INF 271 - Project

Deliverable 1 - Group 6

**1. Cover Page**

**Meet the team!!**

****

**From left to right: [Names ]** Phumzile Phakathi, Busisiwe Khosa, Rethabile Kgotlane, Basetsana Sekhoto, Maria Sadiki

[**Email address:** [u21575755@tuks.co.za](mailto:u21575755@tuks.co.za), [u20473240@tuks.co.za](mailto:u20473240@tuks.co.za), [u21601144@tuks.co.za](mailto:u21601144@tuks.co.za), [u21608611@tuks.co.za](mailto:u21608611@tuks.co.za), [u22651609@tuks.co.za](mailto:u22651609@tuks.co.za) ]

**1.2 Description of the content and purpose of Document**

Welcome to the first leg of our INF 271, Group Project! Over the next few pages, we will explore the African Adventure’s (hot air balloon company) need for an Information System. This document is a nine-part investigation into the current system. African Adventures has and an exploration into what could be. We further go into identifying the preliminary constraints that exist in developing an information system for African Adventures. This is followed by an executive summary and a rich picture of the current system. We will then delve deeper into analysing the user requirements of the new system by means of tables and a well-designed Use Case Diagram. After which we will conduct a feasibility study on three potential ways we can design this system for African Adventures. As we then near the completion of this section of the project we will provide evidence of the progress we made as a team and the meetings we had. The document ends with each team member signing off the document as confirmation of their contribution.

**2.Client Information**

2.1 African Adventures is a new firm that offers hot air balloon rides. They are active in and around Magaliesburg. They've grown so much in the previous seven years that they now need an information system to manage their bookings, on-the-day check-in procedures, and back-office administration.

Mr Hot Flight currently owns African Adventures. He has an office administrator, five technicians, and one foreman working for him. The office administrator oversees all bookings (including phone bookings), meal orders and payments. The foreman is responsible for preparing all of the balloons for the weekend according to the trip plan, as well as managing the inspections after each trip and the overall inspections. The experts help the foreman with balloon preparation and inspections. They also provide balloon maintenance as needed.

**3. Project Request**

3.1 Developing an information system for African Adventures that will help with the management of their bookings, daily checking procedures and administration of the back office.

**4.Preliminary Investigation**

# **4.1 Problem Statement Matrix**

|  |  |
| --- | --- |
| **PROJECT: <African Adventures Project>** | **PROJECT MANAGER: <Busisiwe Khosa>** |
| **CREATED BY: <u22651609>** | **LAST UPDATED BY: <u22651609>** |
| **DATE CREATED: 07/07/2023** | **DATE LAST UPDATED: 28/07/2023** |

# 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Brief Statements of Problem, Opportunity, or Directive** | **Urgency** | **Visibility** | **Annual Benefits** | **Priority or Rank** | **Proposed Solution** |
| 1 Manual booking process whereby the current system relies on phone and walk in bookings which result in potential errors and lack of real-time updates | 3 months | High  (Observable by staff and clients) | Improved customer satisfaction and increased efficiency | 1-Top priority | Create a user-friendly website allowing clients to browse trip options, register and make booking online |
| 2 Handling payments through multiple channels and manual verification delays confirmations and increases the risk of errors. | 5 months | High(Observable by staff and clients) | Improved payment processing speed and enhanced customer experience | 2-Medium | Integrate a web service-based virtual card service for credit/debit card transactions, ensuring secure and real-time payment processing |
| 3 Automate Trip scheduling, because managing the trip schedule using MS excel and manual communication with pilots results in a time-consuming and error-prone process. | 3 months | Moderate (Observable by staff) | Stream-lined scheduling and improved resource utilisation | 2-Medium | Design a system for pilots to indicate their availability online, automatically generating trip schedules based on pilot availability and booking data |
| 4 Automate Indemnity form preparation because manually preparing indemnity forms for each client or passenger on the day of the trip is time consuming and could result in administrative errors | 3 months | High(Observable by staff and clients) | Smoother check-in process | 2-medium | Implement a system to generate indemnity forms with client information automatically when they check-in to reduce manual paperwork and improve administrative efficiency |
| 5 Implement balloon inspection management | 2 months | Moderate  (Observable by staff) | Enhanced safety and improved maintenance planning | 1-Top | Implement a module to record balloon inspections, calculate flight hours, and generate weekly inspection reports for determining full-service requirements |
| 6 Implement newsletter feature | 1 year | Low  (Observable by staff and clients) | Improved customer engagement | 5.Low | Develop a newsletter system to send updates to clients about upcoming specials and events, enhancing customer engagement and retention |
| 7 Integrate Deli order management | 3 months | High  (Observable by staff and clients) | Streamlined refreshment orders and improved vendor relations | 3.Medium | Create a system for placing and managing deli orders, reconciling invoices and handling EFT payments for refreshments and lunch baskets |

# **4.2 Scope of Definition**

* **WHAT IS INCLUDED?**

Online booking system: development of a system that enables clients to browse trip options, register as clients and make booking for hot air balloon trips. It will also allow clients to choose trip date, type of booking and the number of passengers  
  
 Payment and Booking Management: Implement a secure payment gateway to handle credit/debit card transactions for booking payments where clients will be required to make a 50% deposit to secure their bookings and the system will provide payment confirmation emails and the system will also track payment statuses. The system will send reminder emails to clients who only paid the deposit warning them of losing their deposit if they fail to pay the full amount on time.  
  
 Cancellation and refund policy: setting up of an automated cancellation and refund process based on predefined business rules, where full refunds for cancellation made at least two weeks in advance will be handled by the system and the management of refund statuses  
  
 Trip Scheduling: the system will manage the trip schedule which includes the pilot availability and the number of clients booked for each trip and update schedule based on booking.  
  
 Check-in process: the system will streamline the check-in process, allowing the clients to provide their reference number to the office administrator. The system will verify payments, generate indemnity forms, and produce tickets with balloon and pilot information for each trip.  
  
 Balloon inspection and servicing: Design a module to record and manage balloon inspections and servicing. The system will track flight hours, schedule regular inspections, and generate inspection reports.  
  
 Deli food orders and payments: Implement a feature to place and manage orders for lunch baskets and refreshments from the local deli. The system will reconcile deli invoices with orders and handle EFT payments to the deli  
  
 Passenger indemnity forms: The system will automatically generate passenger indemnity forms with client information for easy check-in.

* **WHAT IS EXCLUDED?**  
   Employment management: The system will not handle employment-related processes for office administrators, technicians, and pilots. It will only manage pilot availability for scheduling purposes.

Website design and development: the system scope covers functionality related to booking and management but not the design and development of the website itself.

**4.3 Preliminary Constraints**

Time constraint  
The project needs to be completed within a specific period of time frame to avoid disrupting the current operations of the company. Any delay in implementation may result in inconvenience to clients and staff.

Budget constraint  
 the project must adhere to a predefined budget. cost overruns could limit the resources available for the system’s development and implementation.

Technical expertise  
 The team responsible for developing and implementing the information system must possess the required technical expertise in web development, database management, payment gateway integration, and other relevant technology.

Data security and privacy  
 The system must comply with stringent data security and privacy regulations to protect sensitive client information, especially credit/debit card details and personal data.

Integration with existing system  
 The new system should seamlessly integrate with any existing processes to ensure smooth transition without disruption to day-to-day operations.

User training and adoption  
 Adequate training and support must be provided to staff members to ensure they can effectively use and adopt the new system.

Deli partnership  
 The successful implementation of the system relies on a strong partnership with local deli for lunch basket and refreshment orders. Deli availability and timely deliveries are crucial for seamless customer experience

**5. Problem Analysis**

**5.1 Executive Summary**

Business Description

African Adventures is a Magaliesburg based company that specialises in hot air balloon trips. Their main aim is to provide good services to their clients in an affordable way.

Problem statement

The hot air balloon company is facing operational inefficiencies caused by the current manual processes for bookings, payments and balloon inspections. Bookings are captured via phone or walk-ins, this may lead to potential errors and challenges tracking payments. Scheduling trips relies on manual coordination with pilots, causing scheduling conflicts. Catering orders are managed manually, risking errors in quantities and deliveries. Passenger check-ins and indemnity forms are paper documents this may result in delays and administrative burden. Balloon inspections are manually recorded, potentially risking safety and compliance issues.

The proposed solution

The proposed new information system will be used to manage the company’s bookings, day-to-day operations and the back-office administrations.

The opportunity

With the new system in place, the company will be able to improve their customer services and provide their customers with a better experience, as a result they will be able to realise a growth in their revenue.

**5.2 Overview of the current system**

African Adventures system

The office administrator serves as a central figure in the day-to-day operations of African Adventures. Clients can make bookings and payments either by phone or by visiting the office in person. The office administrator records these details in a diary kept at the reception desk. Payments can be made in cash or by credit card, and for in-person payments, the office administrator issues an invoice with trip details and provides a receipt to the client. If clients pay via electronic funds transfer (EFT), they are required to send proof of payment, which the office administrator verifies against the company's bank statement. Once verified, the receipt is issued, and the client is contacted to confirm the payment and booking. Invoices and receipts are safely kept and handed to clients on the day of their trip.

Confirmed bookings are highlighted in green in the diary, while cancelled bookings are crossed out, and refundable bookings are marked in red. Once refunds are processed, they are ticked off. The current trip prices are displayed on the reception wall to assist the office administrator in calculating booking amounts. Informing clients about final payment due dates and calculating refund amounts are handled by the office administrator.

Setting up trip schedules relies on the availability of pilots. The office administrator sends out an MS Excel file to each pilot, where they indicate their availability. Upon receiving the completed workbook, the office administrator reconciles the information and finalises the schedule, informing pilots of their confirmed trip dates.

Every Friday morning, the office administrator checks the number of lunch baskets and breakfasts booked for the upcoming weekend and places orders with the local deli. The deli delivers the meals on the day of the trip and sends a monthly statement for all orders placed during the month, which the office administrator pays via EFT.

On the day of the trip, the office administrator prepares a list of all passengers and provides each client with a blank indemnity form to complete. Clients show their receipts or proof of payment to the office administrator, who then marks them off the passenger list.

All balloon inspections are recorded in a logbook by a foreman. To determine if a balloon is due for service and if it can still be used for a trip, the foreman calculates flight hours, which are also recorded in the logbook.

**5.3 Detailed recommendation**

PROBLEMS, OPPORTUNITIES, OBJECTIVES AND CONSTRAINTS MATRIX

|  |  |
| --- | --- |
| Project: African Adventures System | Project manager: B Khosa |
| Created by: R. N Kgotlane | Last Updated by: R. N Kgotlane |
| Date Created: 16 July 2023 | Date Last Updated: 28 July 2023 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CAUSE AND EFFECT ANALYSIS | | SYSTEM IMPROVEMENT OBJECTIVES | | |
| Problem or Opportunity | Causes and Effects | System Objective | System Constraint |  |
| 1. Access to client’s information | - No proper restriction on who can access the diary.  - Clients’ information can be stolen. | - Add restrictions on who can access the client's information. | - New system should be secured to protect the client's information from unauthorised access. |  |
| 2. Duplication of client information. | - Unable to make records.  - Dealing with more clients there are less chances of noticing duplication of client’s information. | - Provide a single database for clients. | - For data to be managed effectively, the new system must contain the client's database if the budget is sufficient. |  |
| 3. Implement newsletter features | - Improve client’s engagement.  - Able to attract more clients.  - Keep clients updated. | - Updates clients in a timely manner. | - The new system should keep clients updated, by sending newsletters. |  |
| 4. Increase the number of air balloons | - Economically able to make more money and satisfy more customers. | - To be able to provide services to more clients. | - The new system should allow more balloons to be added. |  |

**5.4 Rich Picture**

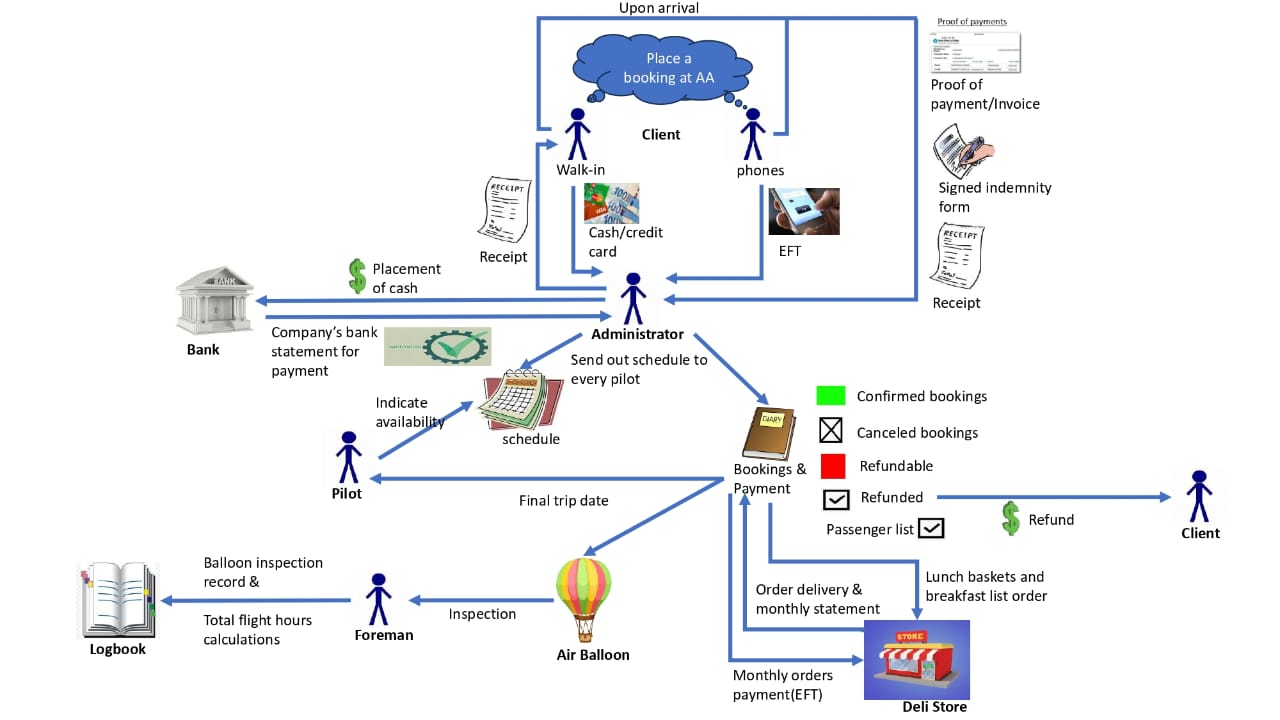
****

Figure Shows day-to-day operations of the organization in the current system.

**6. Requirements Analysis**

This section focuses on the user requirements also known as “user needs” which are a critical analysis of what the users expect from the system, these include the features, functionalities, and lastly the characteristics of the system.

**6.1 List of User Requirements:**

**Booking subsystem**

1.1 Book trip

1.2 Add lunch basket.

1.3 Send booking confirmation.

1.4 Cancel booking.

1.5 Update booking list.

1.6 Update booking status.

1.7 Log in.

**Client Subsystem**

2.1 Register client.

2.2 Capture client debit info.

2.3 Update client.

2.4 Search client.

2.5 Send newsletter.

**Payment Subsystem**

3.1 Send a payment confirmation email.

3.2 Send payment cancellation email.

3.3 Send a reminder email.

3.4 Make a deposit payment.

3.5 Make final payment.

**Balloon subsystem**

4.1 Complete inspection report.

4.2 Record all inspection logs.

4.3 Generate weekly inspection report.

4.4 Add Balloon.

**Check-in subsystem**

5.1 Check in.

5.2 Generate indemnity form and ticket.

5.3 Check payment status.

**Pilot Subsystem**

6.1 Book availability.

6.2 Remove availability**.**

**Order Subsystem**

7.1 Place lunch basket order.

7.2 Pay monthly order.

7.3 Capture payment.

7.4 Search order.

7.5 Generate order report.

**6.2 Requirements description and details (Functional and Non-Functional)**

**System Requirement Description and Detail – African Adventures.**

**Functional Requirements**

1. **BOOKING SUBSYSTEM**

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 1.1 |
| Requirement title: | Book trip |
| Requirement text: | The system must allow the user to make a booking for a trip on behalf of the client. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system must be able to capture the booking details |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system must have input values for making a booking whenever a client comes in to make a booking. The booking must be flexible and allow for trips to be booked for specific trips and schedules. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 1.2 |
| Requirement title: | Add Lunch basket |
| Requirement text: | The system must allow the user the option to include a lunch basket as part of their booking. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system should have the lunch basket option as part of the booking process of the trip and the option is only available for the four-hour trips and the full-day trips. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | It is critical that the user provides the user with an input button for adding a lunch basket to the clients order. The lunch basket must be added to the booking of the trip. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 1.3 |
| Requirement title: | Send booking confirmation. |
| Requirement text: | The system must send a booking confirmation email to the client. |
| Requirement type: | Functional |
| Requirement details and constraints: | The booking confirmation email must be sent to the client must the booking process is done and their booking status has been changed to successful. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system must be able to send a confirmation email only to clients who have made a booking and the booking has been confirmed and verified by the client. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 1.4 |
| Requirement title: | Cancel booking |
| Requirement text: | The system should allow the office administrator to cancel a booking on behalf of the customer. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system should have an option for cancelling a booking in case a client wants to cancel their booking. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system must be able to allow the user to cancel a booking and thus, an option for cancelling the booking must be displayed on the screen, for example, a button. This is to accommodate clients who do not wish to complete their booking. The client’s deposit is forfeited. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 1.5 |
| Requirement title: | Update booking list |
| Requirement text: | The system must only allow the booking list of clients to be updated. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system should allow the booking list to be updated whenever a booking is added or cancelled by the client. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system should allow the booking list to be updated whenever a booking is added or cancelled by the client by allowing the user to change the input values of the clients’ booking and reflect the changes whenever a booking is added or deleted from the system. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 1.6 |
| Requirement title: | Update booking status |
| Requirement text: | The system must display the booking status for each booking to show the progress of the booking. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system must show relevant information about the booking status and be able to show booking statuses in the required period. The booking status must automatically be updated in real-time. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The booking status must change accordingly within five seconds whenever it has been updated. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 1.7 |
| Requirement title: | Login |
| Requirement text: | The system must allow the user to log-in to the system when they want to make a booking or make use of the system. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system should ensure that only users registered as clients are allowed to use the system by making using of unique passwords to access the system |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system must verify the user’s login details before allowing access and forbid access to the system if the user is not registered on the system. |

1. **CLIENT SUBSYSTEM**

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 2.1 |
| Requirement title: | Register Client |
| Requirement text: | The system must allow the user to register themselves as a client on the website. |
| Requirement type: | Functional |
| Requirement details and constraints: | The computer must be able to allow the user to fill in their details and successfully be registered. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system must capture the new client’s registration details. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 2.2 |
| Requirement title: | Capture card information |
| Requirement text: | The system should be able to capture the information of the client’s card for record keeping. |
| Requirement type: | Functional |
| Requirement details and constraints: | The card information as well as all previous transactions should be kept in the system for future use. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The card information must be captured on the system for future reference. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 2.3 |
| Requirement title: | Update client information |
| Requirement text: | The system should be able to update the information of the client. |
| Requirement type: | Functional |
| Requirement details and constraints: | The client information should be kept in the system for future use. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system must be able to reflect the changes of the client’s updated information within 10 seconds. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 2.4 |
| Requirement title: | Search client |
| Requirement text: | The system should be able to search for client information. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system should allow the search for client information regardless of whether the client exists on the system or not to allow for registered clients to be registered. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system must be able to retrieve the client’s information within 10 seconds. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 2.5 |
| Requirement title: | Send newsletter |
| Requirement text: | The system must be able to send newsletters to clients keeping them up to date. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system must be able to send newsletters to clients via email. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | Newsletters must be sent directly to clients email addresses to keep them updated. However, if a client unsubscribes to receiving the newsletters the system must no longer send newsletters to clients. |

1. **PAYMENT SUBSYSTEM**

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 3.1 |
| Requirement title: | Send a payment confirmation email |
| Requirement text: | The system must be able to send a payment confirmation email to clients. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system must only send a payment confirmation email to clients who have paid for their booking. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The confirmation email must be sent immediately after the payment has been made by the client. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 3.2 |
| Requirement title: | Send a payment cancellation email |
| Requirement text: | The system must be able to send a payment cancellation email to clients. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system must only send a payment cancellation email to clients who have cancelled their booking. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The cancellation email must be sent immediately after the payment has been cancelled by the client confirming their cancellation. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 3.3 |
| Requirement title: | Send a reminder email |
| Requirement text: | The system must be able to send a reminder email to clients. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system must be able to send out a reminder email to all clients who only paid the 50% deposit, a week prior to the closing date. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The reminder email must be sent to all client exactly a week prior to the closing date but ONLY to clients who have paid more than 50% of the deposit. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 3.4 |
| Requirement title: | Deposit Payment |
| Requirement text: | The system must be able to allow the user to pay a deposit amount which is 50% of the total amount. |
| Requirement type: | Functional |
| Requirement details and constraints: | The computer must allow half of the full amount payments without restricting the user from making a booking. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system must only allow a deposit amounting to R3250/person for a full day trip. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 3.5 |
| Requirement title: | Make Full payment |
| Requirement text: | The system must be flexible enough to allow the user to make full payment after making the 50% deposit payment. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system must automatically calculate the outstanding amount that the client has to pay in order for the full amount to be recorded. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system will only allow a payment amounting to R6500/person for a full day trip. |

1. **Balloon Subsystem**

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 4.1 |
| Requirement title: | Adding Balloons |
| Requirement text: | The system must be able to allow more balloons to be added in the future. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system must be very flexible and allow for the update of the information of the balloons or the addition of new balloons to be done by the authorised user. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system will give the option to add a new row for a balloon and the new balloon will be added at the bottom of the list of the balloons that are already in the system. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 4.2 |
| Requirement title: | Complete Inspection report |
| Requirement text: | The system must be able to allow the foreman to complete inspection reports. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system must have user input fields that guide the foreman on what derails to provide in order for the inspection repots to be accurate and complete. Invalid values are immediately rejected. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The current fuel capacity field on the inspection report must only allow a minimum of 70 gallons of fuel for a 2 hour trip. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 4.3 |
| Requirement title: | Recording Inspection logs |
| Requirement text: | The system must allow the owner to record all the inspection logs on the system after each inspection has been completed. |
| Requirement type: | Functional |
| Requirement details and constraints: | All inspection logs are to be recorded on the system and stored electronically. All the inspection reports must meet the minimum requirements for the to be processed successfully. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system must be able to validate the inspection reports before they are being processed and show an error message “Invalid input please try again” if there are any violations relating to incorrect input. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 4.4 |
| Requirement title: | Generate weekly inspection report |
| Requirement text: | The system must generate weekly inspection reports that have been approved. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system must provide a detail report that is easy to read and clear. Clearly stating what services need to be done on each balloon and also provide possible costs associated with each of the services and maintenance. Separate reports must be generated for each balloon. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system must have colour codes for each of their inspection report statuses. For instance, if a hot air balloon is in bad condition the report should highlight all the high-risk areas in red and they should be at the top of the report. |

1. **Check-in Subsystem**

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 5.1 |
| Requirement title: | Checking - in |
| Requirement text: | The system must allow the use of the client reference number (generated by the system) to check for the booking status of the client and to check if full payment has been made in order for them to be checked in. |
| Requirement type: | Functional |
| Requirement details and constraints: | The office administrator uses the reference number that the client provides to check them in on the day of the trip and also use it to check if they fully paid. The system should immediately reject any incorrect reference numbers. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The check-in status reads “Successful” when reference and full payment has been validated. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 5.2 |
| Requirement title: | Generate indemnity form and ticket |
| Requirement text: | The system must be able to print out a complete and accurate indemnity form and ticket for the client. |
| Requirement type: | Functional |
| Requirement details and constraints: | When the administrator enters the reference number, the system must be able to allow only the predetermined number of characters of the reference number for the process to continue. The system must generate a complete indemnity form that has the details of the assigned pilot and the balloon number |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system must be able to print out a copy of the ticket and the indemnity form and store the electronical form. The printed out document must have the reference number and the pilot name at the top, clearly visible. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 5.3 |
| Requirement title: | Check Payment Status |
| Requirement text: | The system should be able to display the payment status of the client before they are checked in. |
| Requirement type: | Functional |
| Requirement details and constraints: | The administrator will need to reference the number of the client in order for them to retrieve all the information reading the client's trip and payments. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The payment status must be displayed at the top of the screen so that the administrator can see it first before proceeding with the transaction. |

**6. Pilot Subsystem**

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 6.1 |
| Requirement title: | Booking availability |
| Requirement text: | The system must allow the pilots to book their availability in their shift management a month in advance. |
| Requirement type: | Functional |
| Requirement details and constraints: | The pilots need to ensure that they log-In using their credentials so that the shift that they book for does not get assigned to someone else. The system must lock the shift once it has been so that no one else can have access to it. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system will automatically grey out the booked shift to ensure that no one else books it. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 6.2 |
| Requirement title: | Remove Availability |
| Requirement text: | The system must be flexible in allowing for the pilot to remove their availability if they can’t make it for the shift anymore.. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system should ensure that only the pilot or the administrator upon authorization is able to remove the name of a pilot for a shift they had previously booked for without another shift being altered or affected. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system will remove the availability of a pilot and update the rest of the records that are affected by the removed availability without malfunctioning. |

**7. Order Subsystem**

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 7.1 |
| Requirement title: | Placing a lunch basket order |
| Requirement text: | The user must be able to add a lunch basket during the booking of their trip. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system must be able to allow the user (client/admin) to add a lunch basket as part of their trip booking. The option to choose or not to choose a lunch basket be a mandatory step since the should separate the bookings with a lunch basket from those that do not have a lunch basket. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system should have an option to either add a lunch basket or book the trip without the lunch basket as a button at the end of the screen. The button must be visible enough for the user to see. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 7.2 |
| Requirement title: | Paying of Monthly order to suppliers |
| Requirement text: | African adventures must be able to make their monthly payments to their suppliers using the system. |
| Requirement type: | Functional |
| Requirement details and constraints: | The system must allow the admin to make all the required orders from the authorised supplier list and only pay at the end of each month without the ordering system being disrupted. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system will not display the payment option anytime before month end, to avoid any errors that might be made by the admin (for instance, the admin paying the suppliers before creditors reconciliations are made). |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 7.3 |
| Requirement title: | Capture Payment |
| Requirement text: | The system must allow the user (Admin/Owner) to capture all the payments made for the month on the system and back-ups to be stored electronically. |
| Requirement type: | Functional |
| Requirement details and constraints: | The captured payments must be very detailed (all details of the suppliers, name, surname etc) and carefully recorded to avoid making incorrect payments or making payments to the wrong accounts. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system must sort the captured payments in categories of emergency (The most outstanding payment must be at the top of the list so that it could be paid first). |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 7.4 |
| Requirement title: | Search for Order |
| Requirement text: | The system must all the admin to search for an order that is stored on the system with all its related information. |
| Requirement type: | Functional |
| Requirement details and constraints: | A search order dropdown menu that allows the admin to either select the order from the list of the dropdown menu or have the option to enter the order number and filter the order list to retrieve accurate results. Orders need to be in the system for them to appear in the search results. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | All order numbers generated by the system must be ordered in ascending order starting with the least recent order at the top. The entered order number in the “Search order” input field must display the correct order (order numbers are unique). |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 7.5 |
| Requirement title: | Generate monthly order report |
| Requirement text: | The system must be able to allow the owner to generate a monthly order report. |
| Requirement type: | Functional |
| Requirement details and constraints: | The generated order report should have all the information that was recorded, and the system must be able to present the report in a formal manner that is easy to read and navigate without compromising on the details of the information that is needed. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system must make use of graphical applications to represent the cost of the orders made by African adventures and make a comparison between the suppliers. |

**Non-Functional Requirements (PIECES)**

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 1.1 |
| Requirement title: | Security over client information |
| Requirement text: | The system must verify the credentials of the user before allowing them to access the system to ensure that no unauthorised use of the system is allowed. |
| Requirement type: | Non-Functional |
| Requirement details and constraints: | The verification of the credentials should be done against the stored one on the system to check if they match and all invalid credentials are immediately rejected. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system must always ask for the password of the user on the starting page of every transaction/activity before displaying any other information |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 1.2 |
| Requirement title: | Payments captured on the system |
| Requirement text: | The system must capture all payments (made by the clients to the African Adventures and payments made by African Adventures to their suppliers) immediately after the transaction occurred. |
| Requirement type: | Non-Functional (PIECES) |
| Requirement details and constraints: | All the details of the payments must be captured and recorded in the accounting books of the African Adventures and the system must be able to make immediate updates upon receiving of payments from clients or payments sent out by African Adventures. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | Only complete information can be processed. The user input fields that require the details of the supplier for instance must be at the top of the Payment screen. |

|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 1.3 |
| Requirement title: | Retrieving the client’s information |
| Requirement text: | The system must be able to instantaneously retrieve the information of the client as their reference code is entered. |
| Requirement type: | Non-Functional (PIECES) |
| Requirement details and constraints: | The system should display the client’s complete information without any missing details. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The client information must be retrieved within 3 seconds. |

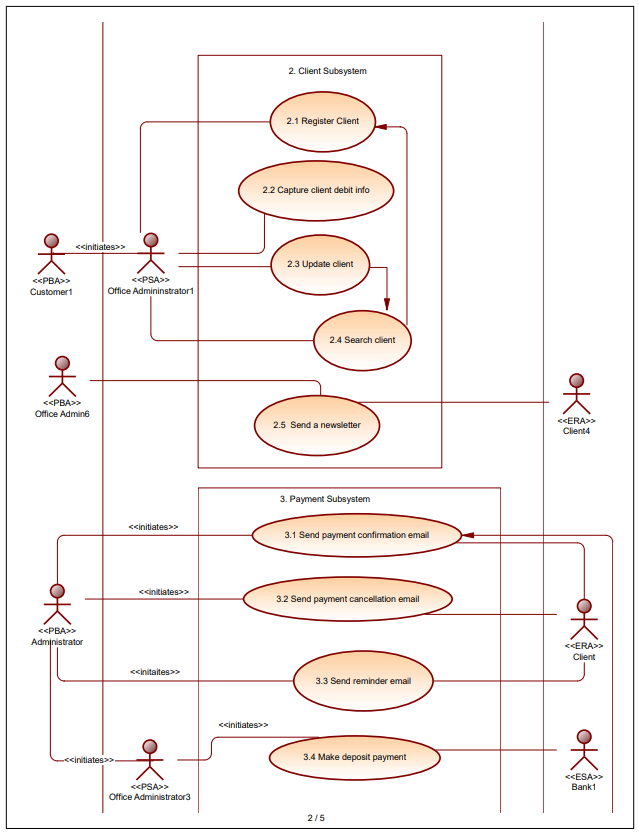
|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 1.4 |
| Requirement title: | Multiple user logins |
| Requirement text: | The system must be able to handle at least 2000 users at the same time without delaying the response time. |
| Requirement type: | Non-Functional (PIECES) |
| Requirement details and constraints: | All the users on the system must be able to access all services of the system without the system crashing. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system should have a Wi-Fi that has a signal strength of about -50 dBm to handle all the users on the website. |

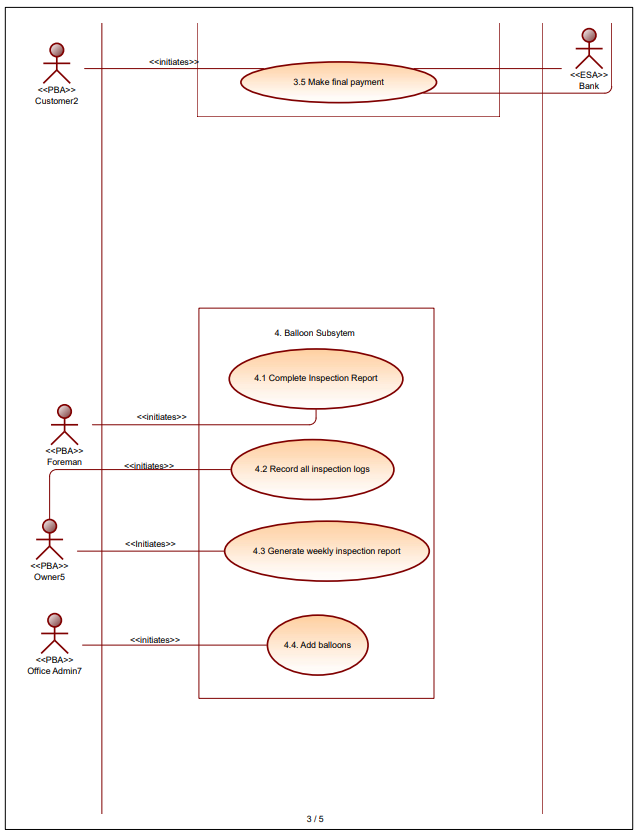
|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 1.5 |
| Requirement title: | Access control |
| Requirement text: | The system should have a control put in place to restrict certain functionalities to unauthorised users and have clear error messages prompting the user to take corrective action. |
| Requirement type: | Non-Functional (PIECES) |
| Requirement details and constraints: | All the users on the system must be able to access all services (when authorised) of the system without the system crashing. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system should display the error message within 3 seconds after an attempt to access unauthorised facilities. |

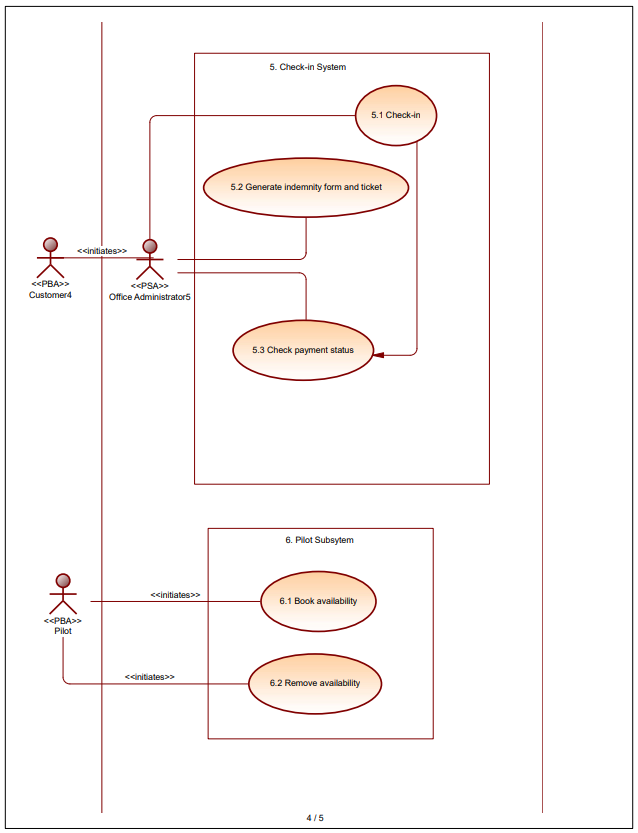
|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 1.6 |
| Requirement title: | Fast Response Time |
| Requirement text: | The system can make use of caching to improve the response time when requests are made on the system. |
| Requirement type: | Non-Functional (PIECES) |
| Requirement details and constraints: | Sensitive data like passwords should not be stored in cache as this might pose a threat to the security of the client’s information. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | Whenever a request is made on the system, the system must respond withing 0.35 seconds. For instance, logging on the website should take 0.35 seconds for the access to be granted. |

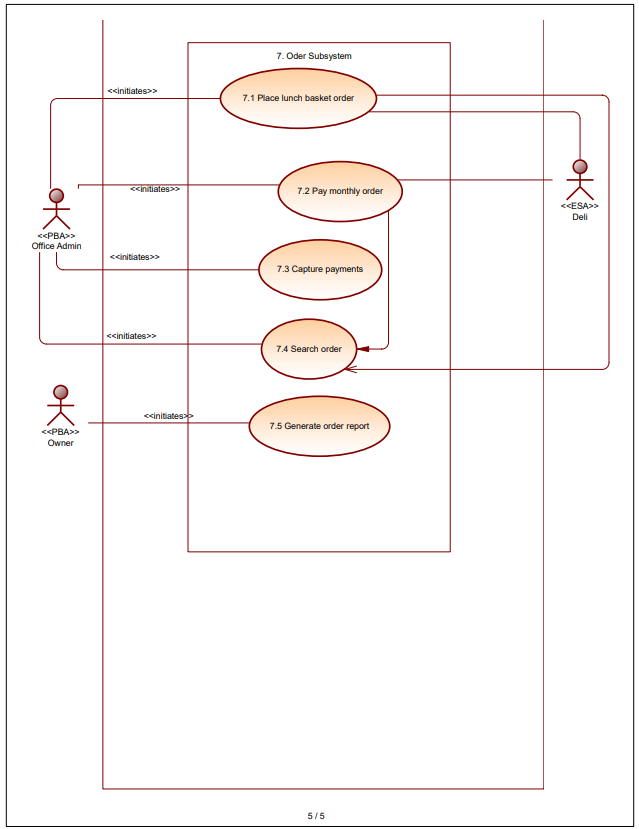
|  |  |
| --- | --- |
| Requirement | Explanation |
| Requirement number: | 1.7 |
| Requirement title: | Website Services |
| Requirement text: | The services offered by African Adventures must be available for the user to view all the time. |
| Requirement type: | Non-Functional (PIECES) |
| Requirement details and constraints: | The system must only display information of their services so that the clients are well aware of what they are booking a trip for. |
| Revision date and Revision number: | Version 1.0 |
| Criticality/Priority: | The system must have an AI chatbot that will handle customer complaints 24/7 every day of the week. |

**6.3. Requirements level Use Case**

****

****

****

****

**6.4 Every requirement properly identified/numbered for later reference.**

**Booking subsystem**

1.1 Book trip.

1.2 Add lunch basket.

1.3 Send booking confirmation.

1.4 Cancel booking.

1.5 Update booking list.

1.6 Update booking status.

1.7 Log in.

**Client Subsystem**

2.1 Register client.

2.2 Capture client debit info.

2.3 Update client.

2.4 Search client.

2.5 Send newsletter.

**Payment Subsystem**

3.1 Send a payment confirmation email.

3.2 Send payment cancellation email.

3.3 Send a reminder email.

3.4 Make a deposit payment.

3.5 Make final payment.

**Balloon subsystem**

4.1 Complete inspection report.

4.2 Record all inspection logs.

4.3 Generate weekly inspection report.

4.4 Add Balloon.

**Check-in subsystem**

5.1 Check in.

5.2 Generate indemnity form and ticket.

5.3 Check payment status.

**Pilot Subsystem**

6.1 Book availability.

6.2 Remove availability**.**

**Order Subsystem**

7.1 Place lunch basket order.

7.2 Pay monthly order.

7.3 Capture payment.

7.4 Search order.

7.5 Generate order report.

**Non-Functional Requirements (PIECES)**

* 1. The system must verify the credentials of the user before allowing them to access the system to ensure that no unauthorised use of the system is allowed.
  2. The system must capture all payments (made by the clients to the African Adventures and payments made by African Adventures to their suppliers) immediately after the transaction occurred.
  3. The system must be able to instantaneously retrieve the information of the client as their reference code is entered.
  4. The system must be able to handle at least 2000 users at the same time without delaying the response time.
  5. The system should have a control put in place to restrict certain functionalities to unauthorised users and have clear error messages prompting the user to take corrective action.
  6. The system can make use of caching to improve the response time when requests are made on the system.
  7. The services offered by African Adventures must be available for the user to view all the time.

**7.Decision Analysis**

The next component of our Project will consider three potential ways that we can develop the Information System for African Adventures. It narrows the alternatives down to one ultimate system. By means of careful analysis of each system’s feasibility, on the basis of economic, technical, operational, and schedule factors. And with the use of the feasibility Matrix.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feasibility Criteria** | **Wt.** | **Candidate 1** | **Candidate 2** | **Candidate 3** |
| **Description** |  | **Develop an Information System by ourselves and with our own expertise Using Visual Studio and SQL Management Studio** | **Purchase an Off -The-Shelf Information System from a Vendor** | **Combination System -Purchase general system [SetMore]from Vendor and customise it according to User requirements** |
| **Operational Feasibility**    **Functionality**. A description of to what degree the candidate would benefit the organisation and how well the system would work.    **Political**. A description of how well received this solution would be from both user management, user, and organisation perspective. | **30%** | **Functionality**  **Fully supports Users System Requirements**  **This Candidate will allow clients to book a trip via the website**  **This Candidate will allow clients to register on the website**  **This Candidate will allow clients to make virtual payments of bookings**    **Political**    **This system will greatly reduce the time spent on walk-in/phone-in orders made .Since all processes related to making a booking for balloon trips will occur online .**  **Pilots will be able to better schedule their flight hours as they will have 24 access to a digital flight schedule in which they may select the slots in which they are available instead of the process of having to go into the office every time to write down flight hours**  **Processing client payments will be made easier and customers will no longer have to go through the extra work off sending proof of payment to African Adventures as most payments will occur online and payment will be confirmed automatically**  **Delays on the morning of Balloon trips will be greatly reduced as indemnity forms will be auto generated for the customer on the morning of the trip**  **Orders to the deli will be sent via the system as soon as client confirms basket selection**  **Score: 95** | **Functionality**  **Partially supports User Requirements, due to it being a generic system, security may also be a concern.**  **This Candidate will allow clients to book a trip via the website**  **This Candidate will allow clients to register on the website**  **This Candidate will allow clients to make virtual payments of bookings**  **Political**      **This system will greatly reduce the time spent on walk-in/phone-in orders made .Since all processes related to making a booking for balloon trips will occur online .**    **Pilots will be able to better schedule their flight hours as they will have 24 access to a digital flight schedule in which they may select the slots in which they are available instead of the process of having to go into the office every time to write down flight hours**  **Processing client payments will be made easier and customers will no longer have to go through the extra work off sending proof of payment to African Adventures as most payments will occur online and payment will be confirmed automatically**  **Delays on the morning of Balloon trips will be greatly reduced as indemnity forms will be auto generated for the customer on the morning of the trip**  **Orders to the deli will be sent via the system as soon as client confirms basket selection**  **Score: 60** | **Functionality**  **Fully supports Users System Requirements, due to the customizations it will make.**  **This Candidate will allow clients to book a trip via the website**  **This Candidate will allow clients to register on the website**  **This Candidate will allow clients to make virtual payments of bookings**  **Political**      **This system will greatly reduce the time spent on walk-in/phone-in orders made .Since all processes related to making a booking for balloon trips will occur online .**    **Pilots will be able to better schedule their flight hours as they will have 24 access to a digital flight schedule in which they may select the slots in which they are available instead of the process of having to go into the office every time to write down flight hours**  **Processing client payments will be made easier and customers will no longer have to go through the extra work off sending proof of payment to African Adventures as most payments will occur online and payment will be confirmed automatically**  **Delays on the morning of Balloon trips will be greatly reduced as indemnity forms will be auto generated for the customer on the morning of the trip**  **Orders to the deli will be sent via the system as soon as client confirms basket selection**    **Score: 95** |
| **Technical Feasibility**    **Technology**. An assessment of the maturity, availability (or ability to acquire), and desirability of the computer technology needed to support this candidate.    **Expertise**. An assessment of the technical expertise needed to develop, operate, and maintain the candidate system. | **30%** | **Technology**  **Development of the System will occur in Visual Studio in tandem with SQL Management Studios Database structure which are languages we as a team are familiar with**    **Expertise**  **As second year IT-Students we are familiar with both applications and will just learn more on linking the two software’s.**          **Score: 85** | **Technology**  **Perl 5 and Java are the most used programming languages on most Booking applications.**    **Expertise**  **This candidate will require our team to receive training in the language to develop competence.**          **Score:60** | **Technology**  **Setmore is a bookings and payments application which can be further customised post purchase.**  **Expertise**  **It is relatively easy to learn to use and may require a few training sessions for staff to gain confidence with the application.**          **Score: 70** |
| **Economic Feasibility**    **Cost to develop:**    **Payback period (discounted):**    **Net present value:**    **Detailed calculations:** | **30%** | **Approx.: R0**    **Approx.:4 years**        **Score:90** | **Approx.: R550 000**    **Approx.:6 years**    **Approx.: R250 000**    **Score: 60** | **Approx.: R160 per user/month**    **Approx.:5 years**    **Approx.: R 70 000**      **Score: 70** |
| **Schedule Feasibility**    An assessment of how long the solution will take to design and implement. | **10%** | **3 months**    **Score: 85** | **6 months**    **Score: 45** | **4 months**    **Score: 55** |
| **Links to information per column-** |  | **Systems Analysis and Design for the Global Enterprise. 7th Edition. [Page 412]** | **https://medium.com/booking-com-development/a-faster-perl-runtime-in-tiny-steps-97a1191d75e5** | **Setmore-https://www.setmore.com/?ref=ext\_admitadpartnernetwork3\_f5e999521926b0db2cde24141ecf659c\_245332&tm\_uid=f5e999521926b0db2cde24141ecf659c&tm\_publisher=245332&tm\_website=520770** |
| **Ranking:** | **100%** | **89.5%** | **58.5%** | **76%** |

**7.2.3 Recommendations from feasibility analysis**

**7.2.3.1 Operational Feasibility**

Ours is the best solution.

Based on the recommendations obtained from the feasibility analysis, with our own system scoring a 95% feasibility; we will proceed to develop our own Information System for African Adventures. This is because our system fully meets user requirements which amongst other things include Booking, Check-in, Pilot Management, Order and Payment functionalities. Candidate 2 will not fully meet all these specificalities as it is a generic system and Candidate 3 requires the extra effort of customising a system we have already purchased.

**7.2.3.2 Technical Feasibility**

Regarding Technical feasibility, our own system, and the combination system scored higher due to them having the least amount of new training or skills requirements.

**7.2.3.3 Economic Feasibility**

Since Candidate 1 will be of no cost to our company, it far outweighs candidate 2 and 3, since they both include monthly subscriptions. Therefore **Candidate 1** is the solution we will continue to develop.

**8. Project Progress**

**8.1** <https://u20473240inf271prac.atlassian.net/wiki/spaces/IG0A/pages/7045137/Meeting+notes+in+space>

**8.2**<https://u20473240inf271prac.atlassian.net/wiki/spaces/IG0A/pages/6815976/2023-06-26+Meeting+1>

**8.3 Problems and Issues (encountered in current deliverable/phase and solution)**- Lack of communication  
- Time management during meetings  
- Needing further clarity on some parts of the project  
- Loadshedding during virtual meetings (google meet)

**9.Sign Off**

By completing the next part of our document each student acknowledges that they contributed to this document and formally signs it off.

|  |  |
| --- | --- |
| **Student Info** | |
| First Name | Busisiwe |
| Last Name | Khosa |
| Student No. | u20473240 |
| Date | 31/07/2023 |
|  | |
| **Student Info** | |
| First Name | Maria |
| Last Name | Sadiki |
| Student No. | u22651609 |
| Date | 31 /07/2023 |

|  |  |
| --- | --- |
| **Student Info** | |
| First Name | Basetsana |
| Last Name | Sekhoto |
| Student No. | u21608611 |
| Date | 31/07/2023 |

|  |  |
| --- | --- |
| **Student Info** | |
| First Name | Rethabile |
| Last Name | Kgotlane |
| Student No. | u21601144 |
| Date | 31 /07/2023 |
|  | |

|  |  |
| --- | --- |
| **Student Info** | |
| First Name | Phumzile |
| Last Name | Phakathi |
| Student No. | 21575755 |
| Date | 31/07/2023 |