
PRIORITY:	High	
SOURCE:	African Adventures case study	
PRIMARY BUSINESS ACTOR	Time	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• Client	
DESCRIPTION:	This use case describes an event where the system wants to send payment reminders to clients who have paid only 50% deposit for their booking to warn them about forfeiting their deposit amount if they fail to make their final payment before the closing date.	
PRE-CONDITION:	The clients made a booking and paid only 50% deposit amount. The client's booking status is "Booked"	
TRIGGER:	The due date for full payment is approaching.	

TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1: The due date for full payment is approaching		Step 2: The system identifies clients who have made only a partial deposit for their booking and have a remaining balance to pay by their booking status which is "paid" in the African adventures database using SQL queries

			Step 3: The system retrieves the client's contact information, including their email address, from the booking entity (invoke use case 1.1 Register client)
			Step 4: Using the client's email address, the system generates a payment reminder email.
			Sep 5: The payment reminder email includes details about the outstanding balance, the due date for full payment, a link to the payment portal, and a warning message encouraging clients to finalize their payment so that they don't lose their deposit amount.
			Step 6: The system schedules the sending of the payment reminder email to ensure it is delivered at an appropriate time, such as a week prior to the due date.
			Step 7: The system sends the payment reminder to the client email address.
ALTERNATE COURSES:	ALT-Step 6:If a client completes the full payment before the reminder email is scheduled to be sent, the use case terminates.		
CONCLUSION:	The payment reminder email has been scheduled and sent to clients with outstanding balances to prompt them to complete their payments.		
POST-CONDITION:	Client with outstanding balance will receive the payment reminder email.		
BUSINESS RULES	• If clients fail to make their final payment, they forfeit the deposit amount.		

IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	None

{African Adventures}

Author (s): **M.SADIKI** **Date:** **2023/10/22**

Version: _____

USE CASE NAME:	Cancel booking	**USE CASE TYPE**	
USE CASE ID:	1.8	Business Requirements:	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	African Adventures case study	System Design:	√
PRIMARY BUSINESS ACTOR	Time		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• Client		
DESCRIPTION:	This use case describes the event when the client wants to cancel their booking whereby the cancellation can be classified as "fully refunded" if the client cancels their booking before closing time and "No refund" for clients who do not cancel on time.		
PRE-CONDITION:	The client has an active booking and wishes to cancel it.		
TRIGGER:	The client initiates the cancellation request.		

TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1: The client logs into their account on the African Adventures website using their credentials.		
	Step 2: The client navigates to the "Online Booking" section of their account.		
			Step 3: The system provides the client with various options to select from in the online booking function

	Step 4: The client selects the booking they want to cancel and clicks on the "Cancel" option.		.
			Step 5: The client is prompted to provide a reason for the cancellation, which they enter in a text box.
	Step 6: The client provides their reason for cancellation and click on the submit button.		
			Step 7: The system displays a message box together with two buttons to the client asking them if they are sure they want to cancel their booking. Buttons: 1.Cancel 2.Confirm
	Step 7: The client clicks the “Confirm” button which confirms that they are sure they want to continue with cancellation.		
			Step 8: The system records the cancellation request, associating it with the client's booking.
			Step 9: The system checks the cancellation policy to determine whether the booking is eligible for a refund or not based on the timing of the cancellation.
			Step 10: If the booking is eligible for a refund, the system initiates the refund process.
			Step 11: The system updates the booking status for the client in the

			database to “Fully Refunded” [ALT]
			Step 12: The system will show the client on the screen that the cancellation was successful.
ALTERNATE COURSES:	Alt-step 11: if the client cancels their booking after the closing time for refund their booking status will be updated to “Cancelled/No refund” after which the use case terminates.		
CONCLUSION:	The use case concludes when the booking was successfully cancelled, and the booking status is updated.		
POST-CONDITION:	The use case concludes when the booking was successfully cancelled, and the booking status is updated.		
BUSINESS RULES	• Clients who fail to request cancellation of their booking two weeks in advance they forfeit 50% deposit amount of their money.		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None		
ASSUMPTIONS:	• None		
OPEN ISSUES:	None		

{African Adventures}

Author (s): **M.SADIKI** **Date:** **22/10/2023**

Version: _____

USE CASE NAME:	Send cancellation confirmation	**USE CASE TYPE**	
USE CASE ID:	1.9	Business Requirements:	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	African adventures case study	System Design:	√
PRIMARY BUSINESS ACTOR	Time		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• Client		
DESCRIPTION:	This use case describes an event when the client has successfully cancelled their booking and the system generates and send them the cancellation confirmation email, providing them with information about the cancellation and any applicable refunds.		
PRE-CONDITION:	The client initiated the cancellation process/request.		
TRIGGER:	The booking has been successfully cancelled from the system.		

TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
			Step 1: Upon receiving the cancellation request from the client, the system processes the cancellation and determines whether a refund is applicable based on the cancellation policy and timing.
			Step 2: If the booking is eligible for a refund, the

			system calculates the refund amount, if any, and prepares the notification content accordingly.
			Step 3: The system generates a cancellation notification email, which includes details such as the cancellation confirmation, refund amount (if applicable), and instructions on how the refund will be processed.[ALT]
			Step 4: The system identifies the client associated with the canceled booking in the database using select statements and retrieves their contact information, including their email address.
			Step 5: Using the client's email address, the system sends the cancellation notification email to the client.
ALTERNATE COURSES:	Alt-Step3:If the booking is not eligible for a refund, the system prepares the notification email to inform the client that the booking has been canceled and no refund will be processed.		
CONCLUSION:	The use case concludes when the detailed cancellation email notification is sent successfully to the client.		
POST-CONDITION:	The email is successfully sent to the client and the client is informed about the refund information		
BUSINESS RULES	• Cancellation notifications should be sent promptly after a booking is canceled. • Refund processing must be clear, and the company must be informative regarding the email content.		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None		
ASSUMPTIONS:	• None		
OPEN ISSUES:	None		

{African adventures}

Author (s): **M.SADIKI** **Date:** **22/10/2023**

Version: _____

USE CASE NAME:	Check-in for flight	**USE CASE TYPE**	
USE CASE ID:	2.1	Business Requirements:	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	African adventures case study	System Design:	√
PRIMARY BUSINESS ACTOR	Passenger		
PRIMARY SYSTEM ACTOR	Office Administrator		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case describes an event when the passenger wants to check-in for a flight on the day of the trip, the passenger will be assisted by the office administrator who will be interacting with the system on behalf of the passenger to ensure that they successfully check-in.		
PRE-CONDITION:	The office Administrator must be logged not the system and the passenger must have an active booking with their status being "paid".		
TRIGGER:	The passenger arrives at the departure location and is ready to board the flight.		

TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1: The passenger wants to check-in for a flight	Step 2: The office administrator logs into the system using their credentials.	Step 3: The Office Admin navigates to the admin section which has various options to choose from.
			Step 4: The office Admin clicks on the check in

			option on the Administration function.
			Step 5: the system prompts the office admin to provide the passenger booking reference number which will be entered on text box.
		Step 6: The office admin requests the client to provide their booking reference number.	
	Step 7: The client provides the requested booking reference number.		Step 8: The office admin enters their client reference number on the text box and click on the Proceed button.
			Step 9: The system validates the information provided by the office administrator to locate the passenger's booking information.
			Step 10: The system verifies the passenger's booking status in the database to see if they are eligible to proceed with checking in (The booking status must be “Paid”) which Confirms that the passenger paid the full amount for the trip.[ALT]
			Step 11: The system records the check-in status for the passenger as "checked in." in the database and display a successful check-in message on the screen.
		Step 12: The office admin informs the passenger that they have been checked in successfully.	

ALTERNATE COURSES:	Alt-Step 10: if the passenger status is still “Booked” which implies that they did not make their full payment the office admin will inform the passenger that they cannot proceed to check-in because they did not make the full payment on time then the use case will terminate.
CONCLUSION:	The use case concludes when the passenger is informed about their successful check-in by the office administrator.
POST-CONDITION:	The check-in is successful. The indemnity form is generated for the client to complete.
BUSINESS RULES	• Passengers should check in within the designated check-in time frame. • The passengers must have a valid booking reference number. • Passengers must make a full payment before checking in.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	None

{AFRICAN ADVENTURE SYSTEM}

Author (s): **BASETSANA SEKHOTO** **Date:** **24/10/2-13**

Version: **1**

USE CASE NAME:	Generate indemnity form	**USE CASE TYPE**	
USE CASE ID:	2.2	Business Requirements:	□
PRIORITY:	High	**System Analysis:**	□
SOURCE:	Case Study	System Design:	√
PRIMARY BUSINESS ACTOR	Passenger		
PRIMARY SYSTEM ACTOR	Office Administrator		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case starts when the passenger has checked and verified that the passenger has booked and paid for the trip using a reference number. The system has generated an indemnity form and generates a ticket which contains the Ballon number and pilot information which the passenger signs and hands it back to the office administrator. This use case concludes when the ticket is generated and handed to the passenger after they have signed the indemnity form.		
PRE-CONDITION:	The office administrator is logged onto the system		
TRIGGER:	The passenger wants to check in.		
TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response: Manual Action (PSA)	System Response: Automated Action (System)
	Step 1: The passenger wants to check-in. [alt]		Step 2: Using C# MVC, the generate indemnity form screen with the following controls: **Buttons** • Home • Online Booking • Administration

			• Balloon System • Back • Next • Generate indemnity form • Navigation **Labels** • Check in
		Step 3: The office administrator clicks the Generate indemnity form	**Step 4:** The system generates the indemnity form.
			Step 5: Using MVC C#, the system displays the indemnity form on the screen
			Step 6: The indemnity form is printed by the system.
		Step 7: The office administrator asks the passenger to fill in the indemnity form	
	Step 8: The passenger fills in the indemnity form.	**Step 9:** The office administrator takes the completed indemnity form from the passenger.	**Step 10**: Using C# MVC, the system displays a messagebox with the following controls. **Buttons:** Generate ticket Generate indemnity form OK **MessageBox:** "You have successfully generated ticket"
		Step 11: The office administrator clicks ok button	**Step 12:** Using MVC, the system generates the ticket.

			Step 13: Using MVC, the system prints out the tickets and displays a messagebox with the following controls **Buttons:** ● OK ● Home ● Online Booking ● Administration ● Balloon System ● Back ● Next **Message** ● Ticket generated for "Client Name" Navigation bar
		Step 14: The office administrator clicks OK and hands the ticket to the passenger.	**Step 15:** The system displays the check in scream with the following controls: Buttons ● Check-in button ● Confirm check in ● Cancel check in ● Home ● Online Booking ● Administration ● Balloon System ● Back ● Next Labels ● Check in for flight ● Enter clients booking reference number ● Admin sign in ● Client check in status ● Check-in table.
	Step 16: The passenger accepts the tickets and the indemnity form has been generated.		
ALTERNATE COURSES:	**Alt Step 1:** The office administrator does not check the customer in due to failure to provide the reference number.		

CONCLUSION	● The passenger received a ticket. ● The indemnity form is signed by the passenger. ● The passenger is checked for their trip.
POST-CONDITION:	The indemnity form is generated.
BUSINESS RULES	● Passenger must provide their booking reference number before checking in ● The indemnity form must be signed before the ticket is generated and given to the passenger.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	● Load shedding ● System offline
ASSUMPTIONS:	● None
OPEN ISSUES:	● None

{THE AFRICAN ADVENTURE SYSTEM}

Author (s): **BASETSANA SEKHOTO** **Date:** **24/10/2023**

Version: **1**

USE CASE NAME:	Generate ticket	**USE CASE TYPE**	
USE CASE ID:	2.3	Business Requirements:	
PRIORITY:	High	**System Analysis:**	
SOURCE:	Case Study	System Design:	√
PRIMARY BUSINESS ACTOR	Passenger		
PRIMARY SYSTEM ACTOR	Office Administrator		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case starts when the passenger has checked and verified that the passenger has booked and paid for the trip using a reference number. The system has generated an indemnity form and generates a ticket which contains the Ballon number and pilot information which the passenger signs and hands it back to the office administrator. This use case concludes when the ticket is generated and handed to the passenger after they have signed the indemnity form.		
PRE-CONDITION:	The office administrator is logged onto the system		
TRIGGER:	The passenger wants to check in.		
TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1: The passenger wants to check-in. [alt]		Step 2: Using C# MVC, the generate indemnity form screen with the following controls: **Buttons**

			● Home ● Online Booking ● Administration ● Balloon System ● Back ● Next ● Generate indemnity form ● Navigation **Labels** ● Check in
		Step 3: The office administrator clicks the Generate indemnity form	**Step 4:** The system generates the indemnity form.
			Step 5: Using MVC C#, the system displays the indemnity form on the screen
			Step 6: The indemnity form is printed by the system.
		Step 7: The office administrator asks the passenger to fill in the indemnity form	
	Step 8: The passenger fills in the indemnity form.	**Step 9:** The office administrator takes the completed indemnity form from the passenger.	**Step 10**: Using C# MVC, the system displays a messagebox with the following controls. **Buttons:** Generate ticket Generate indemnity form OK **MessageBox:** "You have successfully generated ticket"
		Step 11: The office administrator clicks ok button	**Step 12:** Using MVC, the system generates the ticket.
			Step 13: Using MVC, the system prints out the

			tickets and displays a messagebox with the following controls **Buttons:** ● OK ● Home ● Online Booking ● Administration ● Balloon System ● Back ● Next **Message** ● Ticket generated for "Client Name" Navigation bar
		Step 14: The office administrator clicks OK and hands the ticket to the passenger.	**Step 15:** The system displays the check in scream with the following controls: Buttons ● Check-in button ● Confirm check in ● Cancel check in ● Home ● Online Booking ● Administration ● Balloon System ● Back ● Next Labels ● Check in for flight ● Enter clients booking reference number ● Admin sign in ● Client check in status ● Check-in table.

	Step 16: The passenger accepts the tickets and the indemnity form has been generated.		
ALTERNATE COURSES:	**Alt Step 1:** The office administrator does not check the customer in due to failure to provide the reference number.		
CONCLUSION	● The passenger received a ticket. ● The indemnity form is signed by the passenger. ● The passenger is checked for their trip.		
POST-CONDITION:	The indemnity form is generated.		
BUSINESS RULES	● Passenger must provide their booking reference number before checking in ● The indemnity form must be signed before the ticket is generated and given to the passenger.		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	● Load shedding ● System offline		
ASSUMPTIONS:	● None		
OPEN ISSUES:	● None		

{THE AFRICAN ADVENTURE SYSTEM}

Author (s): **BASETSANA SEKHOTO_** **Date:** **24/10/2023**

Version: **1**

USE CASE NAME:	Refund booking	**USE CASE TYPE**	
USE CASE ID:	2.4	Business Requirements:	
PRIORITY:	High	**System Analysis:**	
SOURCE:	Case Study	System Design:	√
PRIMARY BUSINESS ACTOR	Client		
PRIMARY SYSTEM ACTOR	Office Administrator		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• Owner		
DESCRIPTION:	This use case starts when a client wants to cancel their booking for a trip. The client provides their reference number which is used to locate their order and the order is canceled by the office administrator and the booking list is updated. The system makes the transfer of the refund money to the client's account.A cancellation notification is sent to the client. This use case concludes when the client has received an email notifying them about their order cancellation.		
PRE-CONDITION:	The office administrator must be logged in onto the system.		
TRIGGER:	The clients want to cancel their booking.		
TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1: The clients want to cancel their booking.		
			Step 2: Using C# MVC, the system already has the refund screen on display with the following controls: Search feature

			Navigation bar **Buttons**: ● Home ● Online Booking ● Administration ● Balloon System ● Back ● Next Search bar for searching the passenger using their reference number.
	Step 3: The client enters the reference number onto the search bar		**Step 4:** Using **SQL SELECT statement**, the system retrieves the client's booking information from the **booking table**. [ALT]
	.		**Step 5:** Using C# MVC, the system displays the cancel booking screen on display with the following controls: Navigation bar **Buttons**: ● Home ● Online Booking ● Administration ● Balloon System ● Confirm checkin ● Cancel check in ● Check in button schedule ● Client check in status table with passengers check in details ● Textbox **Labels** ● Enter client booking reference number

			• Check in for flight • Signed in button
	Step 7: The client clicks on the cancel check in button		**Step 8:** Using C# MVC, the system displays the cancel booking screen on display with the following controls: Navigation bar **Buttons**: • Home • Online Booking • Administration • Balloon System • Confirm cancellation • Cancel **Labels**: • Booking information • Cancel Selected Booking Signed in icon button Message asking the client if they want to cancel the booking.
	Step 9: The client clicks on the confirm cancellation button.		**Step 10:** Using **SQL UPDATE** statement, the system edits the booking status in the **booking table** for the client to CANCELLED.
			Step 11: Using C# MVC, the system displays the cancel booking screen on display with the following controls: Navigation bar **Buttons**: • Home • Online Booking • Administration

			• Balloon Services • Signed in icon • Back to my bookings • Email **Message boxes** - A message box that informs the client that they have successfully canceled their booking and that their money have been refunded into their account within a few days - Another messagebox with an email message notifying the client that they have successfully cancelled their booking.
	Step 12: The client clicks the Back to my bookings buttons		
ALTERNATE COURSES:	**Alt step 4:** The system fails to locate the client's booking details using a select statement.Go to step 2. Alt step 4b: The system displays a message box notifying the client that their booking details were not found and that the reference number is incorrect.		

CONCLUSION:	This use case concludes when the client has received an email notifying them that they have successfully canceled their booking and have received the refund money into their bank account
POST-CONDITION:	● Cancellation email is sent to the client. ● Booking is canceled. ● The booking list is updated. ● The client has been refunded.
BUSINESS RULES	● The client must provide their reference number. ● A client can only cancel a booking two weeks prior to the trip. ● If a client cancels after two weeks period the deposit is forfeited however, any additional payments will be refunded.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	● Load Shedding ● The system is down. ● The system of the bank that African Adventures uses is down.
ASSUMPTIONS:	● None
OPEN ISSUES:	● None

{THE AFRICAN ADVENTURE SYSTEM}

Author (s): **BASETSANA SEKHOTO** **Date:** **24/10/2023**

Version: **1**

USE CASE NAME:	Set up flight schedule	**USE CASE TYPE** Business Requirements: □ **System Analysis:** □ System Design: √
USE CASE ID:	2.5	
PRIORITY:	High	
SOURCE:	Case Study	
PRIMARY BUSINESS ACTOR	Office Administrator	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	• Owner	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	This use case starts when the office administrator wants to set up a flight schedule. The office administrator adds all the time slots and trips needed on the schedule. The system captures the new flight schedule with the time slots and flights and trips. The office administrator confirms the flight schedule. This use case concludes when the new flight schedule now exists on the system and can now be used for trip bookings.	
PRE-CONDITION:	The office administrator is logged onto the system.	
TRIGGER:	The office administrator wants to set up a flight schedule.	

TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
		Step 1: The office administrator wants to set up a flight schedule.	**Step 2:** Using C# MVC, the system already has the flight schedule screen on display with the following controls: Search feature Navigation bar

			Buttons: ● Home ● Online Booking ● Administration ● Balloon System ● Back ● Next ● Set Up schedule ● Cancel ● Add ● Delete **Labels** ● Flight Schedule **Tabs** ● Trips ● Pilots **Trips Table with the following details:** ● Departure ● Client Name and Surname ● Balloon number ● Pilot assigned ● Lunch basket number
		Step 3: The office administrator clicks on the set up schedule button.	**Step 4:** Using C# MVC, the system already has the flight schedule screen on display with the following controls: **Navigation bar** **Buttons**: ● Home ● Online Booking ● Administration ● Balloon System ● Back ● Next ● Cancel ● Add **Labels**: ● Client ● Departure

			● Balloon number ● Pilot ● Lunch basket Tabs ● Pilots ● Trips
		Step 5: The administrator enters the new flight schedule details on the screen and clicks add.	**Step 6:** The system captures the new flight schedule with the time slots and flights and trips.
			Step 7: Using SQL the system uses the **INSERT SQL statement** to add the new flight schedule details on **flight schedule table**
			Step 9: Using C# MVC, the system already has the flight schedule screen on display with the following controls: Search feature Navigation bar **Buttons**: ● Home ● Online Booking ● Administration ● Balloon System ● Back ● Next ● Confirm flight schedule ● Cancel ● Add ● Delete **Labels** ● Flight Schedule **Tabs** ● Trips ● Pilots **Trips Table with the new flight schedule information with the following details:**

			• Departure • Client Name and Surname • Balloon number • Pilot assigned • Lunch basket number Meesagebox informing the administrator that they have to confirm the flight schedule.
		Step 10: The administrator clicks on the confirm button.[ALT]	**Step 11:** Using C# MVC, the system already has the flight schedule screen on display with the following controls: Search feature Navigation bar **Buttons**: • Home • Online Booking • Administration • Balloon System • Back • Next • OK • Cancel • Add • Delete **Labels** • Flight Schedule **Tabs** • Trips • Pilots **Trips Table with the new flight schedule information with the following details:** • Departure • Client Name and Surname • Balloon number • Pilot assigned • Lunch basket number

			Meesagebox informing the administrator that they have successfully created a new flight schedule.
		Step 12: The administrator clicks the OK button.	**Step 13:** The new flight schedule now exists on the system and can now be used for trip bookings.
ALTERNATE COURSES:	**Alt step 10:** The office administrator may have made a mistake on the flight schedule and chooses the option to edit the schedule. The office administrator clicks the cancel buttons and starts the whole process from scratch.		
CONCLUSION:	The new flight schedule now exists on the system.		
POST-CONDITION:	• The new schedule is created • The new schedule is displayed on the screen. • The new schedule now exists on the system		
BUSINESS RULES	• None		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• Load Shedding • System offline.		
ASSUMPTIONS:	• None		
OPEN ISSUES:	• None		

{THE AFRICAN ADVENTURE SYSTEM}

Author (s): **BASETSANA SEKHOTO** **Date:** **24/10/2023**

Version: **1**

USE CASE NAME:	Register pilot	**USE CASE TYPE**	
USE CASE ID:	2.6	Business Requirements:	
PRIORITY:	High	**System Analysis:**	
SOURCE:	Case Study	System Design:	√
PRIMARY BUSINESS ACTOR	Pilot		
PRIMARY SYSTEM ACTOR	Office Administrator		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• The Owner		
DESCRIPTION:	This use case starts when the pilot wants to be registered on the system. This use case ends when the pilot has been registered on the system.		
PRE-CONDITION:	The day is Thursday.		
TRIGGER:	Full payment is not received and the client has not paid the full payment for their order.		

TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1: Pilot wants to be registered.	**Step 2**: The office administrator asks for the pilot's information	
	Step 3: The pilot provides their details.		**Step 4:** Using C# MVC, the system already has the pilot screen on display with the following controls: Search feature

			Navigation bar **Buttons**: ● Home ● Online Booking ● Administration ● Balloon System ● Back ● Next **Labels** ● Pilot
		Step 5: The office administrator enters the pilot's details on the search bar.	Step 6: As the office administrator types pilot details, the system uses MVC to retrieve pilot details. The screen displays the following controls: **Buttons**: ● Home ● Online Booking ● Administration ● Balloon System ● Back ● Next ● Edit ● Delete ● Cancel ● Add **Labels** ● Pilot ● Pilot List ● Messagebox with a message notifying the administrator that the pilot could not be found. ● Only the existing pilot details are displayed ● MessageBox informing the office administrator

			that the pilot does not exist on the system.[ALT]
		Step 7: The office administrator clicks the add button.	**Step 9:** Using C# MVC, the system displays the pilot screen with the following controls: Search feature Navigation bar **Buttons**: ● Home ● Online Booking ● Administration ● Balloon System ● Back ● Next ● Add ● Cancel **Labels** ● Pilot registration ● Name ● Surname ● Phone ● Address ● Email ● ID Number ● Gender ● Experience ● Marital Status Textboxes X9 Tabs ● Pilots
		Step 10: The office administrator enters the pilot details and clicks the add button.	**Step 11:** Using SQL the system captures the pilot's details on the **PILOT TABLE** Using an INSERT SQL statement.
			Step 12: Using C# MVC, the system displays the pilot screen with the following controls: Search feature

			Navigation bar **Buttons**: ● Home ● Online Booking ● Administration ● Balloon System ● Back ● Next ● OK **Labels** along with the captured pilot details ● Pilot registration ● Name ● Surname ● Phone ● Address ● Email ● ID Number ● Gender ● Experience ● Marital Status **Messagebox** with notifying the administrator that the pilot has been successfully added.
		Step 13: The office administrator clicks the OK button.	**Step 14:** Using C# MVC, the system displays the pilot screen with the following controls: Search feature Navigation bar **Buttons**: ● Home ● Online Booking ● Administration ● Balloon System ● Back ● Next ● OK

			Labels along with the captured pilot details ● Pilot registration ● Name ● Surname ● Phone ● Address ● Email ● ID Number ● Gender ● Experience ● Marital Status
		Step 13: The office administrator notifies the pilot that they have been successfully registered onto the system.	
			Step 14: The use case concludes and the pilot is now registered on the system.
ALTERNATE COURSES:	**Alt step 6:** The client is registered on the system. The office administrator informs the pilot that they have already been registered on the system and asks them if they want to edit their personal information otherwise the use case terminates.		
CONCLUSION:	● Pilot is registered.		
POST-CONDITION:	● The pilot now exists on the system.		
BUSINESS RULES	● None		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	● Load shedding ● The system is offline.		
ASSUMPTIONS:	● None		
OPEN ISSUES:	● None		

{African Adventures System}

Author (s): **u21601144** **Date:** **24 October 2023**

Version: **1**

USE CASE NAME:	Assign pilot to flight	**USE CASE TYPE**	
USE CASE ID:	2.7	Business Requirements:	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	Case study	System Design:	√
PRIMARY BUSINESS ACTOR	Pilot		
PRIMARY SYSTEM ACTOR	Office administrator		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use starts when a pilot come to the office to book for their availability a month before trips, the office administrator will capture the pilot available slot on the system. The system will schedule the pilot a month in advance for their trip slot time. The system will also assign pilot a balloon with reference to which balloon has passed inspection and available at that time.		
PRE-CONDITION:	The office administrator must be logged into the system.		
TRIGGER:	The pilot wants to book their availability.		
TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1: The Pilot come into the office and request to book for their availability.	**Step 2:** The office administrator clicks on "Assign Pilot" button.	**Step 3:** The system loads the Assign Pilot screen and request the office administrator to enter the pilot details. The screen loads with the following controls.

			Navigation bar with navbar image and four buttons - African Adventures logo image - 'Home' - 'Online Booking' - 'Administration' (Active) - 'Balloon Service' Label: 'Book Pilot Availability' Group Box: 'Pilot Information' With four labels: - 'Pilot ID: - 'Name:' - 'Contact:' - 'Physical Address:' Four text boxes One button: - 'Book' Group box on the right-hand side with an image of the employee at the top Label: 'Logged- in Employee (Employee Name) (Employee ID)' Four buttons: - 'Pay Deli' - 'Reconcile Payment' - 'Weekend Schedule' - 'Assign Pilot' (Active)
		Step 4: Office administrator asks the	

		pilot to provide their details.	
	Step 5: The pilot provides their details.	**Step 6:** The office administrator captures the details on the system and clicks the 'Book' button.	**Step 7:** The system loads the Assign Pilot screen and request the office administrator to enter the pilot details. The screen loads with the following controls. Navigation bar with navbar image and four buttons - African Adventures logo image - 'Home' - 'Online Booking' - 'Administration' (Active) - 'Balloon Service' Label: 'Book Pilot Availability' Group Box: 'Booking Information' Label: '(Pilot Name) you have been booked for a flight on:' With four labels: - 'Day:' - 'Month:' - 'Year:' - 'Balloon ID:' Four text boxes (grey out) One button: - 'Confirm.' Group box on the right-hand side with an image of the employee at the top

			Label: 'Logged- in Employee (Employee Name) (Employee ID)' Four buttons: - 'Pay Deli' - 'Reconcile Payment' - 'Weekend Schedule' - 'Assign Pilot' (Active)
		Step 8: The office administrator asks for confirmation.	
	Step 9: The pilot confirms the details.	**Step 10:** Office administrator clicks the 'Confirm.' Button.	**Step 11:** The system invokes use case 3.2 Capture balloon inspection and service details. To check available balloons at that time and have passed inspection.
			Step 12: The system assigns the balloon to the pilot at their available time.
			Step 13: The system saves and updates the pilot availability schedule.
			Step 14: The system loads the Assign Pilot screen and request the office administrator to enter the pilot details. The screen loads with the following controls. Navigation bar with navbar image and four buttons - African Adventures logo image - 'Home'

			- 'Online Booking' - 'Administration' (Active) - 'Balloon Service' Label: 'Book Pilot Availability' Group Box: 'Booking Information' Label: 'Booking successful' With five labels: - 'Pilot Name' - 'Day:' - 'Month:' - 'Year:' - 'Balloon ID:' Five text boxes (grey out) One button: - 'OK.' Group box on the right-hand side with an image of the employee at the top Label: 'Logged- in Employee (Employee Name) (Employee ID)' Four buttons: - 'Pay Deli' - 'Reconcile Payment' - 'Weekend Schedule' - 'Assign Pilot' (Active)
		Step 15: Office administrator clicks on the 'OK' button.	**Step 16:** The system returns to the home page with Group box on the right-hand side with an image of the employee at the top.

<table>
<tr><td rowspan="3"></td><td></td><td></td><td>Label: ‘Logged- in Employee (Employee Name) (Employee ID)’

Four buttons:

- ‘Pay Deli’
- ‘Reconcile Payment’
- ‘Weekend Schedule’
- ‘Assign Pilot’</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="3">Alt-Step 9: The pilot declines the booked time slot, and the system loads the Assign Pilot screen and request the office administrator to enter the pilot details.

The screen loads with the following controls.

Navigation bar with navbar image and four buttons

- African Adventures logo image
- ‘Home’
- ‘Online Booking’
- ‘Administration’(Active)
- ‘Balloon Service’
Label: ‘Book Pilot Availability’

Group Box: ‘Pilot Information’

With four labels:

- ‘Pilot ID:
- ‘Name:’
- ‘Contact:’
- ‘Physical Address:’
Four text boxes

One button:

- ‘Book’
Group box on the right-hand side with an image of the employee at the top

Label: ‘Logged- in Employee (Employee Name) (Employee ID)’

Four buttons:

- ‘Pay Deli’
- ‘Reconcile Payment’
- ‘Weekend Schedule’
- ‘Assign Pilot’ (Active)</td></tr>
</table>

	Alt-Step 11: There are no available balloons, the system will assign the pilot a balloon whenever they are available, the system loads the Assign Pilot screen and request the office administrator to enter the pilot details. The screen loads with the following controls. Navigation bar with navbar image and four buttons - African Adventures logo image - 'Home' - 'Online Booking' - 'Administration'(Active) - 'Balloon Service' Label: 'Book Pilot Availability' Group Box: 'Booking Information' Label: 'Booking successful' With four labels: - 'Pilot Name' - 'Day:' - 'Month:' - 'Year:' Four text boxes (grey out) One button: - 'OK.' Group box on the right-hand side with an image of the employee at the top Label: 'Logged- in Employee (Employee Name) (Employee ID)' Four buttons: - 'Pay Deli' - 'Reconcile Payment' - 'Weekend Schedule' - 'Assign Pilot' (Active)
CONCLUSION:	The use case concludes when the pilot receives a confirmation that there booking have been placed successfully.
POST-CONDITION:	Pilot's booking has been captured on the system.
BUSINESS RULES	• Pilots should indicate their availability a month in advance.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• Whenever the booked pilot is unavailable will be able to be replaced.

OPEN ISSUES:	This might cause trouble to book pilots a month before their trips, as it is far maybe a bet sooner to their trips.

{African Adventures System}

Author (s): **u21601144** **Date:** **24 October 2023**

Version: **1**

USE CASE NAME:	Generate weekend trip schedule	**USE CASE TYPE**
USE CASE ID:	2.8	Business Requirements: ☐
PRIORITY:	High	**System Analysis:** ☐
SOURCE:	Case Study	System Design: √
PRIMARY BUSINESS ACTOR	Time	
PRIMARY SYSTEM ACTOR	Office administrator	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	This use case starts every week Friday to make weekends trip schedules. The system will get all the trips booked for that week's weekend and make a list out of it, including which client will be on which trip, a pilot scheduled for a certain balloon to operate. If there are any lunch basket for a specific trip the system will also include this on the list.	
PRE-CONDITION:	It must be Friday.	
TRIGGER:	It is Friday and there are booked trips for the weekend.	

TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
		Step 1: Office administrator clicks on the 'Weekend Schedule' button.	**Step 2:** The system searches for a list of all the trips for weekend.

			Step 3: The system will search a list of clients that booked a trip for the weekend.
			Step 4: The system searches all the pilots assigned for the trip and their assigned balloon.
			Step 5: The system checks if a lunch basket was included or not.
			Step 6: The system saves the information.
			Step 7: The system loads the Weekend schedule screen. The screen loads with the following controls. Navigation bar with navbar image and four buttons - African Adventures logo image - 'Home' - 'Online Booking' - 'Administration' (Active) - 'Balloon Service' Group Box: 'Weekend Schedule' With a table: With columns: - 'Client' - 'Pilot Name' - 'Balloon ID' - 'Lunch Basket' One button: - 'Print' Group box on the right-hand side with an image

			of the employee at the top Label: 'Logged- in Employee (Employee Name) (Employee ID)' Four buttons: - 'Pay Deli' - 'Reconcile Payment' - 'Weekend Schedule' (Active) - 'Assign Pilot'
		Step 8: The office administrator clicks on the 'Print' button.	**Step 9:** The system returns to the home page with Group box on the right-hand side with an image of the employee at the top. Label: 'Logged- in Employee (Employee Name) (Employee ID)' Four buttons: - 'Pay Deli' - 'Reconcile Payment' - 'Weekend Schedule' - 'Assign Pilot'
ALTERNATE COURSES:			
	Alt-Step 5: The are no booked lunch baskets, process continues to step 6 to save the information.		
CONCLUSION:	The use case concludes when a list of all the trips have been made, composed of all the clints who will be on the trip, pilots scheduled to operate a specific balloon and if there are any lunch basket they are included.		
POST-CONDITION:	The system has made trip schedule for the weekend.		
BUSINESS RULES	• It must be Friday. • A passenger must have booked and paid for their trip. • The balloon must have passed all the inspections. • Pilots should be available.		

IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	The schedule should be sent to every pilot and office administrator.

{African Adventures System}

Author (s): **u21601144** **Date:** **24 October 2023**

Version: **1**

USE CASE NAME:	Place deli order	**USE CASE TYPE**	
USE CASE ID:	2.9	Business Requirements:	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	Case study	System Design:	√
PRIMARY BUSINESS ACTOR	Office administrator		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	• Local deli (ERA)		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case starts every Friday morning, the system will get list/number of lunch basket booked for the weekend. The system will use this to place an order with local deli, for which they will prepare the order.		
PRE-CONDITION:	It must be Friday morning.		
TRIGGER:	It is Friday morning and there are included lunch basket on weekends trips.		

TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
		Step 1: The office administrator clicks on the 'Place order' button.	**Step 2**: The system checks for order placed within that month.

			Step 3: The system loads the Place order screen. The screen loads with the following controls. Navigation bar with navbar image and four buttons - African Adventures logo image - 'Home' - 'Online Booking' - 'Administration' (Active) - 'Balloon Service' Group Box: 'Place order' With a table: With columns: - 'Order ID' - 'Client ID' - 'Description' - 'Price' One button: - 'Place order.' Group box on the right-hand side with an image of the employee at the top Label: 'Logged- in Employee (Employee Name) (Employee ID)' Four buttons: - 'Pay Deli' (Active) - 'Reconcile Payment' - 'Weekend Schedule' - 'Assign Pilot'

		Step 4: The office administrator clicks on the 'Place order' button.	**Step 5:** The system loads the Place order screen. The screen loads with the following controls. Navigation bar with navbar image and four buttons - African Adventures logo image - 'Home' - 'Online Booking' - 'Administration' (Active) - 'Balloon Service' Group Box: 'Place order' Label: 'You have successfully placed order with an overall total of [Total for order]. For orders placed during the month of [Month of the order and the year].' One button: - 'OK'
		Step 6: The office administrator clicks on the 'OK' button.	**Step 7:** The system returns to the home page with Group box on the right-hand side with an image of the employee at the top. Label: 'Logged- in Employee (Employee Name) (Employee ID)' Four buttons: - 'Pay Deli' - 'Reconcile Payment'

			- 'Weekend Schedule' 'Assign Pilot'
ALTERNATE COURSES:	Alt-Step 2: The are no order, so the process is terminated. Home page is displayed with Group box on the right-hand side with an image of the employee at the top. Label: 'Logged- in Employee (Employee Name) (Employee ID)' Four buttons: - 'Pay Deli' - 'Reconcile Payment' - 'Weekend Schedule' 'Assign Pilot'		
CONCLUSION:	This use case concludes when the system sends though the preparation order to local deli and placed order details are stored on the system.		
POST-CONDITION:	- The order has been placed. - The local deli receives the order details.		
BUSINESS RULES	• A passenger must have booked and paid for their trip and included a lunch basket on their trip.		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None		
ASSUMPTIONS:	• The system only places the order on Friday.		
OPEN ISSUES:	The System should send office administrator a confirmation message/emails to state that the order has been placed.		

{African Adventures System}

Author (s): **u21601144** **Date:** **24 October 2023**

Version: **1**

USE CASE NAME:	Reconcile deli payment	**USE CASE TYPE**	
USE CASE ID:	2.10	Business Requirements:	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	Case study	System Design:	√
PRIMARY BUSINESS ACTOR	Owner		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case starts when the owner wants to generate a report for all the orders places to the Local deli. The system will get all the orders places to the Local deli within that month, this will include order details and totals for the orders. The owner will use this report to reconcile with the invoices from the Local deli.		
PRE-CONDITION:	The owner must be logged into the system.		
TRIGGER:	The owner wants to generate a report of all the orders place by African Adventures to the Local deli.		

TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1: The owner clicks on the 'Reconcile Payment' button.		**Step 2:** The system checks for all the orders placed to Local deli.
			Step 3: The system retrieves all this

			information, including order details and total for each order.
			Step 4: The system compresses this information into a report.
			Step 5: The system will do the overall total for all the orders placed.
			Step 6: The system loads the Reconcile Payment screen. The screen loads with the following controls. Navigation bar with navbar image and four buttons - African Adventures logo image - 'Home' - 'Online Booking' - 'Administration' (Active) - 'Balloon Service' Group Box: 'Weekend Schedule' Label: 'Orders placed in [Month orders placed].' With a table: With columns: - 'Order ID' - 'Client ID' - 'Date' - 'Price'' One button: - 'Print' Group box on the right-hand side with an image of the employee at the top

			Label: 'Logged- in Employee (Employee Name) (Employee ID)' Four buttons: - 'Pay Deli' - 'Reconcile Payment' - 'Weekend Schedule' (Active) - 'Assign Pilot'
		Step 7: The owner clicks on the 'Print' button.	**Step 8:** The system sends the report to the owner.
			Step 9: The system returns to the home page with Group box on the right-hand side with an image of the employee at the top. Label: 'Logged- in Employee (Employee Name) (Employee ID)' Four buttons: - 'Pay Deli' - 'Reconcile Payment' - 'Weekend Schedule' 'Assign Pilot'
ALTERNATE COURSES:	Alt-Step 2: The are no order, so the process is terminated. Home page is displayed with Group box on the right-hand side with an image of the employee at the top. Label: 'Logged- in Employee (Employee Name) (Employee ID)' Four buttons: - 'Pay Deli' - 'Reconcile Payment' - 'Weekend Schedule' 'Assign Pilot'		
CONCLUSION:	This use case ends when the owner receives the report for all the orders made to Local deli.		

POST-CONDITION:	A report is sent to the owner for reconciliation.
BUSINESS RULES	• Customers should have booked lunch basket to make a report.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• This use case can only be performed by owner.
OPEN ISSUES:	None

{African Adventures System}

Author (s): **u21601144** **Date:** **24 October 2023**

Version: **1**

USE CASE NAME:	Pay deli	**USE CASE TYPE**
USE CASE ID:	2.11	Business Requirements: ☐
PRIORITY:	High	**System Analysis:** ☐
SOURCE:	Case study	System Design: √
PRIMARY BUSINESS ACTOR	Owner	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	• Bank (ESA)	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	This use case starts when African Adventure receives a consolidated invoice from the local deli to pay their monthly orders. The owner will need to make sure that the invoice sent matches with the orders made. After confirmation the owner can start to make the transaction.	
PRE-CONDITION:	The office owner must be logged into the system.	
TRIGGER:	The owner wants to pay Local deli.	
TYPICAL COURSE		System Response

OF EVENTS:	Actor Action (PBA, ERA, ESA)	Manual Action (PSA)	Automated Action (System)
	Step 1: The office administrator clicks on the 'Place order' button.		**Step 2**: The system invokes use case 2.10 Reconcile deli payment.
			Step 3: The system asks the owner for confirmation with bank details.
			Step 4: The system loads the Pay deli screen. The screen loads with the following controls. Navigation bar with navbar image and four buttons - African Adventures logo image - 'Home' - 'Online Booking' - 'Administration' (Active) - 'Balloon Service' Group Box: 'Pay deli' With a table: With columns: - 'Order ID' - 'Client ID' - 'Date' - 'Price' One button: - 'Pay.' Group box on the right-hand side with an image of the employee at the top

			Label: 'Logged- in Employee (Employee Name) (Employee ID)' Four buttons: - 'Pay Deli' (Active) - 'Reconcile Payment' - 'Weekend Schedule' - 'Assign Pilot'
	Step 5: The owner clicks on the 'Pay' button.		
	Step 6: The owner confirms and enters the bank details for both African Adventures and local deli.		**Step 7**: The system verifies the bank details entered.
			Step 8: The system communicates with the African Adventures bank to verify that indeed there is available credits to perform this task.
			Step 9: The system makes payment to local deli.
			Step 10: The system informs the owner and local deli of the confirmed payment/payment made.
			Step 11: The system loads the Pay deli screen. The screen loads with the following controls. Navigation bar with navbar image and four buttons - African Adventures logo image

			- 'Home' - 'Online Booking' - 'Administration' (Active) - 'Balloon Service' Group Box: 'Pay deli' Label: 'You have successfully paid local deli with an overall total of [Total for order]. For orders placed during the month of [Month of the order and the year]. Confirmation email will be sent.' One button: - 'OK'
			Step 12: The system sends the office administrator an email 'Payment Details – Your payment has been successful. From Account: [Payee] To Account: [Recipient] Amount: [Total Amount] Date: [Date of payment]
		Step 13: The office administrator clicks on the 'OK' button.	**Step 14:** The system returns to the home page with Group box on the right-hand side with an image of the employee at the top. Label: 'Logged- in Employee (Employee Name) (Employee ID)' Four buttons: - 'Pay Deli' - 'Reconcile Payment'

			- 'Weekend Schedule' 'Assign Pilot'
ALTERNATE COURSES:	Alt-Step 2: The are no order, so the process is terminated. Home page is displayed with Group box on the right-hand side with an image of the employee at the top. Label: 'Logged- in Employee (Employee Name) (Employee ID)' Four buttons: - 'Pay Deli' - 'Reconcile Payment' - 'Weekend Schedule' 'Assign Pilot'		
CONCLUSION:	The use case concludes when the owner of African Adventures and local deli receives a confirmation of the payment made.		
POST-CONDITION:	Payment is made.		
BUSINESS RULES	• African Adventures should have placed an order to local deli.		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None		
ASSUMPTIONS:	• Payment can only be made via EFT.		
OPEN ISSUES:	None		

{African Adventures' System}

Author (s): **Busisiwe Khosa** **Date:** **23/10/2023**

Version: **01**

USE CASE NAME:	Send Newsletter	**USE CASE TYPE**	
USE CASE ID:	2.12	Business Requirements:	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	Case Study	System Design:	√
PRIMARY BUSINESS ACTOR	Owner		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case details the process where the owner of African Adventures aims to send newsletters to passengers who have previously booked a trip. The owner uses the system's search functionality by entering the keyword "Booked" in the search bar. Upon entering the keyword, the owner initiates the search, which automatically retrieves passenger records with the booking status 'Booked'. Following this, the owner adds these passengers to the 'send newsletter' list and triggers the newsletter sending process by clicking the 'send' button.		
PRE-CONDITION:	The owner is logged on to the system.		
TRIGGER:	The owner wants to send a newsletter to previously booked passengers.		

TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1: The owner wants to send a newsletter to all previously booked passengers.		
	Step 2: The Owner navigates to the newsletter management section of the system under the 'Administration' tab in the navigation bar.		
			Step 3: The system opens the newsletter

			management section screen which has the following: • Text label: 'Newsletter Management' at the top left corner right after the navigation bar and underneath it, 'Search customer'. • A Search bar • A Search button horizontally to the right after the search bar.
	Step 4: the owner enters "Booked" on the search bar and then clicks the search button.		
			Step 5: The system automatically retrieves the records of all the booked passengers and displays the following: • Text label: 'Booked Passengers', 'Send Newsletter' • A richTextBox underneath the text label containing the records of all the passengers in a table form with only three table headers reading as follows, "Passenger Id", "Full Name", "Email Address", "Booking Status",

			"Action" and all the corresponding records. • Check Box: check box on the right side of every passenger record and another one on the top right corner of the table "Select All". • Buttons: "Send" button at the bottom right of the screen. Followed by a "cancel" button. • DateTime Picker at the top right corner showing the current date.
	Step 6: The owner then selects all the passengers he/she wishes to send the newsletters to by clicking the check box (s).		
			Step 7: The system keeps track of all the passengers that were selected.
	Step 8: The owner then clicks the 'Send' button after he/she is done with the selection.		
			Step 9: The system processes the request to send the newsletter to the selected passengers.
			Step 10: System takes back to the main page.

ALTERNATE COURSES:	Alt to step 5: There are no records for 'Booked' booking status. An error message displays: No records found for booked passengers. The systems prompts the user to click the okay button,		
	Alt to step 7: The owner decides to not send the newsletter and clicks the cancel button. The system takes them back to the previous step.		
CONCLUSION:			
POST-CONDITION:	All the previously booked passengers have newsletters sent to their registered emails.		
BUSINESS RULES	•		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None		
ASSUMPTIONS:	• The owner has knowledge of who they want to send the newsletter to.		
OPEN ISSUES:			

{African Adventures' System}

Author (s): **Busisiwe Khosa** **Date:** **23/10/2023**

Version: **01**

USE CASE NAME:	Add Balloon	**USE CASE TYPE** Business Requirements: ☐ **System Analysis:** ☐ System Design: √
USE CASE ID:	3.1	
PRIORITY:	High	
SOURCE:	Case Study	
PRIMARY BUSINESS ACTOR	Administrator	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	This use case describes an instance where an administrator wants to add a new balloon to the system. The administrator will access the balloon services tab where he/she will find the form with all necessary fields to add a new balloon to the system. Once all the information is filled in, the administrator will review the details and then click the "Add" button to confirm the new added balloon details. The system will then capture the details and save them to the database.	
PRE-CONDITION:	The administrator has the necessary permission to add a new balloon to the system.	
TRIGGER:	The administrator wants to add a new balloon to the system.	

TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1: The administrator wants to add a new balloon to the system.		
	Step 2: The administrator will access the balloon services tab in the navigation bar by clicking "Balloon Services" tab.		
			Step 3: The system will automatically open the Balloon services screen which has the following:

			• Text Labels: 'Balloon services Management' at the top right corner. Other text labels: - Balloon ID - Balloon Name - Balloon registration Number. • Textboxes - Balloon ID - Balloon Name - Balloon registration Number. • A navigation bar with three tabs, "New Balloon", "Inspection log", "Reports" respectively. • Buttons: - Add - Cancel
	Step 4: The Administrator will enter the details of the balloon in the provided fields.		
	Step 5: The administrator will review the details entered and click the "Add" button to confirm the details.		
			Step 6: The system will validate the details entered and that all textboxes are completed according to the predetermined setup in the database. ALT
			Step 7: the system will automatically generate a Balloon ID by adding 1 to the previously generated Balloon ID and displaying

			it in the Balloon ID textbox.
			Step 8: The system will add the new balloon to the balloon database by use of an SQL insert query that will save the balloon in the balloon table.
			Step 9: The system will show a pop up message with the following text 'The balloon was successfully added" followed by an "Okay" button.
	Step 10: The administrator clicks on the "Okay: button.		
			Step 11: The system closes the pop up message and then redirects the user to the add new balloon form.
ALTERNATE COURSES:	Alt to step 5: The system rejects the new entered balloon details and displays a message. "Incorrect values entered, please try again" and then redirects the user to the input field that was incorrectly entered.		
CONCLUSION:	The administrator successfully added a new balloon to the system.		
POST-CONDITION:	The database is updated with correct information of the new balloon.		
BUSINESS RULES	• The system must be flexible enough to add new balloon into the system.		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None		
ASSUMPTIONS:	• The new balloon has all the required details to be captured.		
OPEN ISSUES:			

{African Adventures' System}

Author (s): **Busisiwe Khosa** **Date:** **23/10/2023**

Version: **01**

USE CASE NAME:	Capture balloon inspection and service details	**USE CASE TYPE**
USE CASE ID:	3.2	Business Requirements: ☐
PRIORITY:	High	**System Analysis:** ☐
SOURCE:	Case Study	System Design: √
PRIMARY BUSINESS ACTOR	Foreman	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	• Technician	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	This use case describes an instance when a foreman wants to capture an inspection log and balloon services on the system. The foreman will navigate to the "Balloon services" tab on the navigation bar, where he/she will be presented with an inspection log form. The foreman will then enter the relevant details and that will be saved by the system and stored in the database.	
PRE-CONDITION:	The system is operational and accessible for the foreman to use.	
TRIGGER:	The foreman wants to capture an inspection log and balloon services.	

TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1: The foreman wants to capture the inspection log and balloon services details.		
	Step 2: The foreman will navigate to the "Balloon Services" tab on the navigation bar and click on it.		
			Step 3: The system will automatically open the Balloon services screen which has the following on the "Add New

			Balloon" tab, which is the default tab: • Text Labels: 'Balloon services Management' at the top right corner. Other text labels: - Balloon ID - Balloon Name - Balloon registration Number. • Textboxes - Balloon ID - Balloon Name - Balloon registration Number. • A navigation bar with three tabs, "New Balloon", "Inspection log", "Reports" respectively. • Buttons: - Add - Cancel
	Step 4: The foreman will then navigate further to the "Inspection Log".		
			Step 5: The system will automatically open the "Inspection Log" tab which has the following: • A richTextBox with a Text label "Balloon Inspection Checklist" - Followed by text labels "Inspected by", "Date of

			inspection”, “Witness” - A table underneath it that has: - Five columns: “Inspection activities”, “Yes”, “No”, “Service Type” and “Comments” respectively. • Buttons: - Save - Cancel - Track hours
	Step 6: The foreman enters the details of the inspection and balloon services as required.		
	Step 7: The foreman will review the details entered and click the “Save” button to save the inspection log.		
			Step 7: The system will validate the details entered and save the inspection details and services to the database, by ensuring that the data conforms to the predetermined setup in the database. ALT
			Step 8: The system will automatically generate a inspection log ID by adding 1 to the previously generated inspection log ID and

			displaying it in the inspection log ID textbox.
			Step 9: The system will add the new inspection log to the inspection log database by use of an SQL insert query that will save the balloon in the inspection log table.
			Step 10: The system will show a pop up message with the following text 'The inspection log was successfully added" followed by an "Okay" button.
	Step 11: The foreman will click on the "Okay button.		
			Step 12: The system closes the pop up message and then redirects the user back to the inspection log form.
ALTERNATE COURSES:	ALT to step 7: The system rejects the new entered balloon details and displays a message. "Incorrect values entered, please try again" and then redirects the user to the input field that was incorrectly entered.		
CONCLUSION:	The inspection log is successfully added to the database.		
POST-CONDITION:	The inspection log database has up-to-date records		
BUSINESS RULES	• None		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None		
ASSUMPTIONS:	• The inspection log details for a particular balloon is already available and ready to be captured on to the system.		
OPEN ISSUES:	None		

{African adventures' System}

Author (s): **Busisiwe Khosa** **Date:** **23/10/2023**

Version: **01**

USE CASE NAME:	Track Balloon Flight Hours	**USE CASE TYPE**
USE CASE ID:	3.3	Business Requirements: ☐
PRIORITY:	High	**System Analysis:** ☐
SOURCE:	Case Study	System Design: √
PRIMARY BUSINESS ACTOR	Foreman	
PRIMARY SYSTEM ACTOR	None	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	This use case describes an instance when the foreman wants to track the number of flight hours a balloon has accumulated. The foreman will make use of the system to calculate the total hours accumulated until the specified date. The foreman will entered the required balloon details, the system will search for the balloon on the system, to validate its existence and there after track the inspection log for all the related details of the balloon and then calculate the hours it has flown based on the number of times it was assigned to a trip.	
PRE-CONDITION:	The foreman is logged on to the system.	
TRIGGER:	The foreman wants to track the balloon flight hours.	

TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1:		
	Step 1: the fore man wants to track the balloon flight hours.		
	Step 2 The foreman will navigate to the "Balloon Services" tab on the navigation bar and click on it.		
			Step 3: The system will automatically open the

			Balloon services screen which has the following on the “Add New Balloon” tab, which is the default tab: • Text Labels: ‘Balloon services Management’ at the top right corner. Other text labels: - Balloon ID - Balloon Name - Balloon registration Number. • Textboxes - Balloon ID - Balloon Name - Balloon registration Number. • A navigation bar with three tabs, “New Balloon”, “Inspection log”, “Reports” respectively. • Buttons: - Add - Cancel
	Step 4: The foreman will then navigate further to the “Inspection Log”.		
			Step 5: The system will automatically open the “Inspection Log” tab which has the following: • A richTextBox with a Text label “Balloon Inspection Checklist” • Followed by text labels “Inspected by”,

			"Date of inspection", "Witness" • A table underneath it that has: • Five columns: "Inspection activities", "Yes", "No", "Service Type" and "Comments" respectively. • Buttons: • Save • Cancel • Track hours
	Step 6: the foreman then clicks on the "Track Hours" button.		
			Step 7: The system will then open a new screen which has the following: • Text label: - "Balloon ID" - Search Balloon • Textbox -Balloon ID • Buttons: - Search - Cancel - Done
	Step 8: the foreman will enter the balloon id and then click the "Search Balloon" button.		
			Step 9: The system will use the balloon id to search for the balloon in the system, by means of an SQL select query statement that will

			retrieve the relevant data from the database. ALT
			Step 10: The system will make use of c# .net method to make a calculation of the number of hours the searched balloon has accumulated for the until to date.
			Step 11: The system will then display the number of flight hours the balloon has accumulated.
			Step 12: The system will display a pop-up message box prompting the user to the click the "Done" button.
	Step 13: The foreman will click the "Done" Button.		
			Step 14: The system will redirect the user to the inspection log screen.
ALTERNATE COURSES:	ALT to Step 9: the balloon id entered does not correspond to a valid balloon in the system. An error message pops up. "The Balloon ID you entered is invalid, please try again." Redirect to Balloon Id field.		
CONCLUSION:	The balloon flight hours have been accurately calculated.		
POST-CONDITION:	All the balloon flight hours have been updated.		
BUSINESS RULES	• None		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None		
ASSUMPTIONS:	• All balloons with inspection logs have been registered on the system,		
OPEN ISSUES:			

{African Adventures' System}

Author (s): **Busisiwe Khosa** **Date:** **23/10/2023**

Version: **01**

USE CASE NAME:	Generate Inspection Log Report	**USE CASE TYPE**	
USE CASE ID:	3.4	Business Requirements:	☐
PRIORITY:	High	**System Analysis:**	☐
SOURCE:	Case Study	System Design:	√
PRIMARY BUSINESS ACTOR	Owner		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	The use case describes an instance when the owner wants to generate a full report on the inspection of the balloons. The owner will enter the date to which s/he wants the inspection report for and click the button "Generate". The system will search on the database for the inspection logs that occurred on the specified date and display them. The owner will make a request to generate a full inspection report for the whole week.		
PRE-CONDITION:	The owner is currently logged on to the system.		
TRIGGER:	The owner wants to generate an inspection log report.		

TYPICAL COURSE OF EVENTS:	Actor Action (PBA, ERA, ESA)	System Response	
		Manual Action (PSA)	Automated Action (System)
	Step 1: The owner wants to generate an inspection log report.		
	Step 2 The foreman will navigate to the "Balloon Services" tab on the navigation bar and click on it.		
			Step 3: The system will automatically open the Balloon services screen which has the following on the "Add New

			Balloon" tab, which is the default tab: • Text Labels: 'Balloon services Management' at the top right corner. Other text labels: - Balloon ID - Balloon Name - Balloon registration Number. • Textboxes - Balloon ID - Balloon Name - Balloon registration Number. • A navigation bar with three tabs, "New Balloon", "Inspection log", "Reports" respectively. • Buttons: - Add - Cancel
	Step 4: The foreman will then navigate further to the "Reports" tab on the navigation bar.		
			Step 5: the system will open the Reports screen with the following: • Text Labels: - Date - Generate Report • Textbox - Date • Buttons: - Generate - Cancel

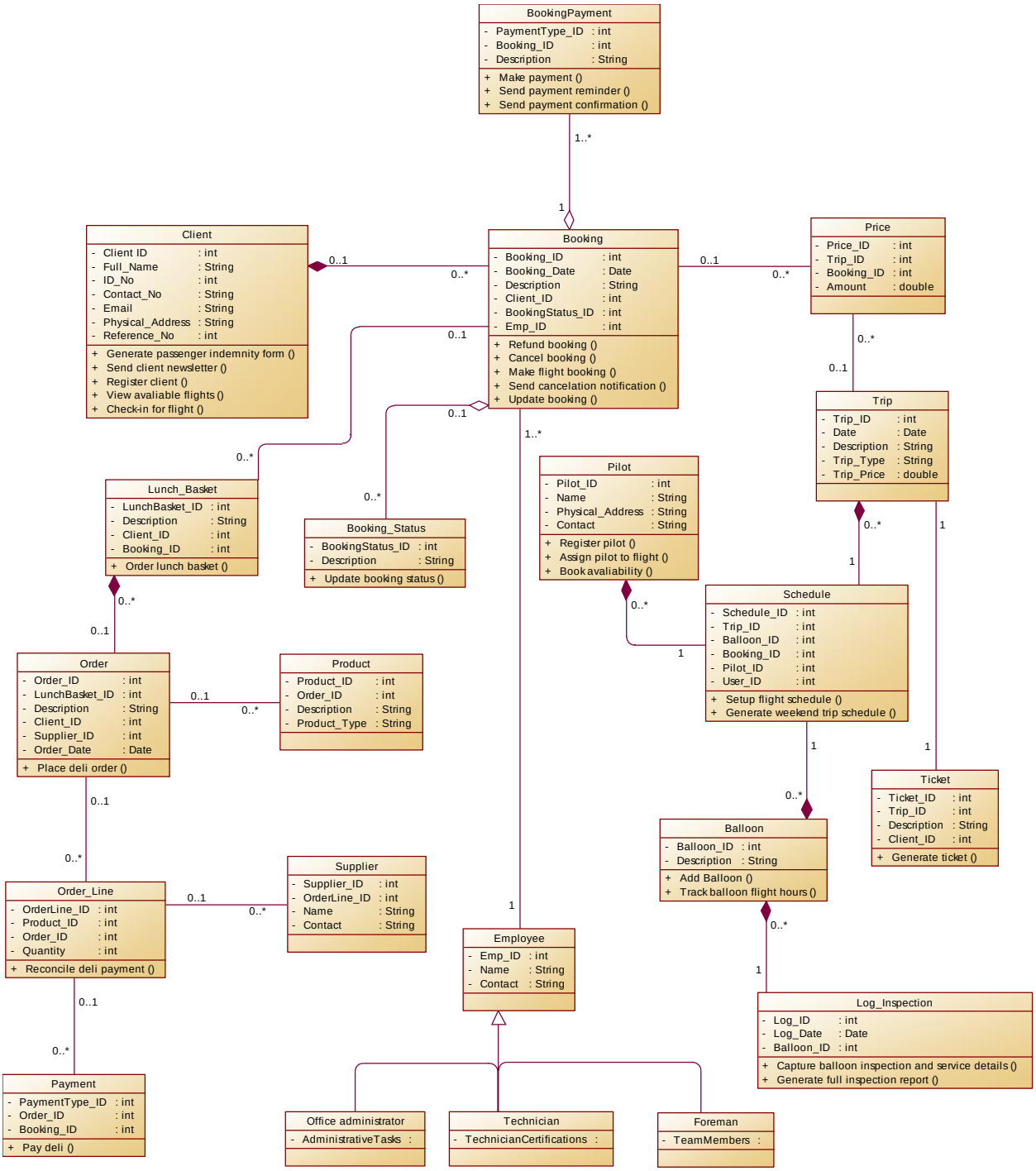
			- Done
	Step 6: The owner will enter the date of the inspection log they want to retrieve and click “Generate” button.		
			Step 7: The system will make use of a c# .net search functionality to retrieve the specified dates.
			Step 8: The system will display the summary of the inspection log.
			Step 9: The system will download report and prompt the user to click “Done” through a message dialog.
	Step 9: the owner will click “Done”		
			Step 10: The system will redirect to the main page.
ALTERNATE COURSES:			
CONCLUSION:	The inspection log report was successfully generated		
POST-CONDITION:	All the inspection reports are currently up to date.		
BUSINESS RULES	• None		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None		
ASSUMPTIONS:	• All inspection reports are captured and saved on to the system		
OPEN ISSUES:			

With this technical narrative, African Adventures gain a better understating of how their system needs to work, they can thrive though innovation and they can have a better clarity of how their system interact with external entities.

2.Object Oriented design

2.1Logical class diagram

This section primarily emphasizes the system's backend, with the goal of consolidating all the system components to illustrate their relationship and provide clear descriptions of their functions and services. In the context of African Adventures, a total of twenty-four objects have been identified.

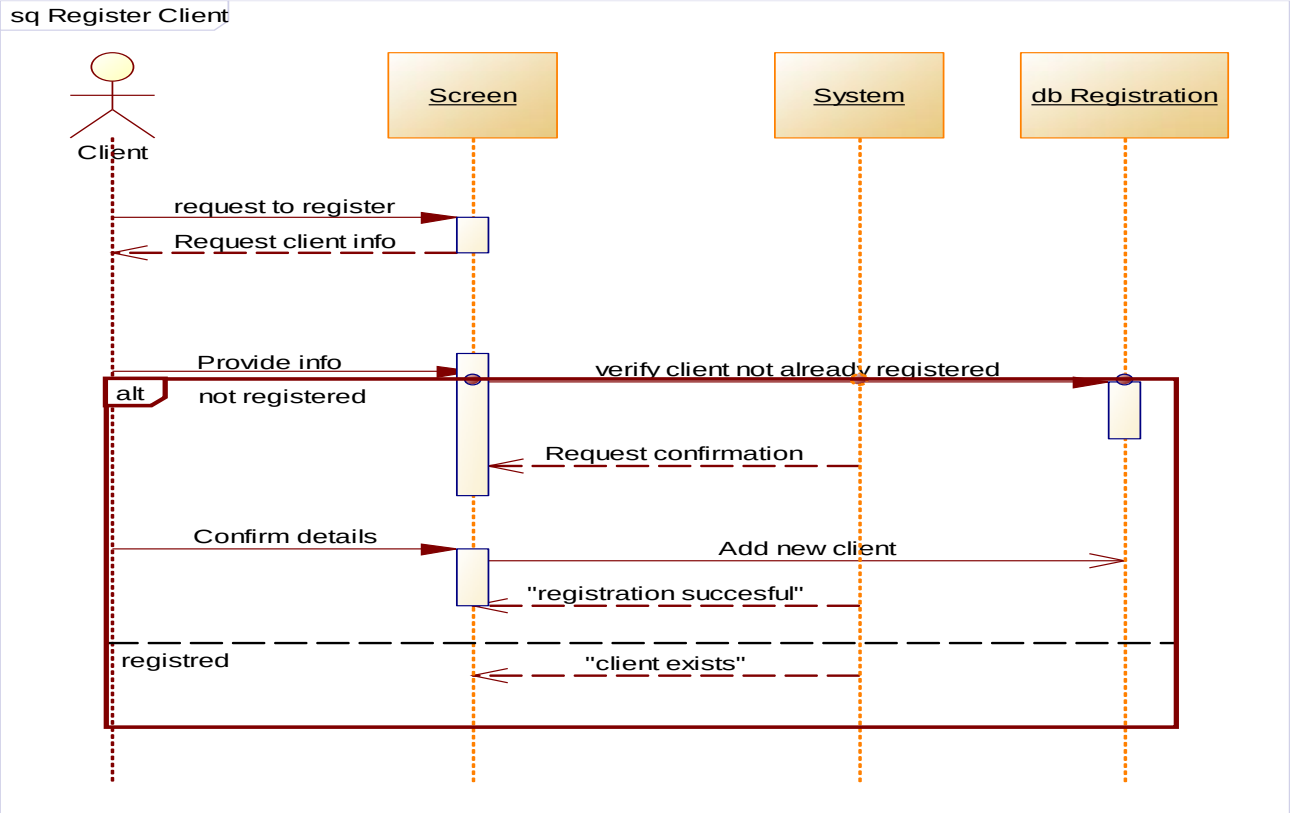

BookingPayment
- PaymentType_ID : int
- Booking_ID : int
- Description : String
+ Make payment ()
+ Send payment reminder ()
+ Send payment confirmation ()
1..*
1
Client
- Client ID : int
- Full_Name : String
- ID_No : int
- Contact_No : String
- Email : String
- Physical_Address : String
- Reference_No : int
+ Generate passenger indemnity form ()
+ Send client newsletter ()
+ Register client ()
+ View avaliable flights ()
+ Check-in for flight ()
0..1
0..*
Booking
- Booking_ID : int
- Booking_Date : Date
- Description : String
- Client_ID : int
- BookingStatus_ID : int
- Emp_ID : int
+ Refund booking ()
+ Cancel booking ()
+ Make flight booking ()
+ Send cancelation notification ()
+ Update booking ()
0..1
0..*
Price
- Price_ID : int
- Trip_ID : int
- Booking_ID : int
- Amount : double
0..*
0..1
Trip
- Trip_ID : int
- Date : Date
- Description : String
- Trip_Type : String
- Trip_Price : double
0..*
1
1
0..1
1..*
Lunch_Basket
- LunchBasket_ID : int
- Description : String
- Client_ID : int
- Booking_ID : int
+ Order lunch basket ()
0..*
0..1
Booking_Status
- BookingStatus_ID : int
- Description : String
+ Update booking status ()
0..*
Pilot
- Pilot_ID : int
- Name : String
- Physical_Address : String
- Contact : String
+ Register pilot ()
+ Assign pilot to flight ()
+ Book avaliability ()
0..*
1
Schedule
- Schedule_ID : int
- Trip_ID : int
- Balloon_ID : int
- Booking_ID : int
- Pilot_ID : int
- User_ID : int
+ Setup flight schedule ()
+ Generate weekend trip schedule ()
1
1
Order
- Order_ID : int
- LunchBasket_ID : int
- Description : String
- Client_ID : int
- Supplier_ID : int
- Order_Date : Date
+ Place deli order ()
0..1
0..*
Product
- Product_ID : int
- Order_ID : int
- Description : String
- Product_Type : String
0..1
0..*
Ticket
- Ticket_ID : int
- Trip_ID : int
- Description : String
- Client_ID : int
+ Generate ticket ()
1
0..*
Balloon
- Balloon_ID : int
- Description : String
+ Add Balloon ()
+ Track balloon flight hours ()
0..*
1
Order_Line
- OrderLine_ID : int
- Product_ID : int
- Order_ID : int
- Quantity : int
+ Reconcile deli payment ()
0..1
0..*
Supplier
- Supplier_ID : int
- OrderLine_ID : int
- Name : String
- Contact : String
1
Employee
- Emp_ID : int
- Name : String
- Contact : String
Log_Inspection
- Log_ID : int
- Log_Date : Date
- Balloon_ID : int
+ Capture balloon inspection and service details ()
+ Generate full inspection report ()
0..1
0..*
Payment
- PaymentType_ID : int
- Order_ID : int
- Booking_ID : int
+ Pay deli ()
Office administrator
- AdministrativeTasks :
Technician
- TechnicianCertifications :
Foreman
- TeamMembers :

By providing such a comprehensive description, African Adventures gain a distinct advantage in the process of designing a meticulously crafted system that functions as intended. This detailed description equips them with the essential insights and understanding required to develop a system that aligns with their objectives.

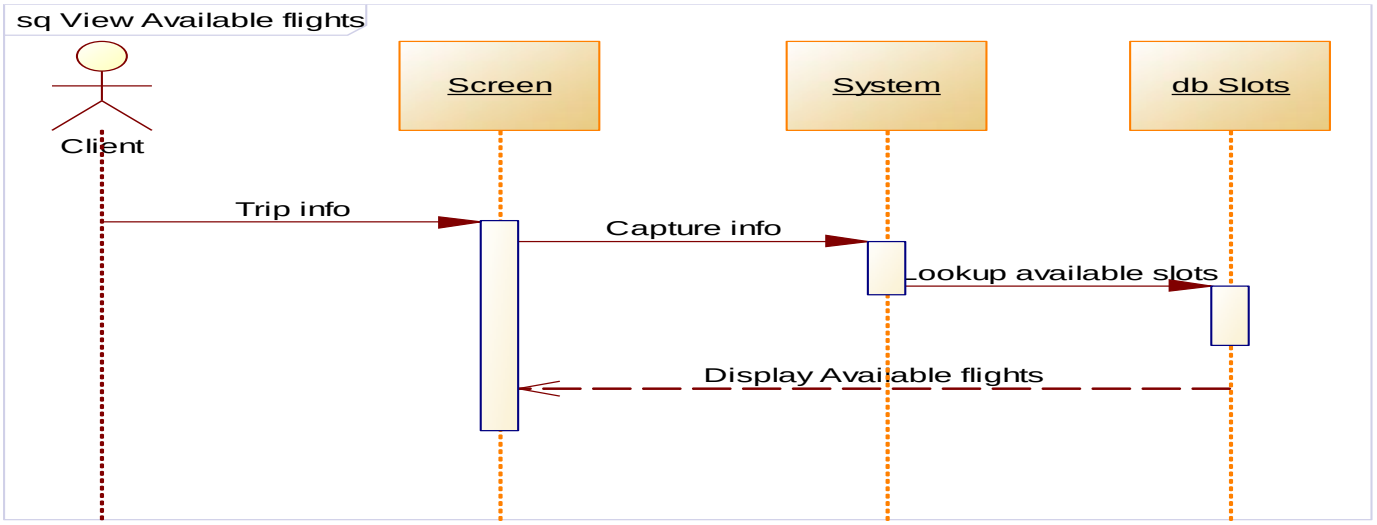
2.2 Sequence Diagrams

Sequence diagrams allow programmers and System Analysts to see the interactions that occur in a use case between actors such as users and the various objects in a system such as the screen and system server. Over the following pages we provide an in-depth analysis of Sequence diagrams for African Adventures.

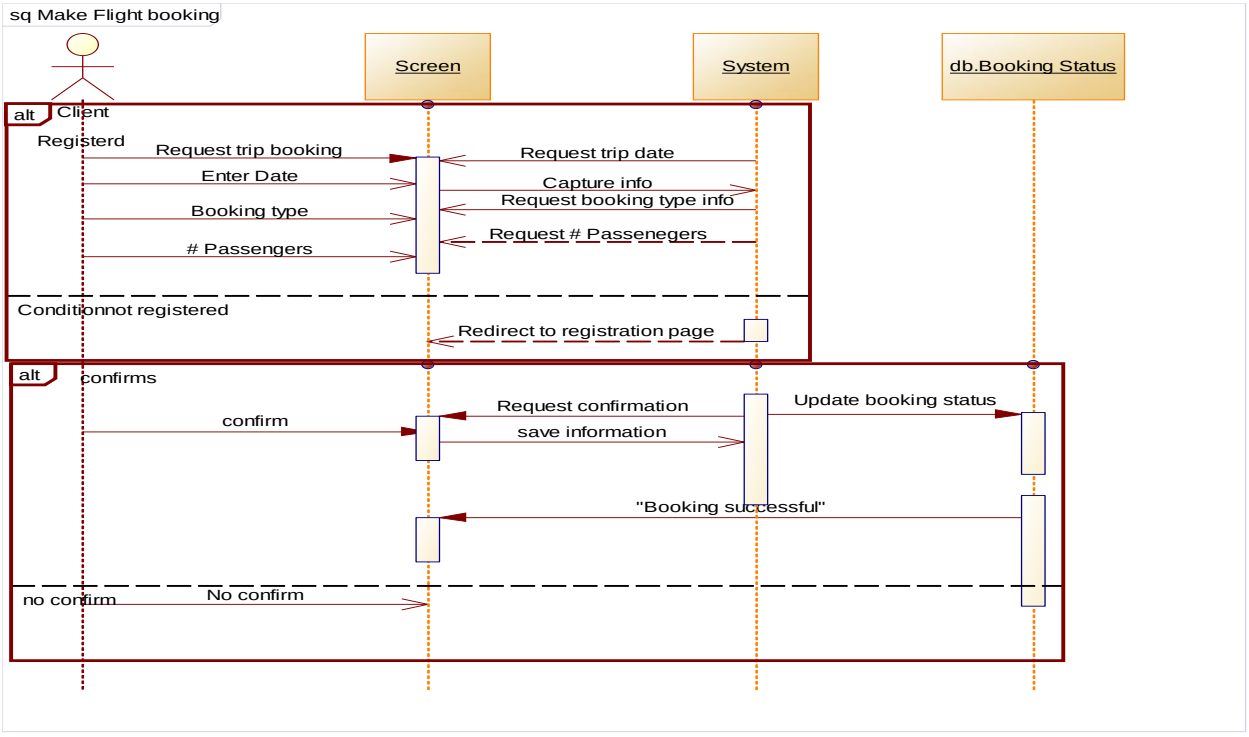
Sequence Diagram 1.1 Register Client



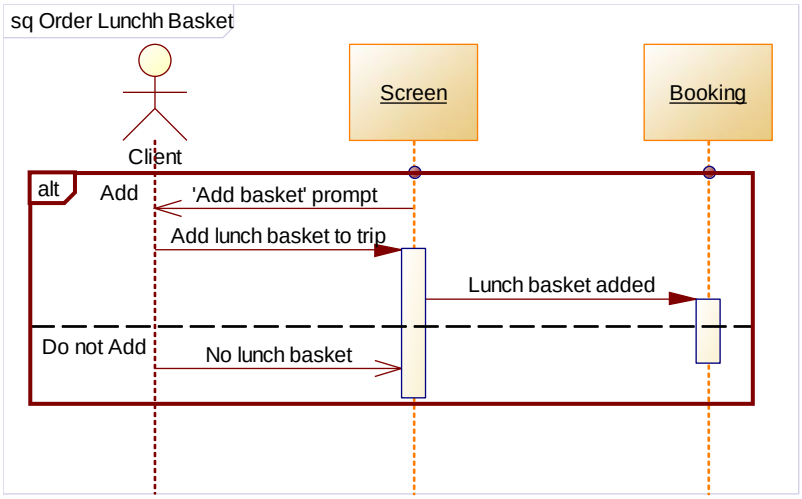
Sequence Diagram 1.2 View Available Flights



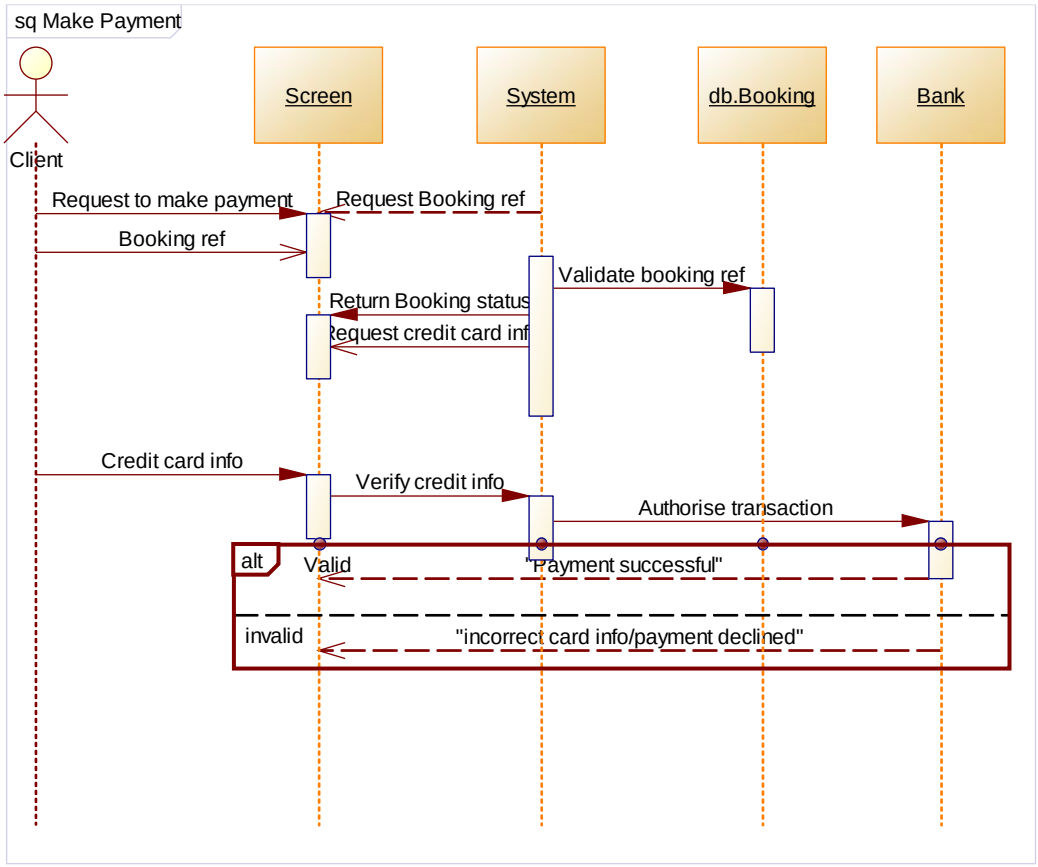
Sequence Diagram 1.3 Make Flight Booking



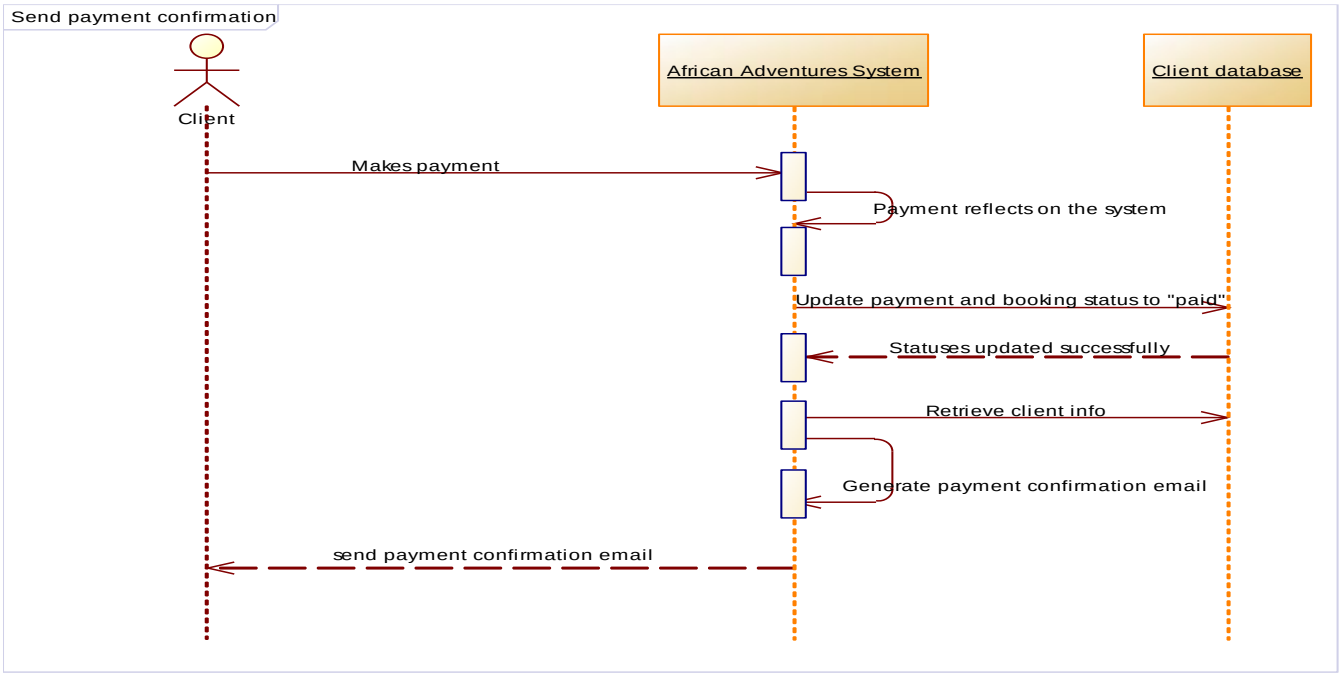
Sequence Diagram 1.4 Order Lunch Basket



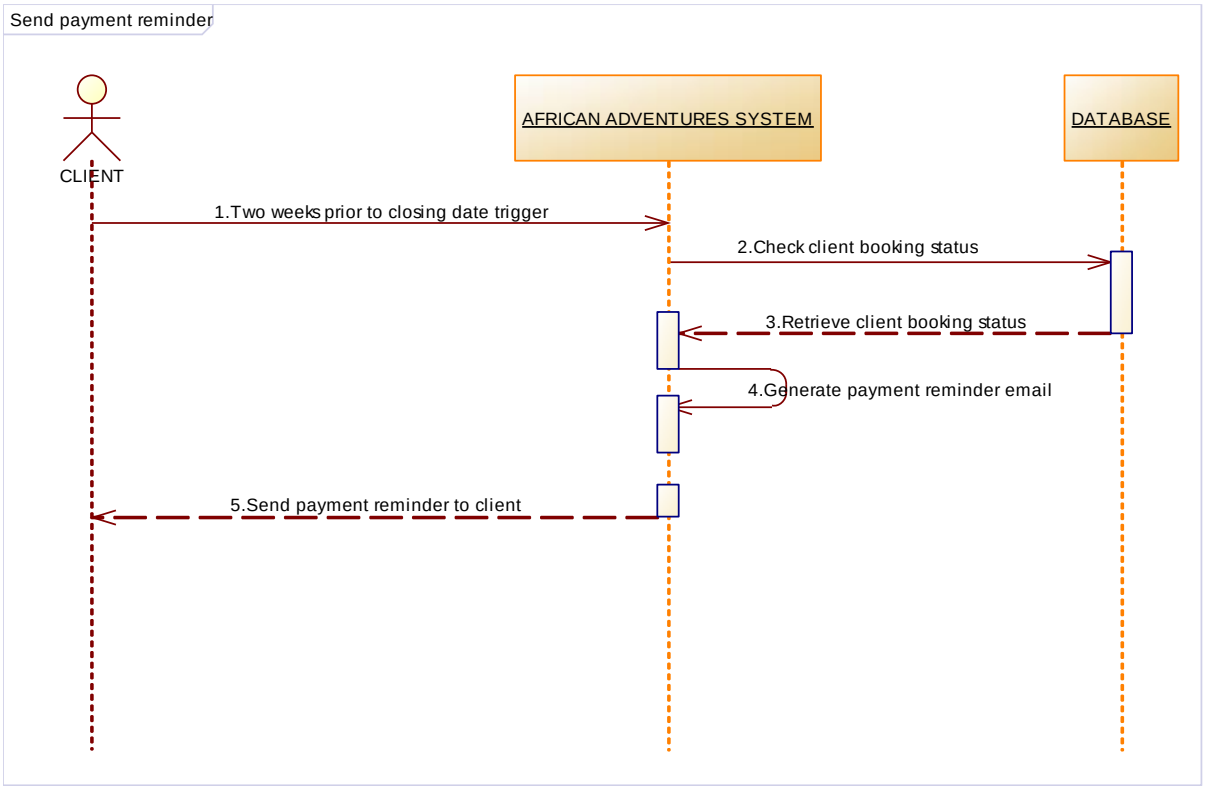
Sequence Diagram 1.5 Make Payment



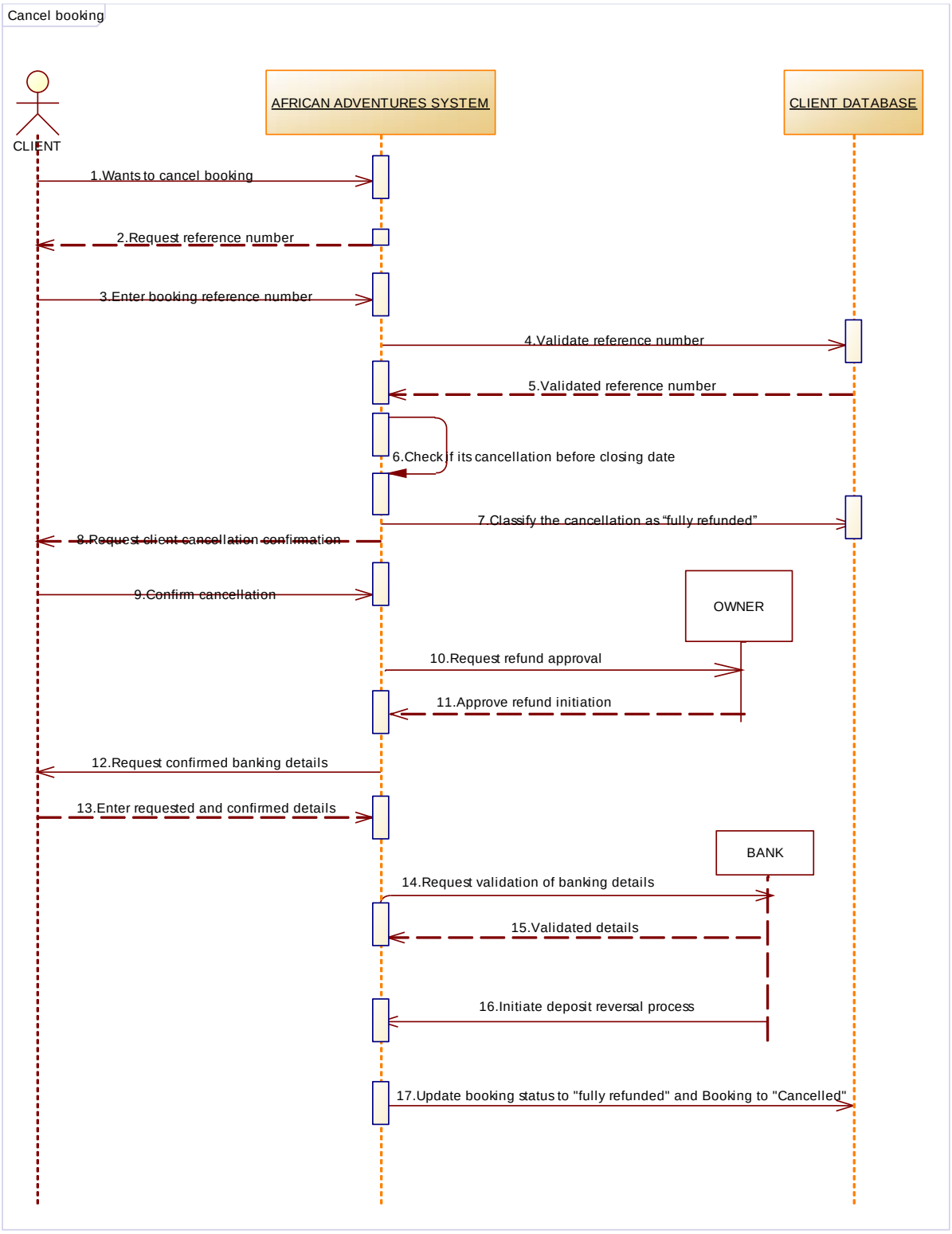
Sequence Diagram 1.6 Send Payment Confirmation



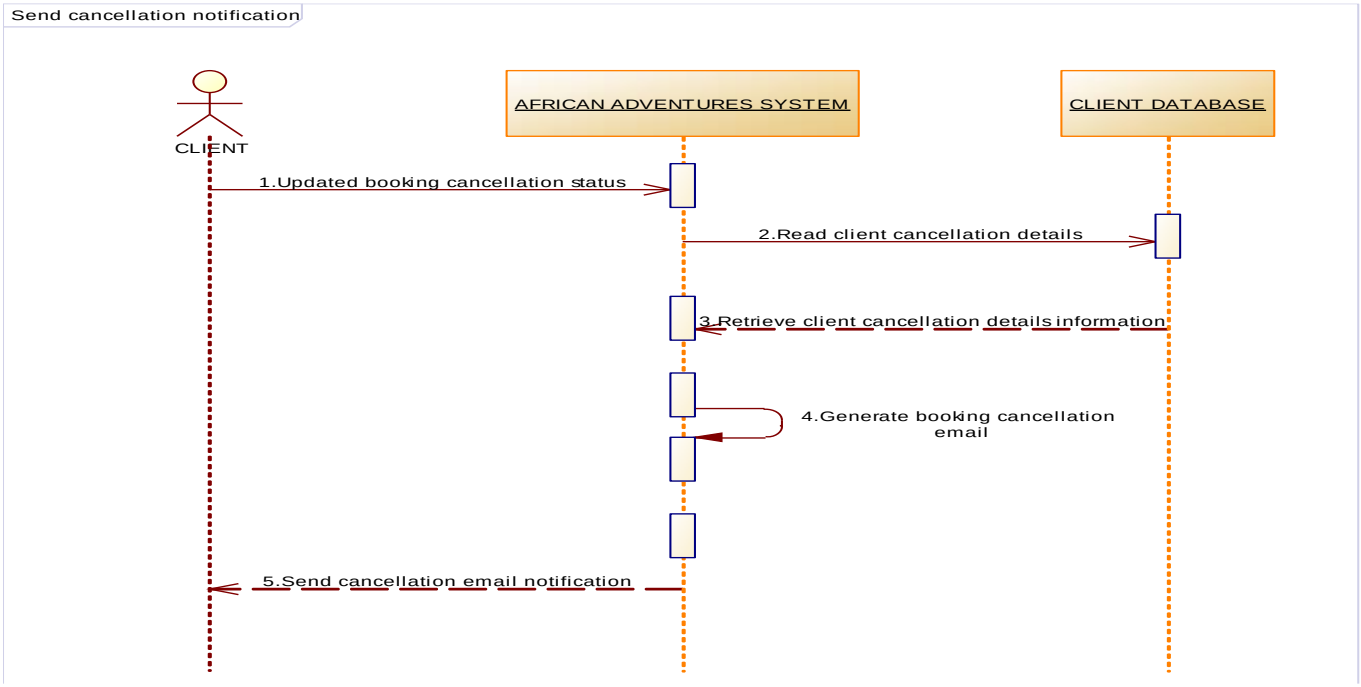
Sequence Diagram 1.7 Send Payment Reminder



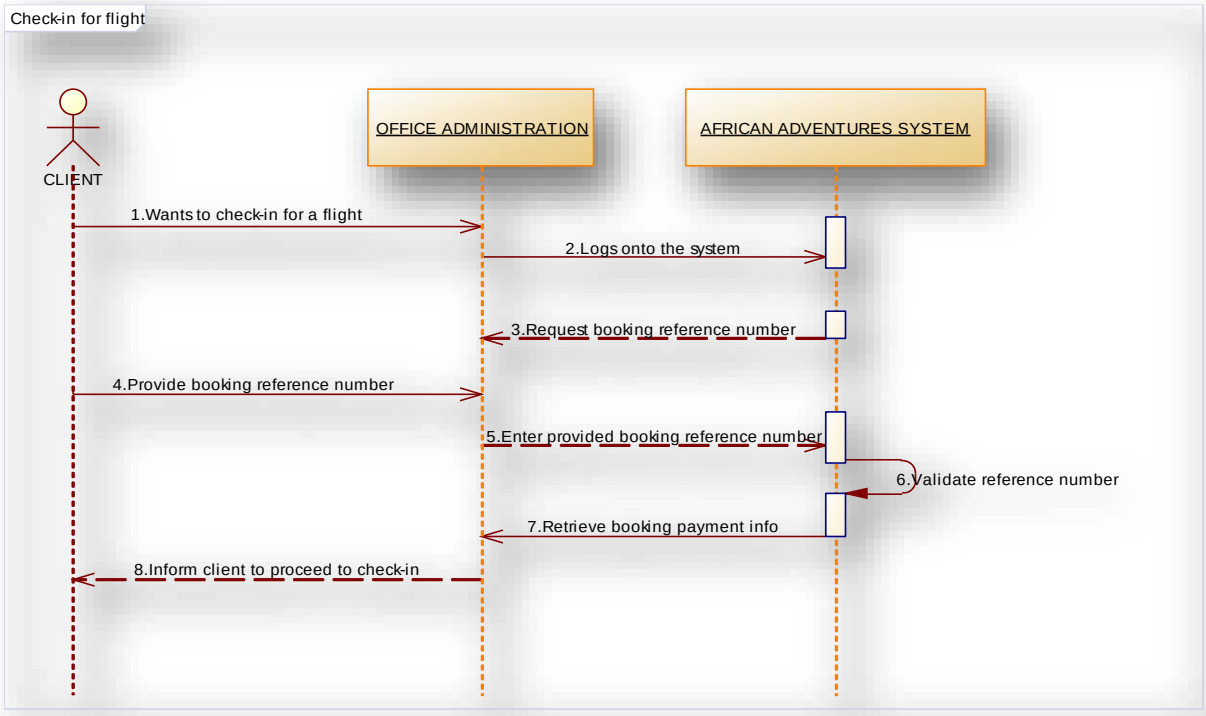
Sequence Diagram 1.8 Cancel Booking



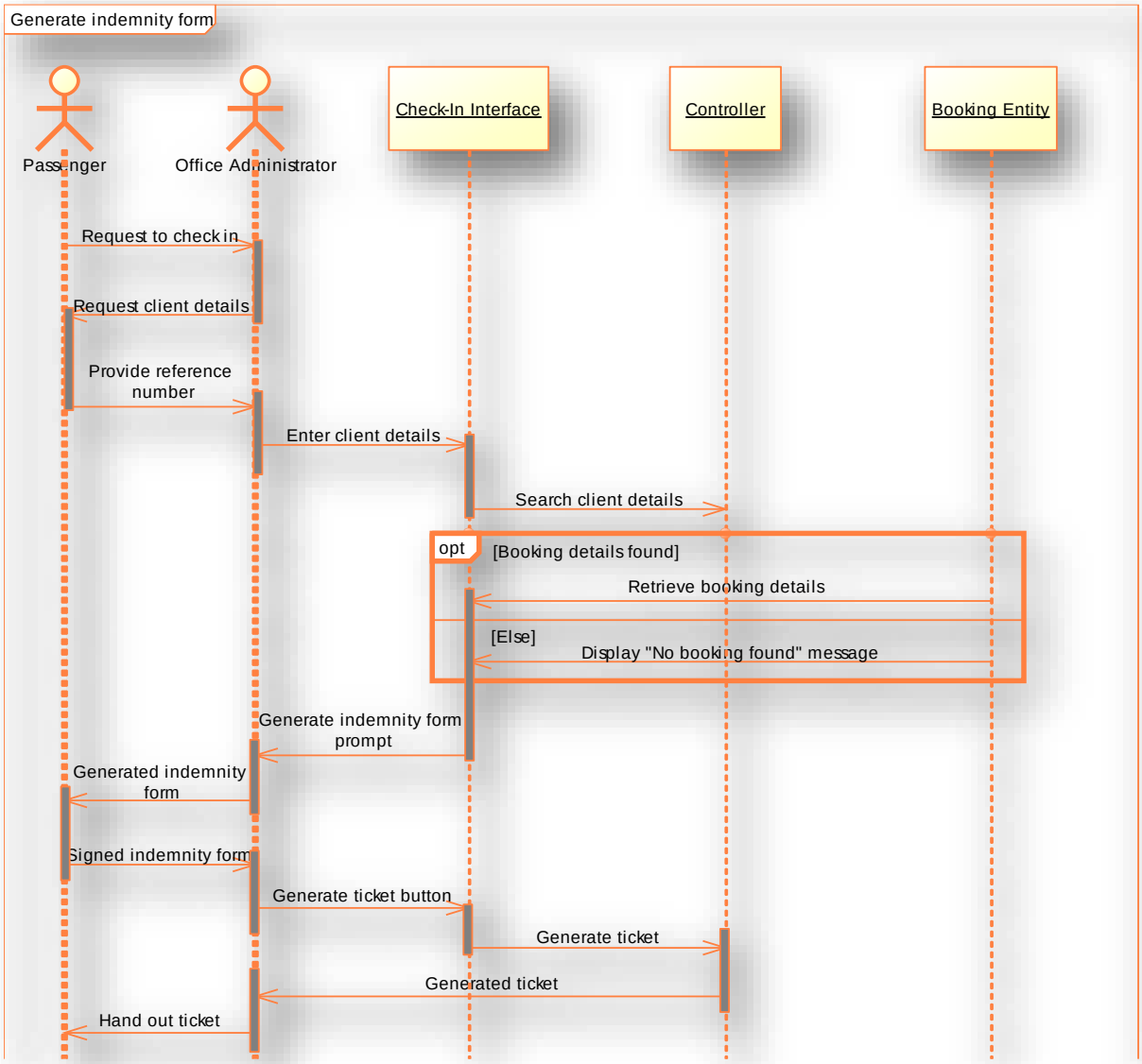
Sequence Diagram 1.9Send Cancellation notification



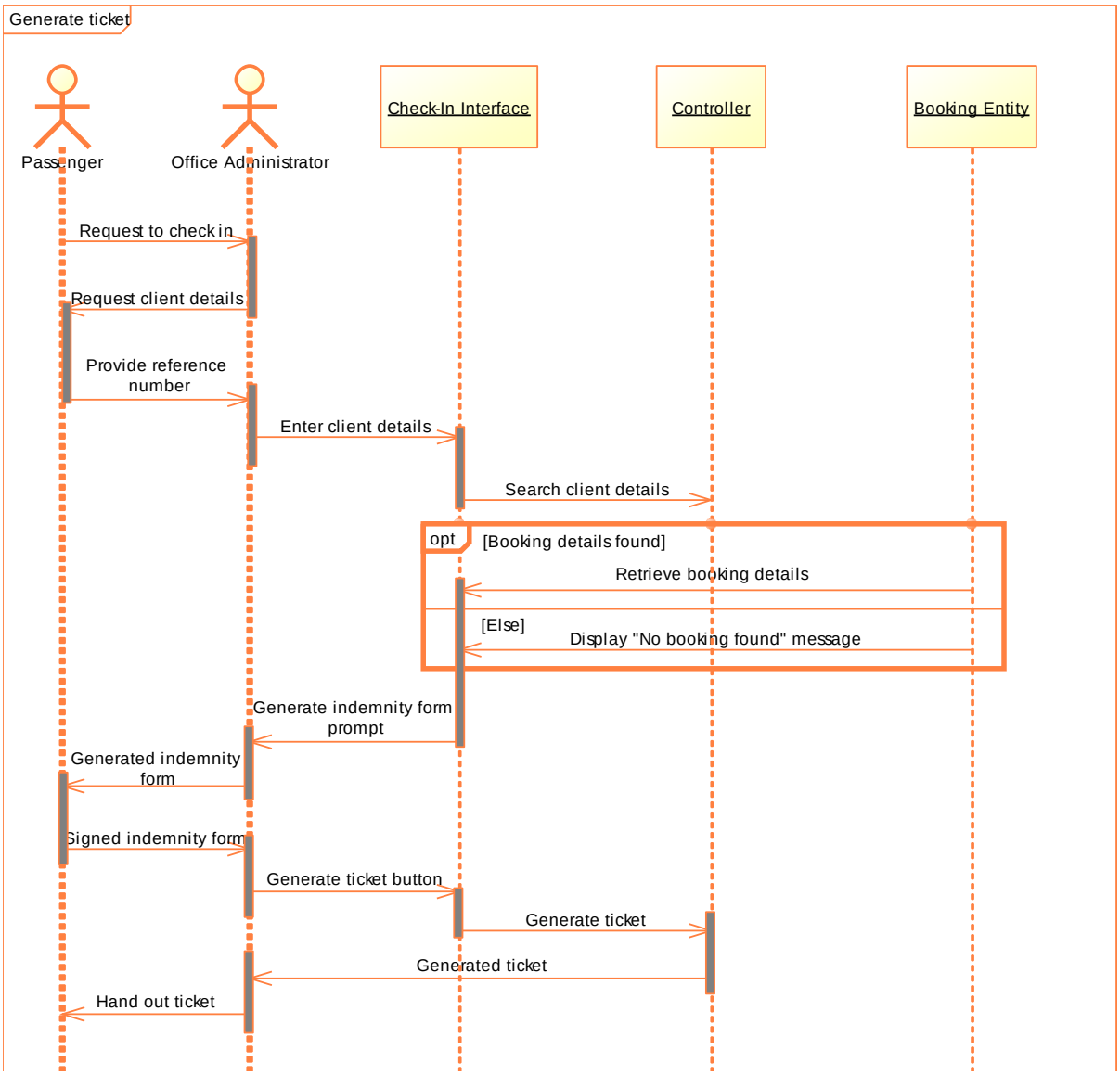
Sequence Diagram 2.1 Check-in for Flight



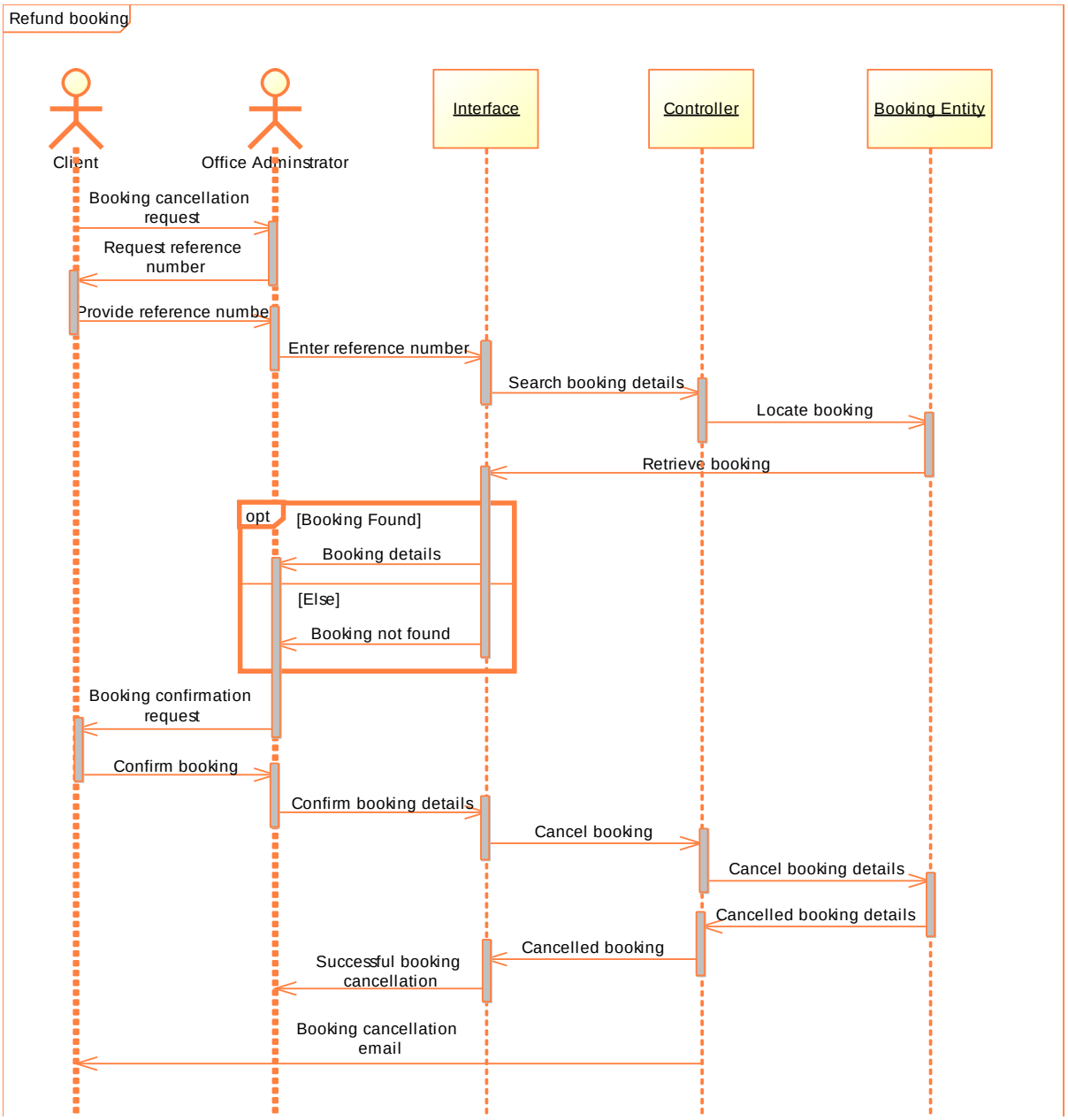
Sequence Diagram 2.2 Check-in for Flight

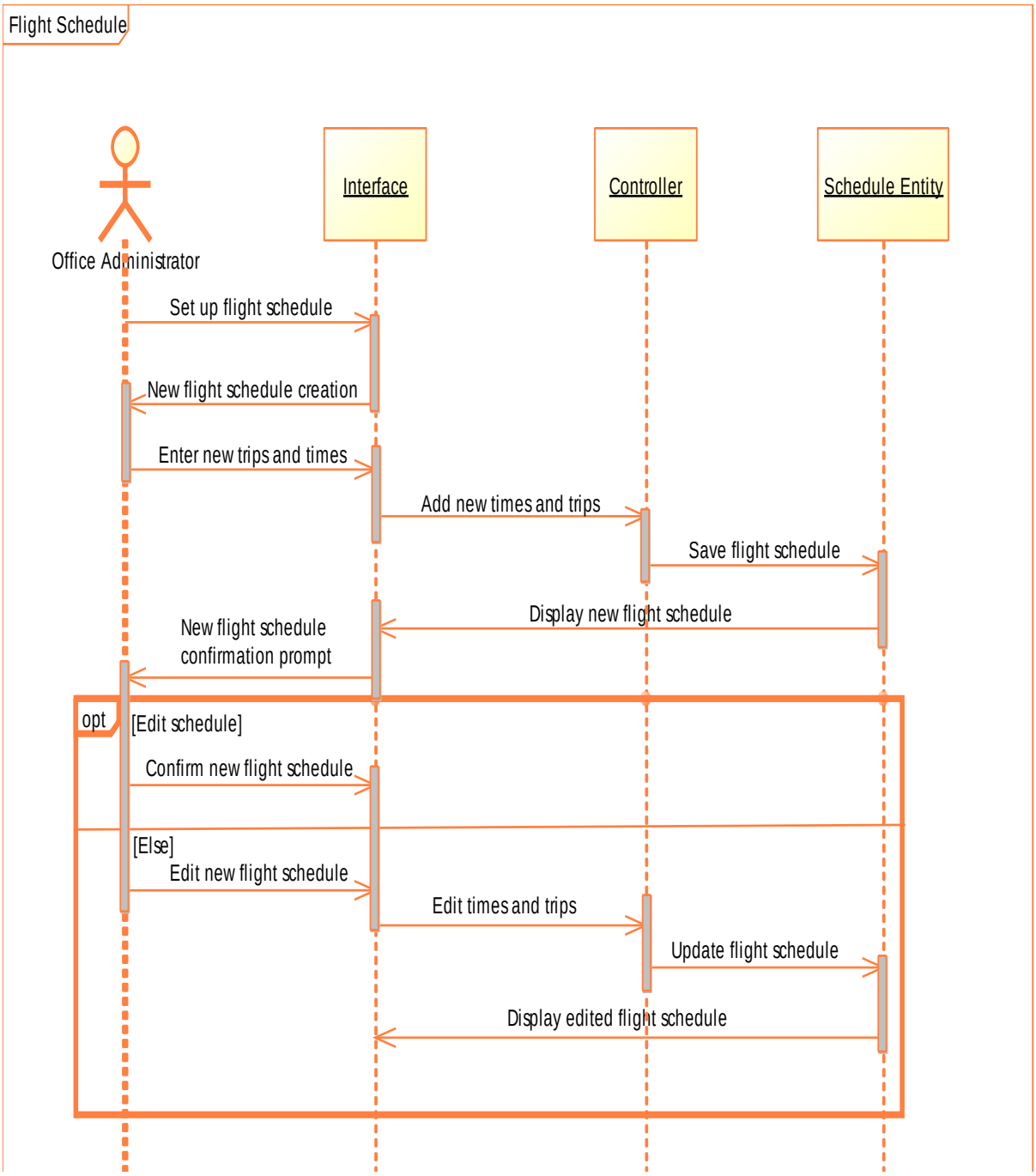


Sequence Diagram 2.3 Generate indemnity form

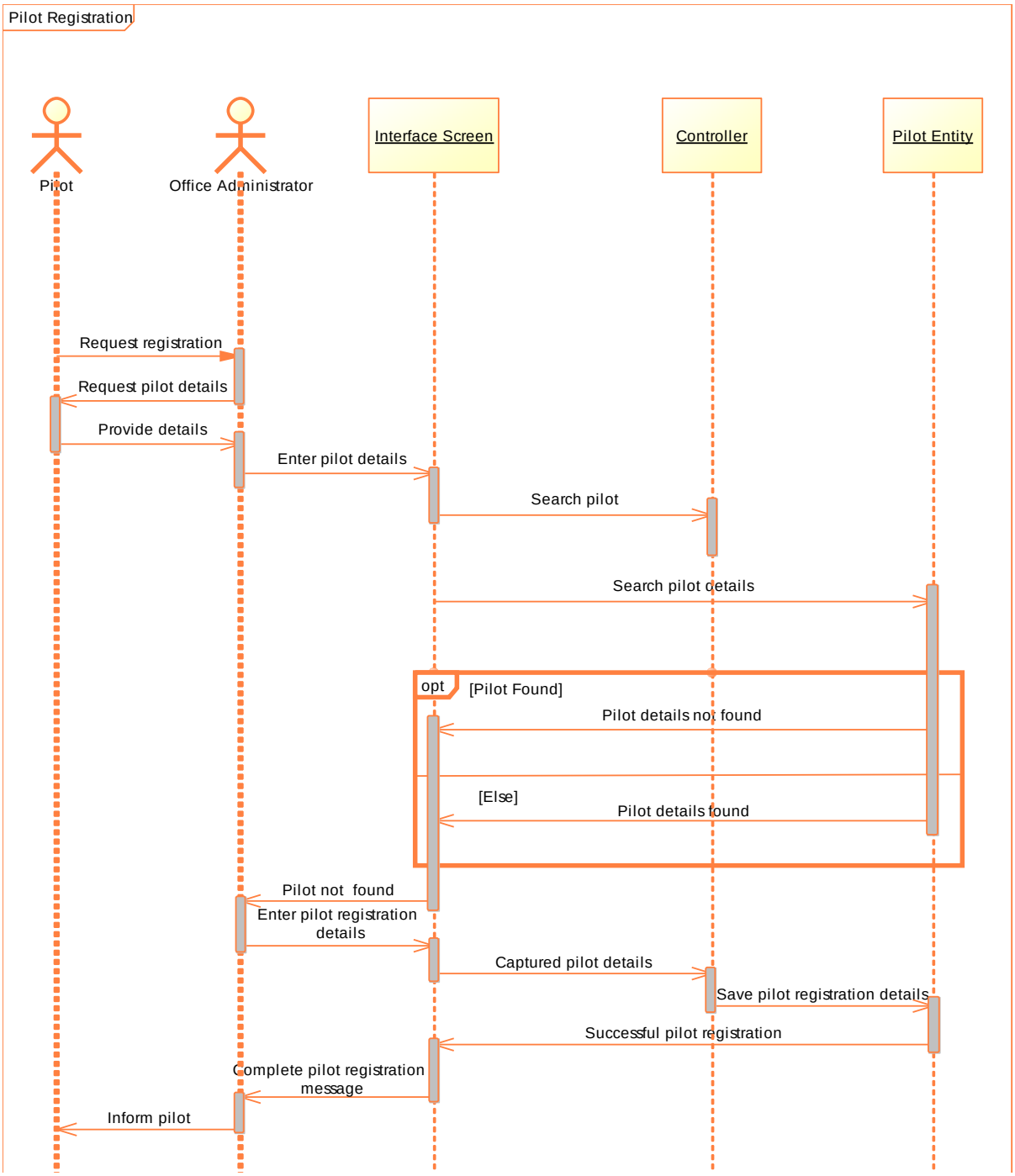


Sequence Diagram 2.4 Generate ticket


Refund booking
Client
Office Adminstrator
Interface
Controller
Booking Entity
Booking cancellation request
Request reference number
Provide reference numbe
Enter reference number
Search booking details
Locate booking
Retrieve booking
opt
[Booking Found]
Booking details
[Else]
Booking not found
Booking confirmation request
Confirm booking
Confirm booking details
Cancel booking
Cancel booking details
Cancelled booking details
Cancelled booking
Successful booking cancellation
Booking cancellation email

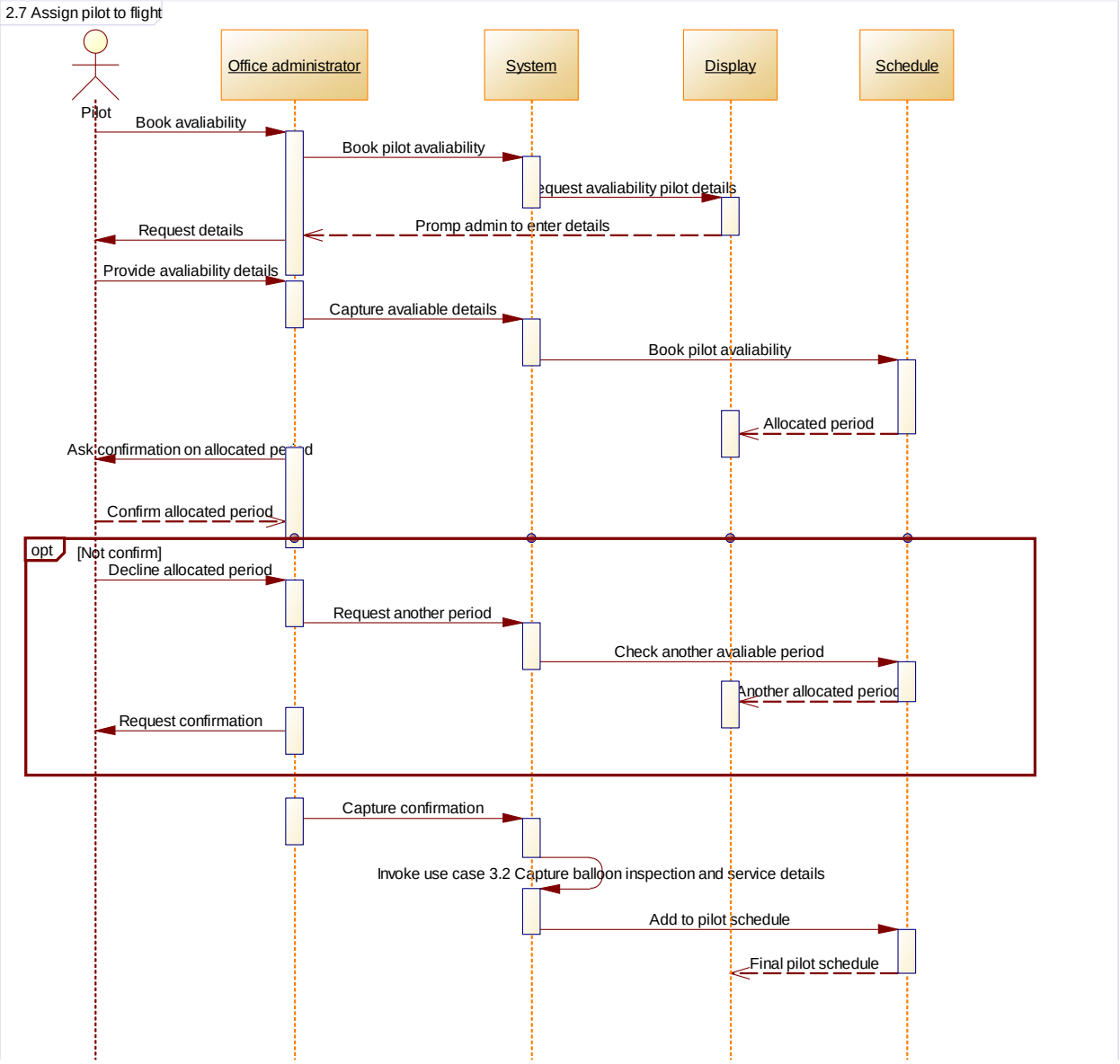


Sequence Diagram 2.5 Flight Schedule

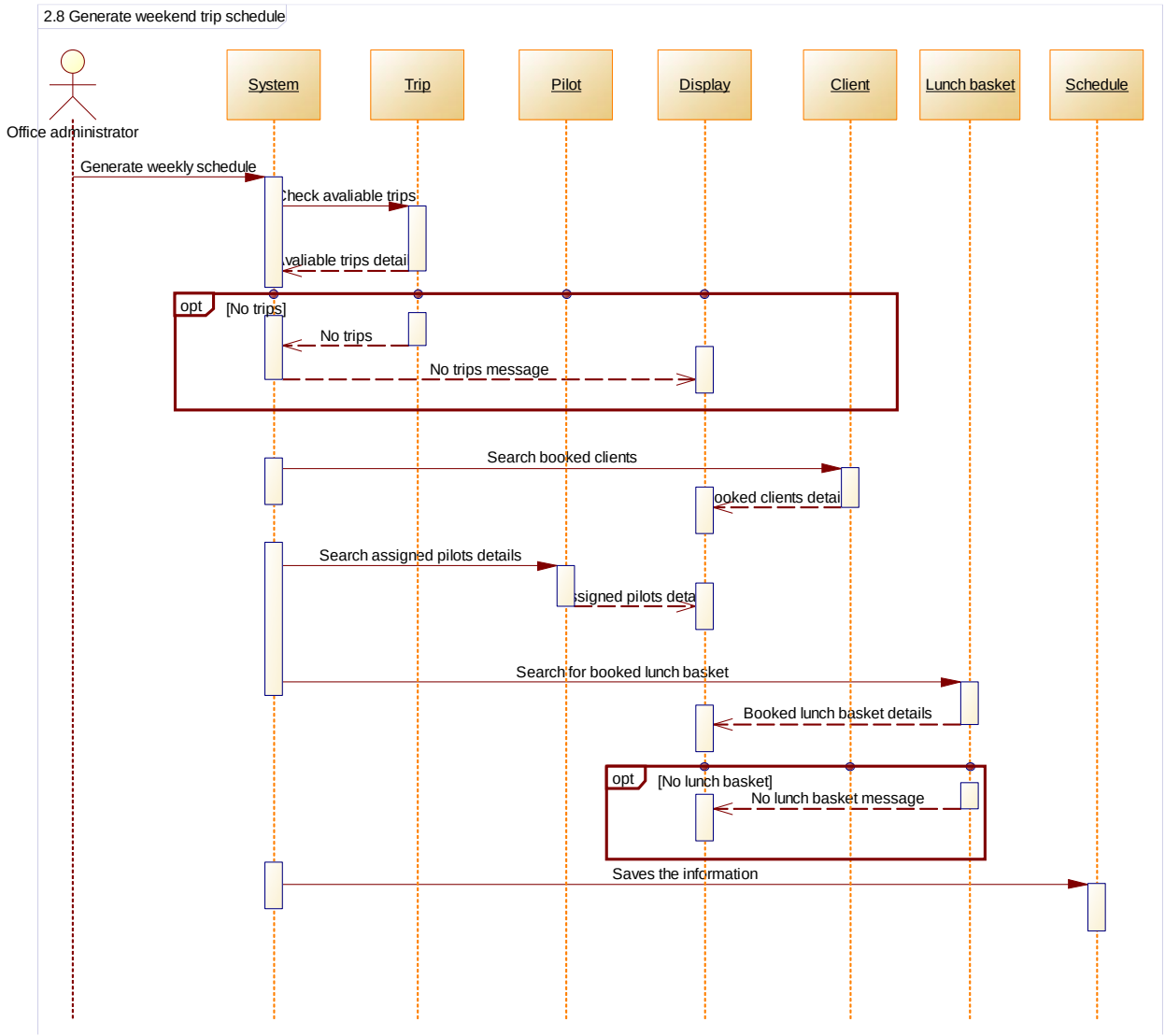


Sequence Diagram 2.6 Pilot Registration

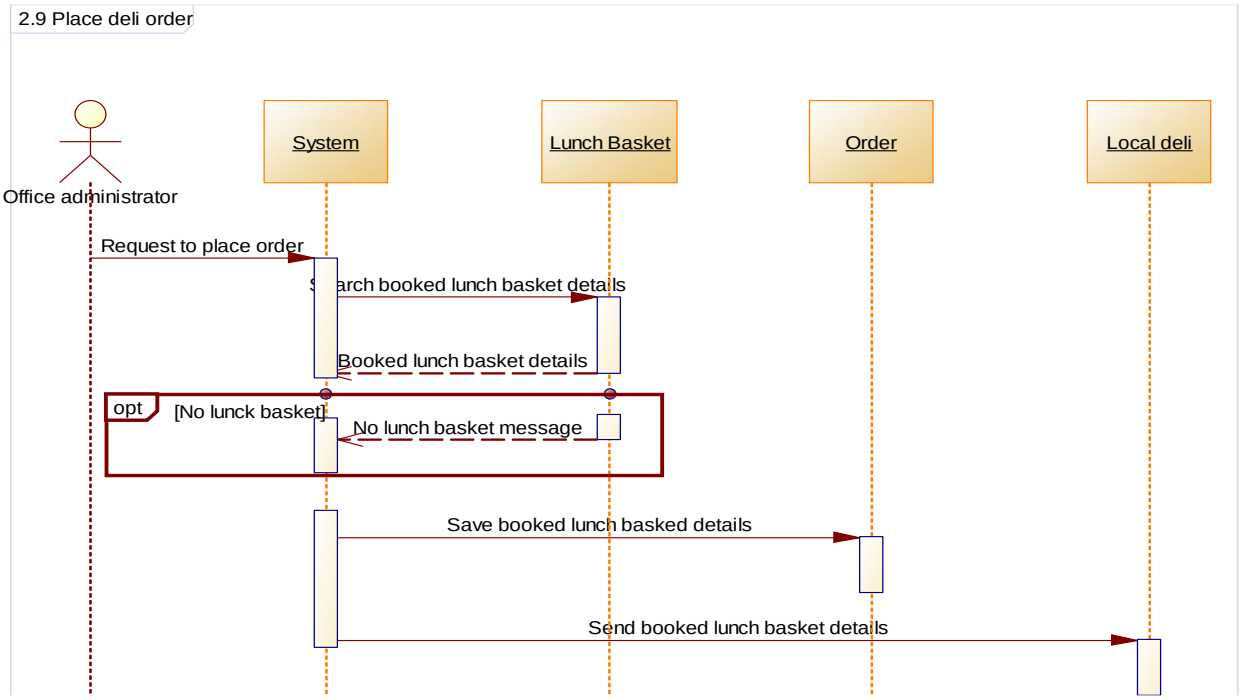
Sequence Diagram 2.7 Assign Pilot to Flight



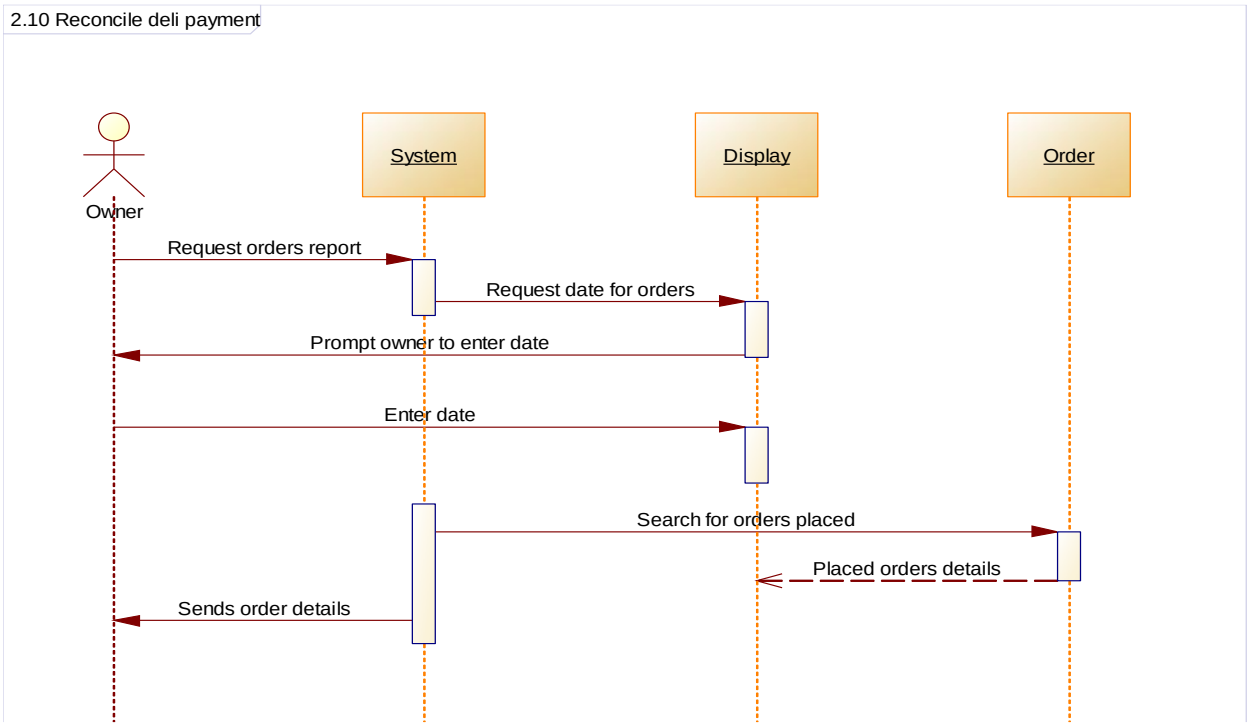
Sequence Diagram 2.8Generate weekend trip schedule



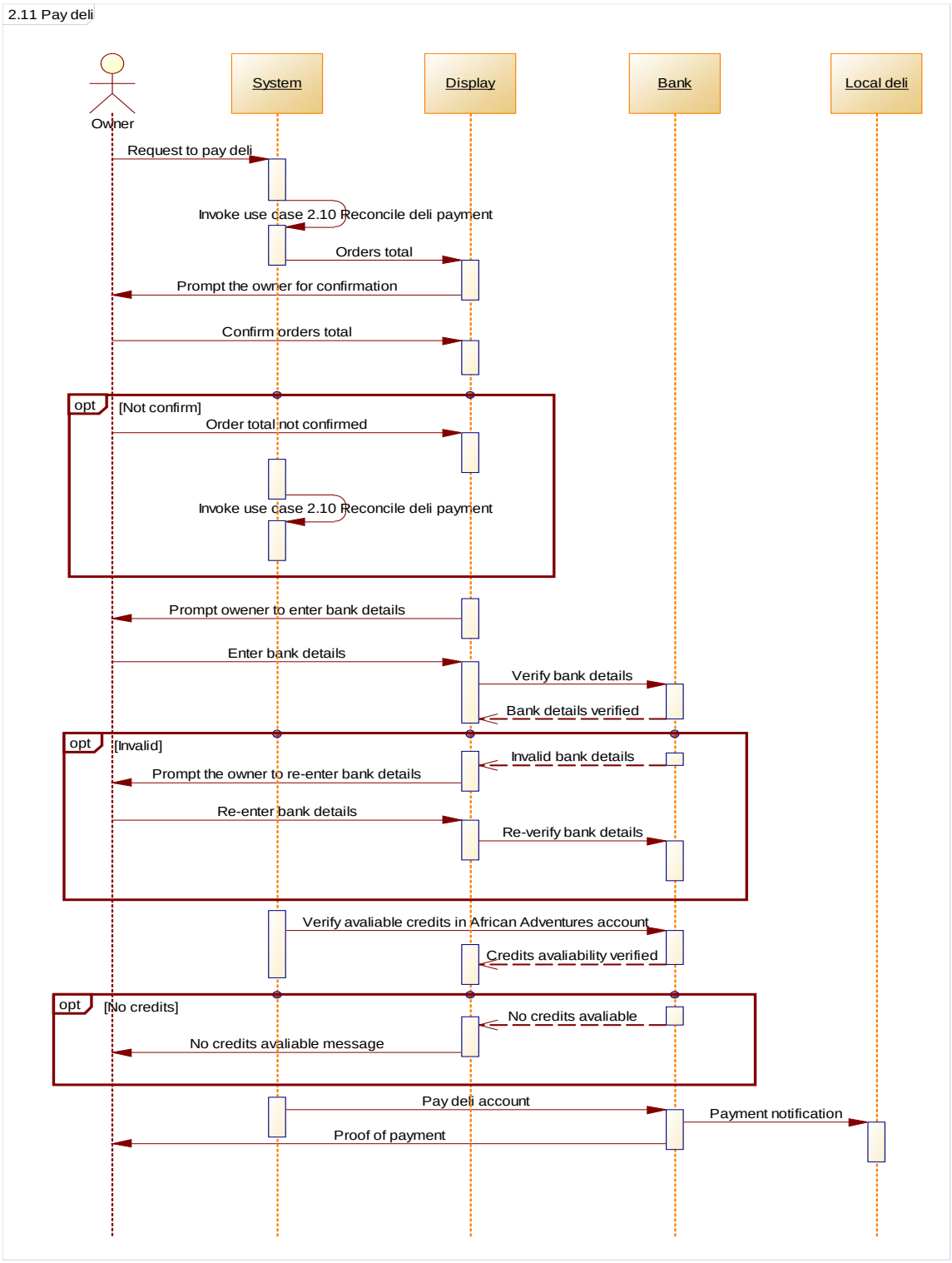
Sequence Diagram 2.9 Place Deli Order



Sequence Diagram 2.10 Reconcile deli Payment.



Sequence Diagram 2.11 Pay Deli



2.3A complete physical description of each method (pseudo code / structured English)

This section outlines the detailed physical description of each method from the user requirements it includes Methods names, input, processing of the input and the generated output.

1.ONLINE BOOKING FUNCTION

1.1Register Client

Method: RegisterClient()
Input: Client Details
Output: Client Registration Status

- Validate Client Details (full name, ID no, Physical Address, contact information).
- Create a new client record with the provided details.
- Store the client record in the database.
- Return Client Registration Status as the output.

1.2View Available Flights

Method: ViewAvailableFlights()
Input: Travel Date
Output: Available Flights List

- Retrieve all available flight details for the specified Travel Date.
- Format the flight details into Available Flights List.
- Return Available Flights List as the output.

1.3Make Flight Booking

Method: MakeFlightBooking()
Input: Client Details, Flight Details
Output: Booking Confirmation

- Validate Client Details (name, contact information).
- Validate Flight Details (travel date, flight type, number of passengers).
- Check flight availability.
- Calculate the booking cost.
- Create a new booking record with client and flight details.
- Store the booking record in the database.
- Generate a Booking Confirmation with booking details.
- Return Booking Confirmation as the output.

1.4Order Lunch Basket

Method: OrderLunchBasket()
Input: Booking Reference, Lunch Details
Output: Order Confirmation

- Retrieve the booking details using Booking Reference.
- Validate Lunch Details (type, quantity).
- Create an Order Confirmation with the lunch details.
- Store the order in the database.
- Return Order Confirmation as the output.

1.5Make Payment

Method: MakePayment()

Input: Booking Details, Payment Info

Output: Payment Status

- Validate Booking Details (i.e., booking reference, total amount).
- Validate Payment Info (i.e., credit/debit card details).
- Check if the booking is still open (before the deadline).
- Calculate the total payment amount.
- Attempt to process the payment using Payment Info.
- If the payment is successful:
    - ✓ Update the booking status to "Paid."
    - ✓ Return Payment Status as "Successful."
- If the payment fails:
    - ✓ Return Payment Status as "Failed."

1.6Send Payment Confirmation

- Method: SendPaymentConfirmation()
- Input: Booking Reference, Payment Status
- Output: Confirmation Email

- Retrieve the booking details using Booking Reference.
- Verify Payment Status.
- Generate a Confirmation Email with payment confirmation.
- Send the email to the client.
- Return Confirmation Email as the output.

1.7Send Payment Reminder

Method: SendPaymentReminder()

Input: Booking Reference, Payment Due Date

Output: Reminder Email

- Retrieve the booking details using Booking Reference.
- Calculate the Payment Due Date.
- Generate a Reminder Email with payment reminder and due date.
- Send the email to the client.
- Return Reminder Email as the output.

1.8Cancel Booking

Method: CancelBooking()

Input: Booking Reference

Output: Cancellation Confirmation

- Retrieve the booking details using Booking Reference.
- Verify the cancellation conditions if client cancel two weeks before closing date .
- If conditions met, mark the booking as cancelled.
- Generate a Cancellation Confirmation.
- Send the confirmation to the client.
- Return Cancellation Confirmation as the output.

1.9Send Cancellation notification

Method: SendCancellationNotification()

Input: Booking Reference

Output: Cancellation Notification Email

- Retrieve the cancelled booking details using Booking Reference.

- Generate a Cancellation Notification Email with cancellation details.
- Send the email to the client.
- Return Cancellation Notification Email as the output.

2.ADMIN FUNCTION

2.1Check-in for flight

Method: CheckInForFlight()
Input: Booking Reference
Output: Check in Confirmation

- Retrieve the booking details using Booking Reference.
- Verify the booking status and payment.
- Mark the passengers as checked in.
- Generate a Check-in Confirmation.
- Return Check-in Confirmation as the output.

2.2Generate Passenger Indemnity Form

Method: GeneratePassengerIndemnityForm()
Input: Passenger Details, Travel Date
Output: Indemnity Form

- Validate Passenger Details (name, ID number, physical address).
- Generate an Indemnity Form with passenger details and travel date.
- Return Indemnity Form as the output.

2.3Generate Ticket

Method: GenerateTicket()

Input: Booking Details

Output: Boarding Ticket

- Retrieve the booking details using booking reference number.
- Generate a Boarding Ticket with flight information and passenger names.
- Return Boarding Ticket as the output.

2.4Refund Booking

Method: RefundBooking()

Input: Booking Reference

Output: Refund Status

- Retrieve the booking details using Booking Reference.
- Check the booking cancellation conditions.
- Process refunds if applicable.
- Update the booking status.
- Return Refund Status as the output.

2.5Setup Flight Schedule

Method: SetupFlightSchedule()

Input: Pilot Availability, Trip Details

Output: Schedule Confirmation

- Retrieve pilot availability for the specified period.
- Create a new trip schedule with available pilots and trip details.
- Store the schedule in the system.
- Generate a Schedule Confirmation.
- Return Schedule Confirmation as the output.

2.6Register Pilot

Method: RegisterPilot()
Input: PilotDetails
Output: Pilot Registration Status

- Validate PilotDetails (name, contact information).
- Create a new pilot record with the provided details.
- Store the pilot record in the system.
- Return Pilot Registration Status as the output.

2.7Assign Pilot to Flight

Method: AssignPilotToFlight()
Input: PilotDetails, Flight Details
Output: Assignment Status

- Retrieve the available pilots and flight details.
- Assign a pilot to the specified flight.
- Update the flight record with pilot information.
- Return Assignment Status as the output.

2.8Generate Weekend Trip Schedule

Method: GenerateWeekendTripSchedule()
Input: Trip Details
Output: Weekend Schedule

- Retrieve trip details for the upcoming weekend.

- Create a schedule with available pilots and trips.
- Store the schedule in the system.
- Return Weekend Schedule as the output.

2.9Place Deli Order

Method: PlaceDeliOrder()
Input: Deli Order Details
Output: Order Confirmation

- Retrieve Deli Order Details (order items, quantities).
- Place an order with the local deli based on the details.
- Receive an order confirmation from the deli.
- Return Order Confirmation as the output.

2.10Reconcile Deli Payment

Method: ReconcileDeliPayment()
Input: Deli Invoice, Order Details
Output: Reconciliation Status

- Retrieve the deli's monthly invoice and order details.
- Compare the invoice with the orders placed by the company.
- Calculate the total amount due.
- Update the system with the payment status.
- Return Reconciliation Status as the output.

2.11Pay Deli

Method: PayDeli()
Input: Invoice Amount
Output: Payment Status

- Retrieve the amount due from the invoice.
- Initiate an EFT payment to the deli.
- Record the payment transaction in the system.
- Return Payment Status as the output.

2.12Send Client Newsletter

Method: SendClientNewsletter()
Input: Newsletter Content
Output: Delivery Status

- Prepare a newsletter with Newsletter Content.
- Retrieve a list of client email addresses.
- Send the newsletter to all clients on the list.
- Record the delivery status for each email.
- Return Delivery Status as the output.

3.BALLOON SERVICE FUNCTION

3.1Add Balloon

Method: AddBalloon()
Input: Balloon Details
Output: Balloon Add Status

- Validate Balloon Details (balloon type, registration).
- Create a new balloon record with provided details.
- Store the balloon record in the system.
- Return Balloon Add Status as the output.

3.2Capture Balloon Inspection and Service Details

Method: CaptureBalloonInspection()
Input: Balloon Registration, Inspection Details
Output: Inspection and Service Status

- Retrieve the balloon using Balloon Registration.
- Capture inspection details (e.g., defects, service requirements).
- Update the balloon record with inspection data.
- Return Inspection and Service Status as the output.

3.3 Track Balloon Flight Hours

Method: TrackBalloonFlightHours()
Input: Balloon Registration, Flight Hours
Output: Flight Hour Status

- Retrieve the balloon using Balloon Registration.
- Update the total flight hours with the provided Flight Hours.
- Check if the balloon requires a full service.
- Return Flight Hour Status as the output.

3.4Generate Full Inspection Report

Method: GenerateFullInspectionReport ()

Input: Inspection Data

Output: Inspection Report

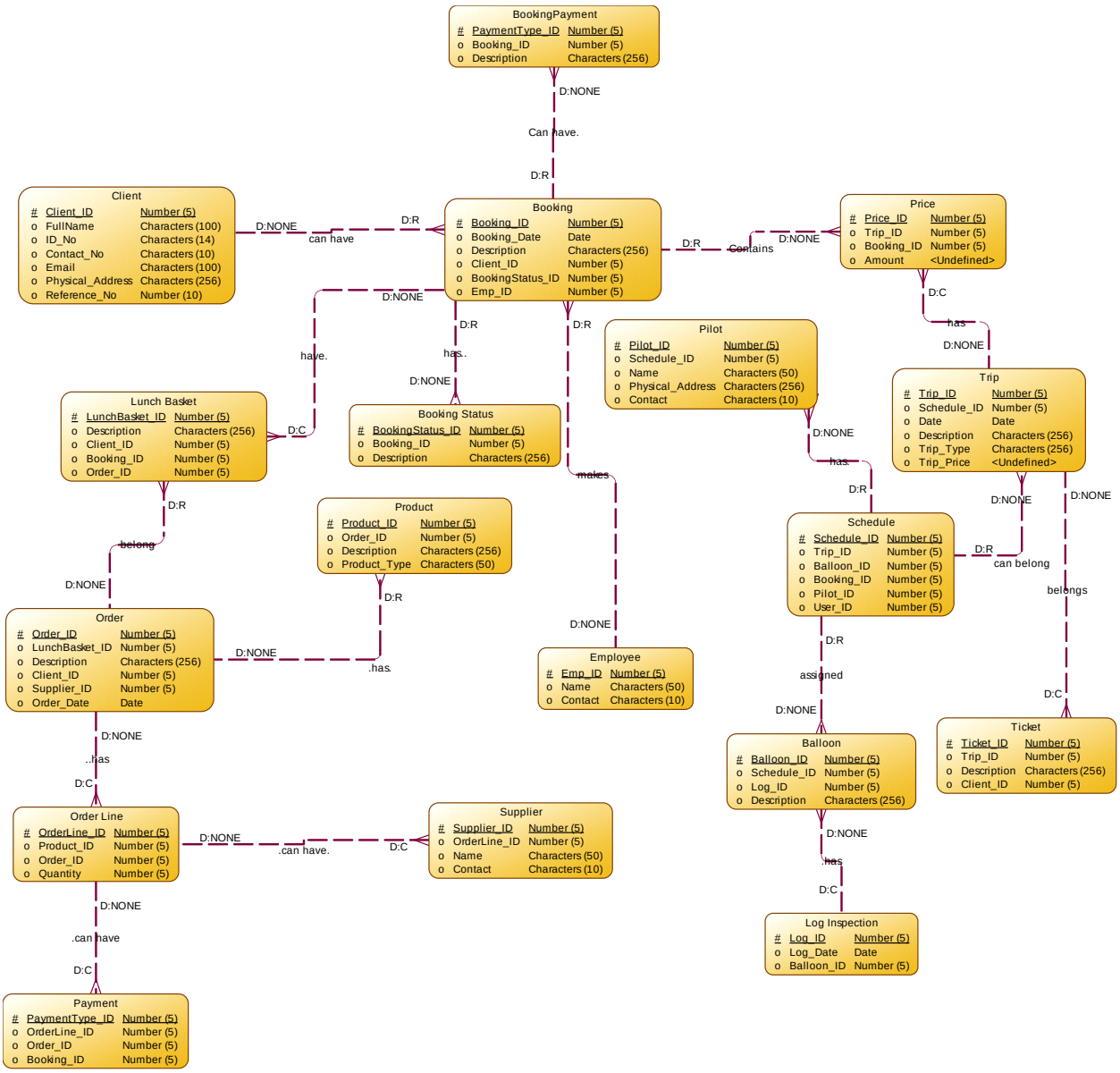
- Retrieve inspection data for all balloons.
- Generate a comprehensive inspection report.
- Provide details on balloon conditions and service requirements.
- Return Inspection Report as the output.

Conclusion

In conclusion, the provided pseudocodes for the methods in the system help facilitate various functionalities, such as booking, check-in, and administrative tasks, ensuring efficient operations for African Adventures.

3.Enitity Relationship Model

A technical ERD provides African Adventures with a visual represantation of their database structure, the relationships between the entities ensures data integrity and it offeres more information on how the database should be created.



These offers African Adventures a higher chance of successfully implemeting, creating, maintaining, and optimising their database system.

4.Test Specifications

4.1Test Scenarios and procedures

A test scenario describes an instance in which a system user may want to achieve a certain objective from the system such a “Make Booking”. With the sole purpose of validating our software’s execution of each use case. The following test scenarios are designed to assess the efficiency of the 3 core use cases of the African Adventures Information System

Test Case :1.1 Register Client	
Use Case	1.1 Register Client
System	African Adventures
Sub-system	Online Booking
Short Description	A Client wishes to register on the African Adventures Website
Preconditions	-User Must be logged into the system -System has loaded home page
Expected Functionality	This use case begins when a client wants to register on the company’s website. They will accomplish this by clicking on Online Booking Tab.Thereafter selecting the ‘register as client ‘radio button. After which they will have to click on the select button. This will be followed by the Register Client screen loading. Here the customer will follow the sequential prompts from the website which require Id number, email address, phone numbers, name, and surname of the client, once the customer has confirmed this detail, they will see a ‘customer successfully registered message box”

Test Case :1.3 Make flight booking	
Use Case	1.3 Make flight booking
System	African Adventures
Sub-system	Online Booking
Short Description	Client wishes to book a flight.
Preconditions	-Logged into system, -User must be a registered Client -System must show the Online booking tab
Expected Functionality	This use case begins when a client wants to make an online booking for a hot air balloon trip. They will accomplish this by clicking on Online Booking Tab.Thereafter selecting the 'Book a Trip 'radio button. After which they will have to click on the select button. This will be followed by the Booking Screen loading. Here the customer will follow the sequential prompts from the website which require trip date selection, trip type dropdown option, number of passengers attending. The client will then click on the submit button this will be with a 'confirm booking notification 'to which the client clicks on the confirm button. This will the direct the Client to the Make payment page before a "Booking confirmed message box appears

Test Case :2.6 Register Pilot	
Use Case	2.6 Register Pilot
System	African Adventures
Sub-system	Admin
Short Description	Pilot wants to register on System.
Preconditions	- Administrator logged in to system -System should load Admin tab
Expected Functionality	The use case begins with the Office Administrator clicking on the admin tab. It is then followed by the Administrator selecting the register pilot dropdown option. This results in the loading of the register pilot screen. Once loaded the Administrator will enter the pilots provided id number to search if the pilot exists on the system. The system will return "user not found "notification which will prompt the administrator to enter the pilots' full names, phone number, email address and id number. The administrator will then click on the save pilot button, after which a notification of successful registration will be generated

4.2Test Data

Test Scenario	Test Data		Expected Results		Actual Results		Pass/Fail	
Use case 1.1 Register Client	Correct	Incorrect	Correct test data	Incorrect TD	Correct TD	IncorrectTD	Pass	Fail
1.Register Screen Loads			All Controls load. With enabled Next button. Text boxes load empty by default					
2.User will type them Name and surname in Respective textboxes	Enter Valid Name and Surname		Tab moves to ID textbox					
3.User Enter Id, phone number email and address in respective textboxes	Enter 13-digit ID. With address Valid email		All controls filled with Client detail					
4.User clicks on the Next Button	-		Message box instructing Confirmation of entered details appear					
5.User Clicks on the okay button	-	Click Cancel button	Client Registered Message box	Registration error notification				

6.User Clicks the close button	-							

Test Scenario	Test Data		Expected Results		Actual Results		Pass/Fail	
Use case 1.3. Make Flight Booking	Correct	Incorrect	Correct test data	Incorrect TD	Correct TD	IncorrectTD	Pass	Fail
1.Booking Screen Loads			All Controls load. With enabled View dates and submit button. Text boxes load empty by default					
2.Client will select Booking type from dropdown	Booking type dropdown: 'Full day'		Dropdown value set to 'Full day'					
3.Client will press on the View dates button	Available flights page appears		Calendar with available slots appears					
4.Click back button	-		Redirect to booking					
5.Fill in number of passengers	No of passengers numeric up down Valid number		No passenger saved					
6.Click submit button	-		Confirm information notification with					
7.Click okay button		Click Cancel	Redirection to payment page					

Test Scenario	Test Data		Expected Results		Actual Results		Pass/Fail	
Use case 2.6. Make Flight Booking	Correct	Incorrect	Correct test data	Incorrect TD	Correct TD	IncorrectTD	Pass	Fail
1.Pilot list Screen Load			All Controls load. With enabled add, delete, edit, and cancel button. Text boxes load empty by default					
2.Administrator click on Add button	-		Search Pilot screen appears with search box					
3.Administrator searches pilot name on system	Enter relevant. Name:' John'		Message box 'Pilot not found appears'					
4.Adminstrator Clicks Ok button	-		Add pilot screen loads with relevant textboxes for name surname email, id, gender, experience, and marital status					
5.Administrator fills in fields with relevant data	Fill in textboxes. With pilot name ,surname ,ID,email and phone		Confirm details notification. With ok and cancel button					
6.Administrator Clicks ok button	-		Pilot successfully registered notification	Click cancel button				

Conclusion:

With the above test data and scenarios a potential user can succinctly test out ,our prototype for African Adventures as we have included detailed descriptions of the processes involved in completing 3 of a core fundamental Functions namely ,registering a client, pilot and making a booking for a hot air balloon trip.

5.Validation

8.VALIDATION

Introduction

This section is dedicated to delivering a comprehensive account of the processes taking place on the system response side, particularly focusing on primitive-level processes. It elucidates the intricacies of process execution and the diverse array of entities implicated. Within this context, it meticulously outlines each requirement and adeptly delineates which use cases, primitive-level processes, entities, methods, screens, and reports are in precise alignment with these requirements.

BOOKING SUBSYSTEM							
Requirement	Use Case	Process on data flow diagram	Invoke	Entities	Methods	Screens	Outputs AND/OR Reports
1.1 Register client	1.1Register client	1.1.1Process client details	None	None	Process client details()	Process client details screen	Processed client details
		1.1.2Confirm client details	None	None	Confirm client details()	Confirm client details screen	Confirmed client details
		1.1.3Validate and capture client details	None	Client	Validate and capture client details ()	Validate and capture client screen	Validated and capture client details
		1.1.4Finalise Registration	None	None	Register client ()	Finalise Registration screen	Finalised Registration details
1.2View available flights	1.2View available flights	1.2.1 Display available flights	None	flights	View available flights ()	View available flights screen	Available flights
1.3 Make flight booking	1. Make flight booking	1.3.1 Select trip date	1.1Register client	None	Select trip date()	Select trip date screen	Selected trip details
		1.3.2 Capture trip date	None	None	Capture trip date()	Capture trip date screen	Captured trip date details
		1.3.3 Booking type details	None	None	Booking type details()	Booking type details screen	Booking type details
		1.3.4 Capture entered information	None	None	Capture entered information()	Capture entered information screen	Captured entered information
		1.3.5 Request passenger info	None	None	None	None	None
		1.3.6 Capture passenger details	None	None	Capture passenger details ()	Capture passenger details screen	Captured passenger details
		1.3.7 Confirmation	None	None	Display confirmation	Confirmation screen	Confirmation

		1.3.8Update booking status	None	Booking status	Update booking ()	Update booking screen	Updated booking details
1.4Order lunch basket	1.4Order lunch basket	1.4.1Lunch basket selection	None	Lunch basket	Order lunch basket ()	Order lunch screen	Order lunch basket confirmation
1.5Make Payment	1.5Make payment	1.5.1 Retrieve booking details	None	Booking status	Search booking status()	Search booking status screen	Retrieve
		1.5.2 Credit card	None	None	None	Credit card screen	Credit card
		1.5.3Validate and authorize payment	None	None	Validate booking status()	Validate booking status screen	Validated booking status
		1.5.4 Finalise payment	None	None	Finalise payment()	Finalise payment screen	Invoice
1.6Send payment confirmation	1.6Send payment confirmation	1..6.1 Retrieve full payment	None	None	Search booking status()	Search booking status screen	Retrieved booking status
		1.6.2Update payment and booking status	1.1register client	Payment status	Update booking status()	Update booking screen	Updated booking details
		1.6.3 Generate client payment confirmation email	None	None	Generate client payment confirmation email()	Generate client payment confirmation email screen	Generated client payment confirmation email
		1.6.4 Send payment confirmation email	None	None	Send payment confirmation email()	Send payment confirmation email screen	Sent payment confirmation email message
1.7Send payment reminder	1.7Send payment reminder	1.7.1 Retrieve booking and payment status info	None	Payment status	Retrieve booking and payment status info ()	Retrieve booking and payment status info screen	Retrieved booking and payment status info
		7.1.2 Generate payment reminder email	None	None	Generate payment reminder email()	Generate payment reminder email screen	Generated payment reminder email
		1.7.3 Send payment reminder email to client	None	None	Send payment reminder email to client()	Send payment reminder email to client screen	Sent payment reminder email to client message
1.8Cancel booking	1.8Cancel booking	1.8.1Retrieve reference number	None	Booking	Search reference number()	Search reference number	Retrieved reference number
		1.8.2 Check refund type	None	None	Check refund type()	Check refund type	Checked refund type details

		1.8.3 Cancellation confirmation request	None	None	None	None	None
		1.8.4 Validate refund against cancellation	None	None	Validate refund against cancellation()	Validate refund against cancellation screen	Validated refund against cancellation details
		1.8.5 Banking details request	None	None	None	None	None
		1.8.6 Banking details confirmation request	None	None	None	None	None
		1.8.7 Initiate refund reversal	None	None	Cancel booking()	Cancel booking screen	Refund reversal details
		1.8.8 Update cancellation status to "fully refunded"	None	Booking	Update cancellation status()	Update cancellation status screen	Updated cancellation status details
1.9Send cancellation notification	1.9Send cancellation notification	1.9.1 Retrieve updated cancellation status	None	Booking	Search updated cancellation status()	Search updated cancellation status screen	Retrieve updated cancellation status details
		1.9.2 Generate cancellation Confirmation notification email	None	None	Cancel booking()	Cancel booking screen	Generated cancellation Confirmation notification email details
		1.9.3Send cancellation email notification to client	None	None	Send cancelation notification()	Send cancelation notification screen	Sent cancelation notification

ONLINE SUBSYSTEM							
Requirement	Use Case	Process on data flow diagram	Invoke	Entities	Methods	Screens	Reports AND/OR outputs
2.1 Check-in for flight	2.1 Check-in for flight	2.1.1Booking reference number request	None	None	None	None	None
		2.1.2Validate booking reference number	1.5Make payment	Booking	Check-in for flight()	Validate booking reference number screen	Validate booking reference number details
		2.1.3Validate successful check-in	None	None	Check-in for flight()	Check-in for flight screen	Validate booking check-in details
2.2 Generate passenger	2.2 Generate passenger	2.2.1 Search passenger task	None	Booking, Passenger	Search passenger()	Search passenger screen	Search passenger details

indemnity form	indemnity form	2.2.2 Check in passenger task	2.1 Check-in for flight	None	Check-in for flight()	Check in passenger screen	Checked in passenger details
		2.2.3 Generate indemnity form task	None	None	Generate indemnity form()	Generate indemnity form	Indemnity form
2.3 Generate ticket	2.3 Generate ticket	2.3.1 Search passenger details task	None	Booking, Passenger	Search passenger()	Search passenger details screen	Retrieved passenger details
		2.3.2 Check in passenger task	Generate passenger indemnity form	None	Check-in for flight()	Check in passenger screen	Checked in passenger details
		2.3.3 Generate ticket task	None	None	Generate ticket()	Generate ticket screen	Generated ticket
2.4 Refund booking	2.4 Refund booking	2.4.1 Enter reference number on the system task	None	None	None	Enter reference number screen	None
		2.4.2 Locate passenger booking using reference number task	None	None	Search booking()	Locate passenger booking screen	Booking list
		2.4.4 Cancel booking task	1.8 Cancel booking	Booking	Cancel booking()	Cancel booking screen	Cancel booking details
		2.4.5 Update passenger booking list task	None	None	Update booking list()	Update passenger booking list screen	Updated passenger booking list
		2.4.6 Transfer refund money task	None	None	Transfer money()	Transfer refund screen	Transferred refund details
		2.4.7 Send cancellation notification email task	None	None	Refund booking()	Send cancellation notification email screen	Email
2.5 Setup flight schedule	2.5 Setup flight schedule	2.5.1 Display timeslot and trips task	None	None	Display flight schedule()	Display timeslot and trips screen	Flight schedule
		2.5.2 Update schedule task	None	Flight schedule	Update flight schedule()	Update schedule screen	Update schedule
		2.5.3 Capture new flight schedule task	None	None	Setup flight schedule()	Capture new flight schedule screen	Captured new flight schedule
		2.5.4 Display flight schedule task	None	None	Display new flight schedule()	Display flight schedule screen	Flight schedule
2.6 Register pilot		2.6.1 Register pilot task	None	Pilot	Register pilot()	Register pilot screen	Registered pilot details
		2.6.2 Capture pilot task	None	None	Capture pilot()	Capture pilot screen	Captured pilot details
		2.7.1 Booking	None	None	None	Booking screen	None

2.7 Assign pilot to schedule	2.7 Assign pilot to schedule	2.7.2 Load pilot availability details	None	Booking	Load pilot()	Display pilot availability details screen	Load pilot details
		2.7.3 Assign pilot to flight	3.2 Capture balloon inspection and service details	None	Assign pilot to flight()	Assign pilot to flight screen	Assigned pilot to flight details
		2.7.4 Assign pilot a balloon	None	None	Assign pilot to flight()	Assign pilot a balloon screen	Assigned pilot to a balloon
		2.7.5 Send flight booked confirmation	None	None	None	Send flight booked confirmation screen	Notification
2.8 Generate weekend trip schedule	2.8 Generate weekend trip schedule	2.8.1 Weekend schedule	None	None	None	Display weekend schedule screen	Weekend schedule details
		2.8.2 Load booked trips	None	Trip	Display booked trips()	Display booked trips screen	Booked trips
		2.8.3 Load passenger details	None	Passenger	Display passenger details()	Display passenger details screen	Passenger details
		2.8.4 Load pilot details	None	Pilot	Display pilot details()	Display pilot details screen	Pilot details
		2.8.4Load lunch basket details	None	Lunch basket	Display lunch basket details()	Display lunch basket details screen	Lunch basket details
		2.8.5 Generate weekend schedule	None		Generate weekend trip schedule()	Generate weekend trip schedule screen	Flight schedule
2.9 Place deli order	2.9 Place deli order	2.9.1 Load order details	None	Order	Display order details()	Display order details screen	Order details
		2.9.2 Request order confirmation	None	None	None	None	None
		2.9.3 Place order	None	Order	Place deli order()	Place order screen	Invoice
		2.9.4 Send order confirmation	None	None	None	Send order confirmation screen	Order confirmation details
2.10 Reconcile deli order	2.10 Reconcile deli order	2.10.1 Report	None	None	Display report()	Display report screen	Report
		2.10.2 Load order date	None	None	Load order date()	Load order date screen	Order date
		2.10.3 Search order	None	Order	Display order details()	Display order details screen	Order details
		2.10.4 Generate report	None	None	Generate report()	Generate report screen	Report
		2.10.5 Calculate order total	None	None	Calculate order total()	Calculate order total screen	Order Total
		2.10.6 Send order report	None	None	Send order report()	Send order report screen	Sent order report confirmation message

2.11 Pay deli	2.11 Pay deli	2.11.1 Order payment	2.10 Reconcile deli payment	None	Make payment()	Order payment screen	None
		2.11.2 Confirm order	2.10 Reconcile deli payment	None	Reconcile deli payment()	Confirm order screen	Confirmed order details
		2.11.3 Load bank details	None	None	Load bank details()	Load bank details screen	Bank details
		2.11.4 Confirm bank details	None	None	Confirm bank details()	Confirm bank details screen	Confirmed bank details
		2.11.7 Send payment confirmation	None	None	Send payment confirmation()	Send payment confirmation screen	Sent payment confirmation notification
2.12Send client newsletter	2.12Send client newsletter	2.12.1 Retrieve passenger info	None	Passenger	Search passenger info()	Search passenger info screen	Passenger info
		2.12.2 Send Newsletter info	None	Newsletter	Send client newsletter()	Send client newsletter screen	Sent newsletter confirmation message

Ballon Subsystem							
Requirement	Use Case	Process on data flow diagram	Invoke	Entities	Methods	Screens	Outputs AND/OR Reports
3,1 Add balloon	3,1 Add balloon	3.1.1 Add new balloon details	None	Balloon	Add Balloon()	Add Balloon screen	Added ballon details
		3.1.2 Confirm new balloon details	None	None	Confirm new balloon details()	Confirm new balloon details screen	Confirmed balloon details
		3.1.3 Save new balloon details	None	Balloon.	Add Balloon()	Save new balloon details screen	Save balloon details
3,2 Capture balloon inspection and service details	3,2 Capture balloon inspection and service details	3.2.1 Inspection log details request	None	None	None	Inspection log screen	Inspection log details
		3.2.2. Search for balloon details	Invoke use case 3.1 Add Balloon	Balloon	Retrieve balloon()	Retrieve balloon screen	Retrieved balloon details
		3.2.3 Save inspection log details	None	Inspection log	Save inspection log details()	Save inspection log details screen	Save inspection log details
		3.2.4 Generate Inspection log report	None	None	Generate inspection log report()	Generate inspection log report screen	Inspection log report

3,3 Track balloon flight hours	3,3 Track balloon flight hours	3.3.1 Search inspection log details	3.2 Capture balloon inspection and service details	Balloon Inspection Log	Search balloon inspection()	Search balloon inspection screen	Balloon Inspection Log
		3.3.2 Search balloon details	None	Balloon	Search balloon details()	Search balloon details screen	Retrieved balloon details
		3.3.3 Calculate balloon flight hours	None	None	Calculate balloon flight hours()	Calculate balloon flight hours screen	Flight hours
3,4 Generate full inspection report	3,4 Generate full inspection report	3.4.1 Search inspection log details	None	Balloon Inspection Log2	Search inspection log details()	Search inspection log details screen	Retrieved inspection log details
		3.4.2 Display inspection log details	None	None	Display inspection log details()	Display inspection log details screen	Balloon Inspection Log
		3.4.2 Display inspection log details2	None	None	Display inspection log details2()	Display inspection log details2 screen	Balloon Inspection Log

Conclusion

This comprehensive table illuminates the intricacies of the system's processes, leaving no room for ambiguity regarding process initiation. It serves as a robust verification of the correctness of system analysis, confirming the accuracy of each requirement and clearly depicting which use cases, primitive-level processes, entities, methods, screens, and reports align with these requirements. Through this validation process, it guarantees that every aspect has been executed accurately, providing an authentic representation of the system.

6. Conclusion

In conclusion, our journey to develop technical specifications for African Adventures Information System has been marked by dedication, thorough planning, and collaborative teamwork. We've systematically outlined the technical specifications, starting with Use Case diagrams and process models to define user interactions and business processes. Our UML Diagram solidified the system's structure, followed by detailed data modelling to ensure efficient data management. We've also carefully considered inputs and interfaces and emphasized the importance of clear, well-defined outputs. Validation procedures have been scrutinized to guarantee system functionality and data accuracy. Throughout this process, we've maintained detailed records of our progress and fostered effective communication through regular meetings. Our individual signoffs confirm our commitment to this project's success.