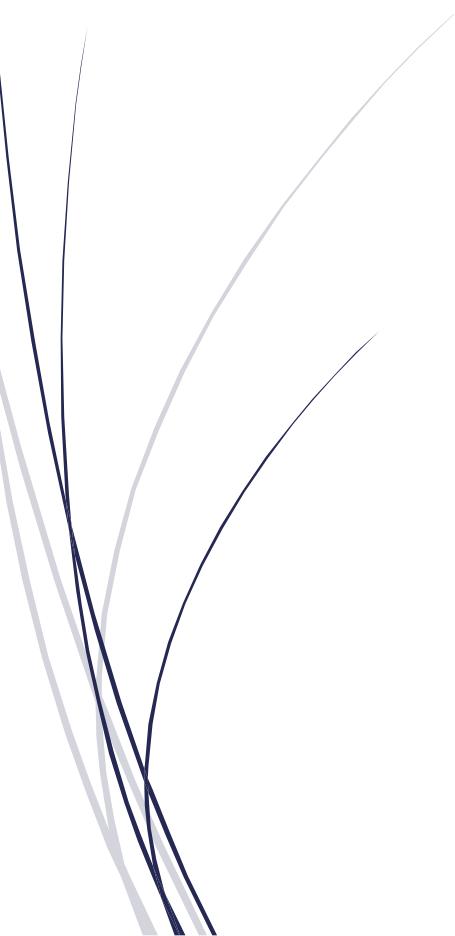




3/5/2023

Systems Specifications and Design Report



Systems Specification and Design

Version: 1.4

History

Version	Note	Date	Author
1.0	Initial draft. Current Systems description, proposed systems specifications and design results	1 Mar 2023	DU
1.1	Diagrams and tables editing	03.03.2023	DU
1.2	Revised draft, following group's review	03.03.2023	DU
1.4	Revised Finals	04.03.2023	DU

Table of Contents

1	INTRODUCTION	3
1.1	PURPOSE AND SCOPE OF REPORT	3
2	REQUIREMENTS SPECIFICATION.....	4
2.1	EXISTING SYSTEM.....	4
2.1.1	<i>General Requirements.....</i>	4
2.1.2	<i>Existing Implementation and Limitations.....</i>	4
2.2	USE CASE DIAGRAM	5
2.3	USE CASE SPECIFICATIONS	7
2.4	USE CASE INDEXED LIST	20
3	SYSTEM DESIGN	25
3.1	DESIGN CLASS DIAGRAM.....	25
3.2	ER DIAGRAM.....	27
3.3	DDL STATEMENTS.....	28
3.4	DML STATEMENTS (SQL)	36
3.5	GUI DESIGN	38
4	APPENDIX.....	49
4.1	TABLE OF FIGURES.....	49
4.2	TABLE OF TABLES	49

1 Introduction

This document contains the design specification requested by AirVia Ltd for the development of a new airline ticket booking and reports software. AirVia Ltd is an international airline that sells flight tickets through local travel agents. AirVia Ltd is looking to develop a new Air Ticket Sales (ATS) system including upgraded maintenance features to replace an out-of-date existing system with modern software. This is to make use of software developments that have occurred over the years and improve the efficiency and experience for both customers and the airline. Our DreamSoft team has been contracted to design and develop the software. The below report details the specification and design that the team have developed.

1.1 Purpose and Scope of Report

The scope of this report contains Requirements Specification and Design, which consist of:

- Requirements Specification
 - Description of the existing ATS system and its limitations
 - Motivations for design of the new system
 - Use case diagram defining the full functionality of the new system
 - Use case specifications
 - Indexed list of all use cases
- System Design
 - Design class diagrams including:
 - Control, entity and boundary classes
 - Attributes and methods.
 - Set of operations including parameter lists, return types, exceptions, visibility of set and get operations
 - Constructors and destructors.
 - Database connectivity, architecture and interfaces
 - ER diagram, DDL statements representation, SQL DML statements
 - GUI designs for visual understanding , GUI diagram mapped to the boundary classes represented in the diagram, visual look of user navigation.

2 Requirements Specification

2.1 Existing System

2.1.1 General Requirements

Airvia require travel agents that sell tickets on their behalf to record these sales using their ATS software. It must support bookings using the bespoke ticket system that Airvia use and the ability to apply discounts and to these sales where relevant. This software is also used to produce reports for the Airvia financial department.

The airline uses a ticketing system where tickets are represented by objects known as *blanks*. These blanks are bundles of flight coupons and auditor coupons which represent various booking types such as international or domestic flights and distinctions between manually or computer-based bookings. Blanks are distributed to agents by the airline.

The system needs to handle several key requirements including but not limited to:

- Entry of ticket bookings
- Apply discounts
- Apply currency conversions for tickets sold outside of USD
- Apply refunds
- Manage and maintain accountability of blanks to protect against their loss or misuse
- Record commissions of agents

2.1.2 Existing Implementation and Limitations

The existing system was first released in the 90s. This comes with a host of limitations today including:

- There is no GUI making it less intuitive, and more difficult to use.
- It does not contain security measures that limit access based on user need. This is a security concern.
- Once an entry is completed, the exchange rate used on non-USD bookings can't be altered. The entire entry must be deleted and re-entered.
- Commission rates are hard coded which makes adjusting them a complicated and error prone process. As a new release is required to update these rates, compatibility issues between versions are a result.
- The payment and discount systems for existing customers is limited.

The new system requirements will solve for the above issues in addition to requested new features as required. These are detailed on the following pages using use case and design class diagrams.

2.2 Use Case Diagram

The following pages show the proposed design of the new system with Use Case diagrams and specifications.

Visual Paradigm Professional(metaphoresque(City University London))

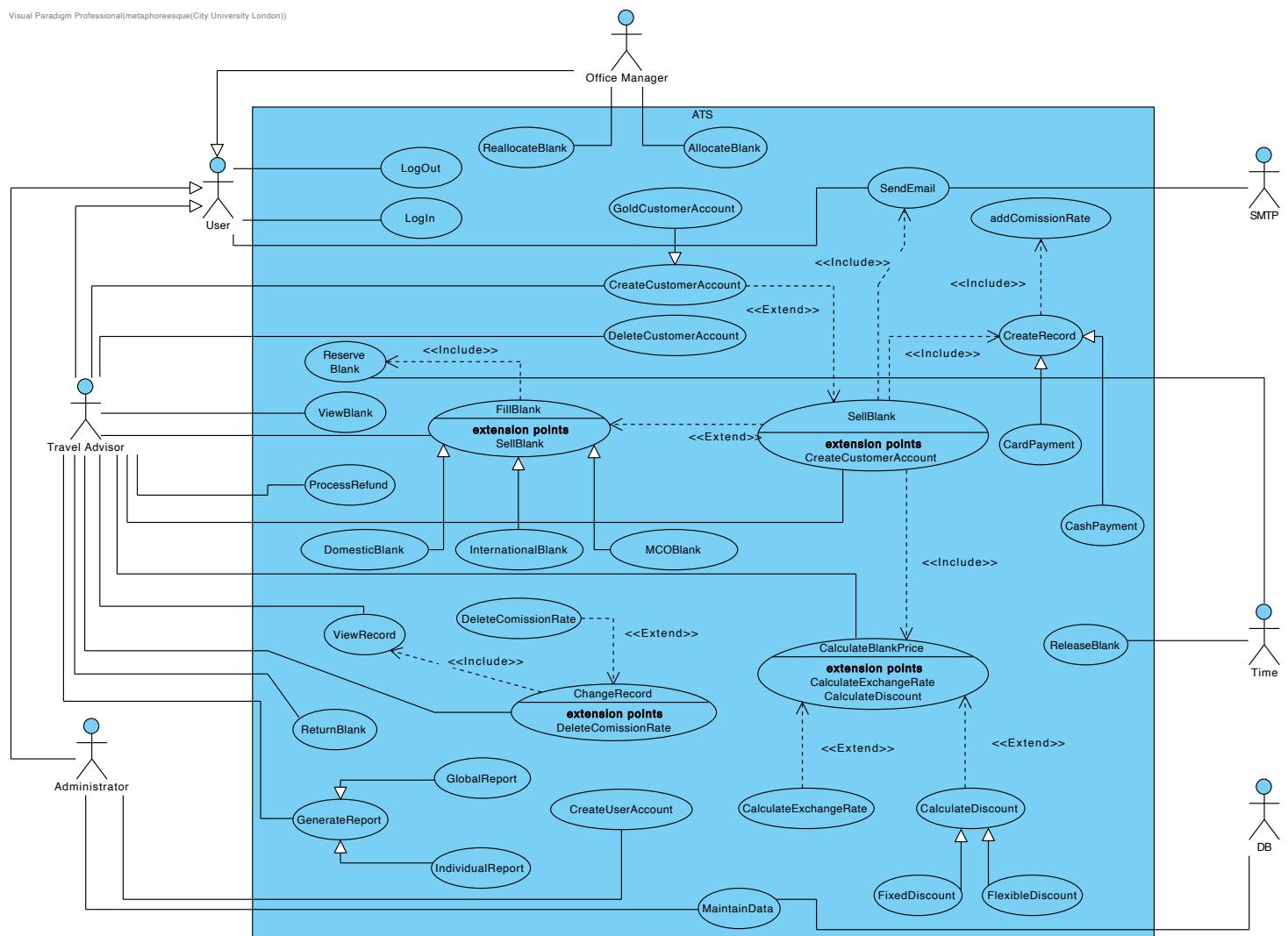


FIGURE 1 USE CASE DIAGRAM

2.3 Use Case Specifications

ID: 01	Use case: FillBlank
Brief description:	
The user inputs the desired origin, destination, date, and number of passengers. The system returns a list of available flights, including flight number, departure/arrival times, and ticket prices. The user selects a flight and proceeds to the booking process.	
Primary actors:	
User	
Secondary actors:	
None	
Preconditions:	
1. User is logged in and has provided the necessary information	
Flow of events:	
<ol style="list-style-type: none"> 1. User inputs search criteria 2. System returns list of available flights 3. User selects a flight 4. System starts the booking process. 	
Postconditions:	
1. A flight has been selected and the booking process has been initiated	
Alternative flow:	
ModifySearch	

TABLE 1 ID:01 USE CASE: FILLBLANK

ID: 01.1	Alternative Flow: FillBlank:ModifySearch
Brief description:	
System informs user that they must modify search	
Primary actors:	
User	
Secondary actors:	
None	
Preconditions:	
1. User has input a filter that is not available	
Flow of events:	
1. Alternative flow begins after step 1 of the main flow 2. System prompts a message to inform the user to modify their search	
Postconditions:	
User has modified their search	

TABLE 2 ID: 01.1 ALTERNATIVE FLOW: FILLBLANK:MODIFYSEARCH

ID: 02	Use case: SellBlank
Brief description:	
The Travel Agent sells a blank to a customer	
Primary actors:	
Travel Advisor, User	
Secondary actors:	
None	
Preconditions:	
The blank has been assigned to the travel advisor	
Flow of events:	
1. The Travel Advisor selects the blank type from the available options 2. The System displays the available blanks 3. The Travel Advisor selects an unassigned blank 4. The system creates a ticket number and gives the blank to the Travel Advisor 5. Travel Advisor fills in the itinerary details on the blank	

- | |
|--|
| 6. Then saved in the ATS System and Database |
|--|

Postconditions:

- | |
|---|
| 1. The blank has been filled successfully |
|---|

Alternative flow:

NoUnassignedBlanks

TABLE 3 ID: 02 USE CASE: SELLBLANK

ID: 02.1	Alternative Flow: SellBlank:NoUnassignedBlanks
Brief description:	
System informs Travel Advisor that there are no unassigned blanks	
Primary actors:	
Travel Advisor	
Secondary actors:	
None	
Preconditions:	
1. There are no assigned blanks for the travel advisor	
Flow of events:	
1. Alternative Flow begins after step 2 in the flow of events 2. System shows there are no available unassigned blanks	
Postconditions:	
1. None	

TABLE 4 ID: 02.1 ALTERNATIVE FLOW: SELLBLANK:NOUNASSIGNEDBLANKS

ID: 03	Use case: CalculateBlankPrice
Brief description:	
The travel advisor calculates the price of a blank based on the details and the currency the user is using	
Primary actors:	
Travel Advisor	
Secondary actors:	
None	
Preconditions:	
Travel Advisor is logged into the ATS system	
Flow of events:	
<ol style="list-style-type: none"> 1. Travel Advisor selects which blank they want the price to be calculated for 2. The system then receives the details from the blank 3. The ticket price is then calculated by the system based on the currency, the discount that is applied and the itinerary details 4. The system then displays the ticket price to the Travel Advisor 	
Postconditions:	
The Price is calculated	
Alternative flow:	
none	

TABLE 5 ID: 03 USE CASE: CALCULATEBLANKPRICE

ID: 03.1	Extension Use case: CalculateBlankPrice
Brief description:	
Segment 1: The system calculates the exchange rate for the ticket	
Segment 2: The system calculates the discount depending on the blank	
Primary actors:	
Travel Advisor	
Secondary actors:	
None	
Segment 1 Preconditions:	
<ol style="list-style-type: none"> 1. Information on the customer must be available 	

Segment 1 Flow of events:
<ol style="list-style-type: none"> 1. The travel advisor checks what country the customer is from 2. Based off the country the customer is from the system can then automatically change the price according to the currency of the customer 3. Ticket price is the present for the travel advisor
Segment 1 Postconditions:
The Price is calculated according to the currency rate
Segment 2 Preconditions:
<ol style="list-style-type: none"> 1. Details of blank and initiaty must be available
Segment 2 Flow of events:
<ol style="list-style-type: none"> 1. The travel advisor uses the information on the blank 2. Travel advisor then inputs the information onto the system 3. The system gives the discounted price
Segment 2 Post Conditions:
Discounted price is present

TABLE 6 ID: 03.1 EXTENSION USE CASE: CALCULATEBLANKPRICE

ID: 04	Use case: ChangeRecord
Brief description:	
Record in the system needs to be changed	
Primary actors:	
Travel Advisor	
Secondary actors:	
None	
Preconditions:	
The Travel Advisor is logged into the ATS System	
Flow of events:	
<ol style="list-style-type: none"> 1. Travel Advisor selects the record 2. System displays the information 3. If Travel Advisor wants to update the record or delete then they would make the changes, could be a global or an individual report 4. The system updates or deletes the record in the ATS 	
Postconditions:	
The record has been updated or deleted	

Alternative flow:

none

TABLE 7 ID:04 USE CASE: CHANGERECORD

ID: 04.1	Extension Use case: ChangeRecord
Brief description:	
Segment 1: Would delete the commission rate on the ticket	
Primary actors:	
Travel Advisor	
Secondary actors:	
None	
Segment 1 Preconditions:	
A Record is available	
Segment 1 Flow of events:	
<ol style="list-style-type: none"> 1. Travel Advisor logs into the system 2. Travel Advisor finds the record they want to change 3. Travel Advisor selects the record to change 4. Travel Advisor deletes the commission rate 	
Segment 1 Postconditions:	
The records commission rate has been deleted	

TABLE 8 ID: 04.1 EXTENSION USE CASE: CHANGERECORD

ID: 05	Use case: SendEmail
Brief description:	
Allows Travel Advisors and User to send emails	
Primary actors	
Travel Advisor, User	
Secondary actors:	
none	

Preconditions:
1. User email address is available in the system
Flow of events:
1. Travel Advisor selects the User to send an email to 2. Travel Advisor writes the email and puts in crucial files 3. The user receives the email
Postconditions:
1. Email is sent.
Alternative flow:
SMTPError

TABLE 9 ID: 05 USE CASE: SENDEMAIL

ID: 05.1	Alternative Flow: SendEmail:SMTPError
Brief description:	
Email does not send due to email providing service having issues	
Primary actors	
Travel Advisor, User	
Secondary actors:	
none	
Preconditions:	
1. There must be a problem with the SMTP Emailing Service	
Flow of events:	
1. Alternative flow begins after step 3 of the flow of events 2. System prompts an error message to say that there was a problem sending the email	
Postconditions:	
1. Error Message	

TABLE 10 ID: 05.1 ALTERNATIVE FLOW: SENDEMAIL:SMTPERROR

ID: 06	Use case: ReallocateBlank
Brief description:	
The Office Manager assigns the blanks to travel advisors	
Primary actors	
Office Manager	
Secondary actors:	
Travel Advisor	
Preconditions:	
<ol style="list-style-type: none"> 1. Office Manager has the permission to perform reallocation 2. Office Manager has logged into the system 	
Flow of events:	
<ol style="list-style-type: none"> 1. Office Manager checks the system for blanks 2. Office Manager selects the blank that needs to be reallocated 3. Office Manager selects a travel advisor to assign the blank for 4. The system updates the database to reassign the blank to the new travel advisor 5. The system notifies the travel advisor of the reassigned blank 	
Postconditions:	
<ol style="list-style-type: none"> 1. The travel advisor can sell the reassigned blank 	
Alternative flow:	
NoAvailableBlanks	

TABLE 11 ID: 06 USE CASE: REALLOCATEBLANK

ID: 06.1	Alternative Flow: ReallocateBlank:NoAvailableBlanks
Brief description:	
There are no Available blanks for the office manager to reallocate	
Primary actors	
Office Manager	
Secondary actors:	
Travel Advisor	
Preconditions:	
<ol style="list-style-type: none"> 1. No available blanks 	

Flow of events:

1. Alternative Flow starts from step one of the flow of events
2. Office Manager checks the system and there are no available blanks for them to reallocate to travel advisor

Postconditions:

1. No blanks available

TABLE 12 ID: 06.1 ALTERNATIVE FLOW: REALLOCATEBLANK:NOAVAILABLEBLANKS

ID: 07	Use case: AllocateBlank
Brief description:	
Office Manager assigns new blanks to travel advisor	
Primary actors	
Office Manager	
Secondary actors:	
Travel Advisor	
Preconditions:	
There are new blanks available for allocation	
Flow of events:	
<ol style="list-style-type: none"> 1. Office Manager checks system for blanks 2. Office Manager selects the blanks types to allocate 3. Office Manager selects the travel advisor to assign the blanks 4. The system updates the database to allocate the selected blanks to the travel advisor 5. The system notified the travel advisor of the assigned blanks 	
Postconditions:	
The Travel Advisor can sell the assigned blank	
Alternative flow:	
NoAvailableBlanks	

TABLE 13 ID: 07 USE CASE: ALLOCATEBLANK

ID: 07.1	Alternative Flow: AllocateBlank:NoAvailableBlanks
Brief description:	
There are no available blanks for the office manager to allocate	
Primary actors	
Office Manager	
Secondary actors:	
Travel Advisor	
Preconditions:	
1. No blanks available	
Flow of events:	
1. Alternative Flow begins from step 1 2. Office Manager checks the system and there is no available blanks	
Postconditions:	
No blanks available	

TABLE 14 ID: 07.1 ALTERNATIVE FLOW: ALLOCATEBLANK:NOAVAILABLEBLANKS

ID: 08	Use case: GenerateReport
Brief description:	
The system generates a report, including information such as total sales, number of bookings, and average ticket price.	
Primary actors	
Travel Advisor	
Secondary actors:	
None	
Preconditions:	
The necessary data is available.	
Flow of events:	
1. User requests report 2. System gets data from database 3. System puts all of the data in to a report 4. System generates the report	

Postconditions: The sales report has been generated.
Alternative flow: none

TABLE 15 ID: 08 USE CASE: GENERATEREPORT

ID: 09	Use case: MaintainData
Brief description: Allows the administrator to maintain the stock of blanks and fix problems with the database	
Primary actors Administrator	
Secondary actors: None	
Preconditions: 1. Administrator is logged in to ATS	
Flow of events: 1. Administrator selects the maintenance option 2. Displays the menu 3. Selects the required option 4. System performs the action 5. System displays the results to the Administrator	
Postconditions: The action is performed and the result is displayed	
Alternative flow: None	

TABLE 16 ID: 09 USE CASE: MAINTAINDATA

ID: 10	Use case: CreateCustomerAccount
Brief description:	
The Travel Advisor creates a new customer account for recording customer information	
Primary actors	
Travel Advisor	
Secondary actors:	
None	
Preconditions:	
Travel Advisor has logged into the system	
Flow of events:	
<ol style="list-style-type: none"> 1. The Travel Advisor selects the option to create a new customer account 2. Travel Advisor enters the customers personal information into the system 3. The system validates the information and gives a unique customerID 4. The system adds the information and ID to the database 5. System shows the customerID to the travel advisor 	
Postconditions:	
The Travel Advisor can now associate ticket sales with the new customer	
Alternative flow:	
UnavailableEmail	

TABLE 17 ID: 10 USE CASE: CREATECUSTOMERACCOUNT

ID: 10.1	Alternative Flow: CreateCustomerAccount:UnavailableEmail
Brief description:	
If the email does not exist then the Travel Advisor cannot make an account for the customer	
Primary actors	
Travel Advisor	
Secondary actors:	
None	
Preconditions:	
There is an unavailable email	

Flow of events:

1. Alternative flow begins from step 2
2. Travel Advisor enters the customers personal information and it is invalidated
3. System prompts an error message stating that email is not available

Postconditions:

Error Message

TABLE 18 ID:10.1 ALTERNATIVE FLOW: CREATECUSTOMERACCOUNT:UNAVAILABLEEMAIL

2.4 Use Case Indexed List

ID	USE CASE WITH PRIORITY	DESCRIPTION	IMPACT OF PROJECTED RISKS
			TIME/BUDGET
1	CreateUserAccount	Administrator must have access to create user's account	If there are security risks, it can impact the overall system security and user's privacy
2	CreateCustomerAccount	Only travel advisor can create customer account	Legal and regulatory risks include issues such as non-compliance with data protection regulations or changes in laws that impact the project. This may lead to legal disputes, fines or damage to the travel agency's reputation
3	GoldCustomerAccount	Have a privilege to a function in the system of a 'late payment' and discount as a percentage of the ticket fare which is maintained by Office Manager	Low time/budget impact
4	DeleteCustomerAccount	Case when customer is no longer using services	Low time/budget impact
5	LogIn	Everyone who have an account in the system can login, but depends on the type of account, role and access criteria	If the system is not compatible with the operating system of the devices used by the users, it may result in compatibility issues and technical problems. This will result in delays and increased development costs
6	LogOut	Every user of the system can logout	Low time/budget impact
7	SendEmail	Travel advisor, administrator, office manager has access to send an email	Emails can be overwhelming to the customers. Hence reduce the engagement from the customers not attracting them to be interested in the

			flight deals. Analytics should be used to track the performance of their campaigns and make data driven decisions
8	AllocateBlank	Office Manager allocates blanks to each travel advisor. Only the advisor who has been assigned a blank can sell it to the customer	Low time/budget impact
9	ReallocateBlank	Case when blank is being assigned to a different advisor or when unused blanks are returned to the airline.	Low time/budget impact
10	ViewBlank	At any time, blank can be viewed by Office Manager or travel advisor	Low time/budget impact
11	FillBlank	Travel advisor is responsible for the information put into the blank	If the travel advisor makes a mistake, the blank can be void, but also cause further delays
12	DomesticBlank	Filled in using computer	In case of a system crash, this can cause time and budget problems, while fixing the system to be back up and running
13	InternationalBlank	The tickets consist of up to 4 ticket coupons and auditor's coupons	Low time/budget impact
14	MCOBlank	Miscellaneous Charges Order blank are used for excess luggage or other miscellaneous services	Low time/budget impact
15	ReserveBlank	A blank becomes a valid ticket if the itinerary of the journey is placed on the flight	If there are operational risks, such as a failure in the booking system, it can lead to a loss of revenue

		coupons	
16	CalculateBlankPrice	Calculation of the blank is based on the type of destination, domestic or interlines including agent's commission	Risks also depend on the global currency exchange, resulting in bigger spendings and less customers desiring expensive tickets
17	CalculateExchangeRate	The rate used is the one specified by the National Bank on the day of the sale.	Sudden National Bank's change can result in longer consideration whether to sell the blank for that price, also the total amount of the blank might not be something the company is happy to offer to the customer
18	CalculateDiscount	The discount plan is set by the Office Manager	The duration of the discount must be long enough for customers to act upon the urgency. Otherwise, this would not be able to attract the customers , hence less revenue is being produced
19	FixedDiscount	Same percentage of discount given to customer for each ticket fare irrespective of type and number of tickets sold	Low time/budget impact
20	FlexibleDiscount	The percentage of discount depends on the value of the sales to the same customer accumulated within a calendar month	Low time/budget impact
21	SellBlank	A blank can only be sold once it is assigned to a Travel Agent	The period of assigning blanks to the travel agents can take a long of time, hence will cause a delay or no sales as other agencies might get to the customers first.
22	CreateRecord	When a sale has been made, a record of the sales is created	Low time/budget impact
23	CardPayment	The precision for the rate is 4 digits after the decimal	Card payment involves a third -party software in order to accept the

		point	payment. If the system fails, it will create a delay in the transaction and increase the cost of fixing the system.
24	CashPayment	The precision for the rate is 4 digits after the decimal point	Accepting cash payment means that the person in charge of the register would need to know how to spot counterfeit money. There is a risk of counterfeit money being overlooked, hence loss of revenue
25	ReleaseBlank	After the sale has been made, the travel advisor can release the blanks to the customers	Low time/budget impact
26	addComissionRate	Percentage of commission rate depends on the type of ticket and is agreed in a contract between the Travel Agent and the airline	Low time/budget impact
27	ReturnBlank	The date and type of blanks returned to the airline are recorded in the Travel Agent's log file	Low time/budget impact
28	ProcessRefund	Customers can cancel the ticket up to one year after the sale	However, if the blanks are damaged by the customers, then the blanks cannot be returned to office manager and be reassigned. Depending on the agreement between the travel agent and the airline , the commission given may be retracted .
29	ViewRecord	The record can be viewed by the travel advisor, administrator and the office manager	Low time/budget impact
30	ChangeRecord	When a customer cancels a ticket a record about this is made detailing the ticket and the amount refunded	Low time/budget impact

	DeleteComissionRate	In case of a refund, commission rate that were used up to one year ago can be used.	Low time/budget impact
31	GenerateReport	Sales report must list all sales transactions together with the corresponding sub-totals and grand totals	The accuracy and completeness of data are critical. Data quality risks can arise if there are errors in the data or it is inconsistent. This may result in incomplete reports, affecting the travel agency's decision-making process.
32	GlobalReport	Each report contains one line per customer	Low time/budget impact
33	IndividualReport	Individual reports are prepared by travel advisor	Low time/budget impact
34			

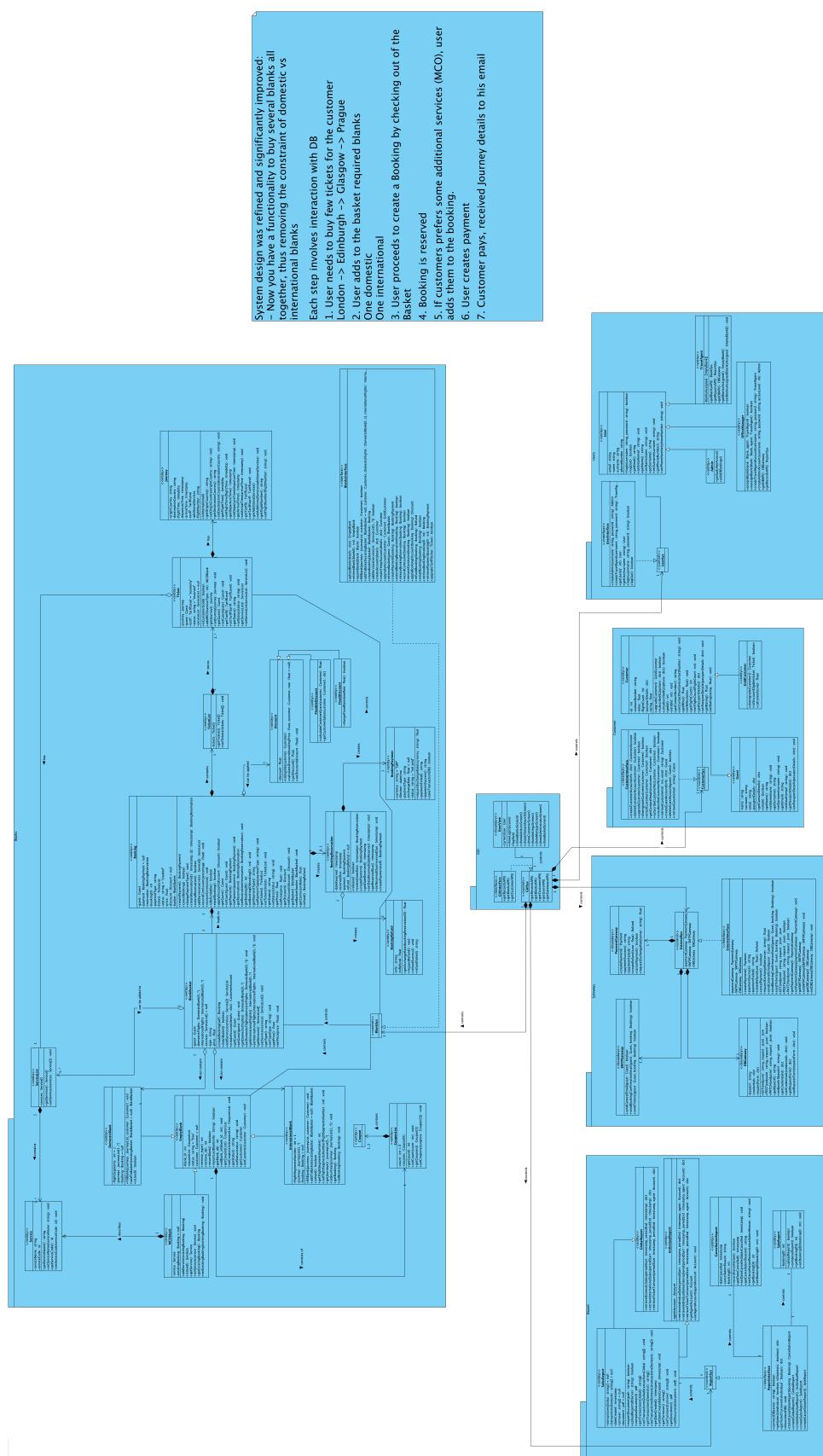
TABLE 19 USE CASE INDEXED LIST

3 System Design

3.1 Design Class Diagram

Please see the design class diagram on the following page. Due to page size limitations, it is necessary to rotate the view and use zoom tools. The graphic type will maintain resolution as zoom is used.

FIGURE 2 DESIGN CLASS DIAGRAM



3.2 ER Diagram

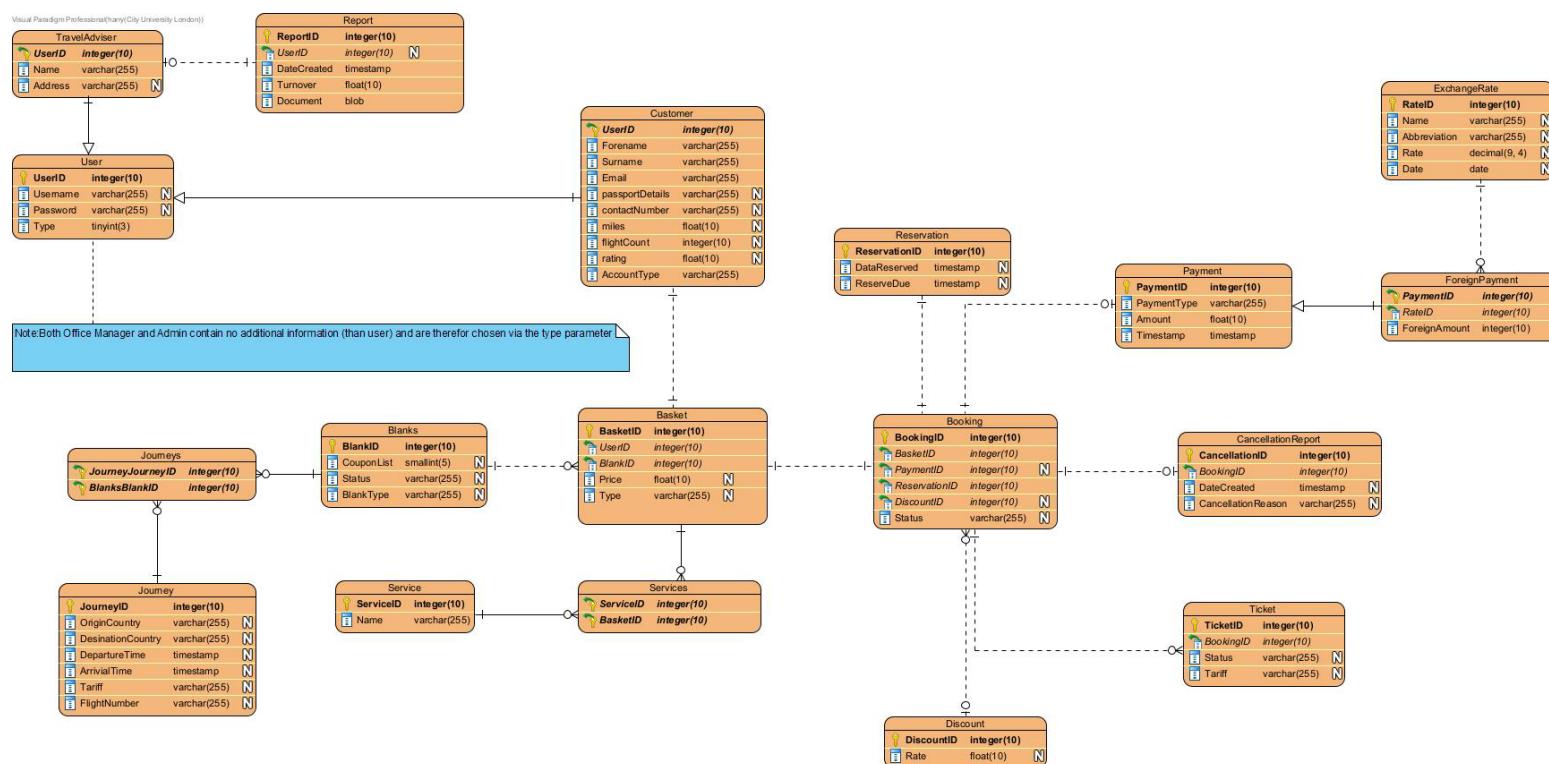


FIGURE 3 ER DIAGRAM EXAMPLE

3.3 DDL Statements

```
-- Create all tables
```

```
CREATE TABLE Customer (
    UserID int(10) NOT NULL,
    Forename varchar(255) NOT NULL,
    Surname varchar(255) NOT NULL,
    Email varchar(255) NOT NULL,
    passportDetails varchar(255),
    contactNumber varchar(255),
    miles float,
    flightCount int(10),
    rating float,
    AccountType varchar(255) NOT NULL,
    PRIMARY KEY (UserID)
);
```

```
CREATE TABLE `User` (
    UserID int(10) NOT NULL AUTO_INCREMENT,
    Username varchar(255),
    Password varchar(255),
    Type tinyint(3) NOT NULL,
    PRIMARY KEY (UserID)
);
```

```
CREATE TABLE Reservation (
    ReservationID int(10) NOT NULL AUTO_INCREMENT,
```

```
    DataReserved timestamp NULL,  
    ReserveDue timestamp NULL,  
    PRIMARY KEY (ReservationID)  
);
```

```
CREATE TABLE Payment (  
    PaymentID int(10) NOT NULL AUTO_INCREMENT,  
    PaymentType varchar(255),  
    Amount float,  
    Timestamp timestamp NULL,  
    PRIMARY KEY (PaymentID)  
);
```

```
CREATE TABLE Service (  
    ServiceID int(10) NOT NULL AUTO_INCREMENT,  
    Name varchar(255) NOT NULL,  
    PRIMARY KEY (ServiceID)  
);
```

```
CREATE TABLE Discount (  
    DiscountID int(10) NOT NULL AUTO_INCREMENT,  
    Rate float,  
    PRIMARY KEY (DiscountID)  
);
```

```
CREATE TABLE Journey (  
    JourneyID int(10) NOT NULL AUTO_INCREMENT,  
    OriginCountry varchar(255),
```

```
DesinationCountry varchar(255),  
DepartureTime timestamp NULL,  
ArrivialTime timestamp NULL,  
Tariff varchar(255),  
FlightNumber varchar(255),  
PRIMARY KEY (JourneyID)  
);
```

```
CREATE TABLE Ticket (  
TicketID int(10) NOT NULL AUTO_INCREMENT,  
BookingID int(10) NOT NULL,  
Status varchar(255),  
Tariff varchar(255),  
CONSTRAINT TicketID PRIMARY KEY (TicketID)  
);
```

```
CREATE TABLE Booking (  
BookingID int(10) NOT NULL AUTO_INCREMENT,  
BasketID int(10) NOT NULL,  
PaymentID int(10),  
ReservationID int(10) NOT NULL,  
DiscountID int(10),  
Status varchar(255),  
PRIMARY KEY (BookingID)  
);
```

```
CREATE TABLE Basket (  
BasketID int(10) NOT NULL AUTO_INCREMENT,
```

```
UserID int(10) NOT NULL,  
BlankID int(10) NOT NULL,  
Price float,  
Type varchar(255),  
PRIMARY KEY (BasketID)  
);
```

```
CREATE TABLE Blanks (  
    BlankID int(10) NOT NULL AUTO_INCREMENT,  
    CouponList smallint(5),  
    Status varchar(255),  
    BlankType varchar(255),  
    PRIMARY KEY (BlankID)  
);
```

```
CREATE TABLE Report (  
    ReportID int(10) NOT NULL AUTO_INCREMENT,  
    UserID int(10),  
    DateCreated timestamp NOT NULL,  
    Turnover float NOT NULL,  
    Document blob NOT NULL,  
    PRIMARY KEY (ReportID)  
);
```

```
CREATE TABLE TravelAdviser (  
    UserID int(10) NOT NULL,  
    Name varchar(255) NOT NULL,  
    Address varchar(255),
```

PRIMARY KEY (UserID)

);

CREATE TABLE CancellationReport (

CancellationID int(10) NOT NULL AUTO_INCREMENT,

BookingID int(10) NOT NULL,

DateCreated timestamp NULL,

CancellationReason varchar(255),

PRIMARY KEY (CancellationID)

);

CREATE TABLE Services (

ServiceID int(10) NOT NULL,

BasketID int(10) NOT NULL,

PRIMARY KEY (ServiceID, BasketID)

);

CREATE TABLE Journeys (

JourneyJourneyID int(10) NOT NULL,

BlanksBlankID int(10) NOT NULL,

PRIMARY KEY (JourneyJourneyID, BlanksBlankID)

);

ALTER TABLE

Customer

ADD

CONSTRAINT FKCustomer106541 FOREIGN KEY (UserID) REFERENCES `User` (UserID) ON
DELETE CASCADE;

ALTER TABLE

Ticket

ADD

CONSTRAINT FKTicket16114 FOREIGN KEY (BookingID) REFERENCES Booking (BookingID);

ALTER TABLE

Booking

ADD

CONSTRAINT FKBooking378939 FOREIGN KEY (BasketID) REFERENCES Basket (BasketID);

ALTER TABLE

Basket

ADD

CONSTRAINT FKBasket871522 FOREIGN KEY (UserID) REFERENCES Customer (UserID);

ALTER TABLE

Basket

ADD

CONSTRAINT FKBasket809670 FOREIGN KEY (BlankID) REFERENCES Blanks (BlankID);

ALTER TABLE

Booking

ADD

CONSTRAINT FKBooking243396 FOREIGN KEY (PaymentID) REFERENCES Payment (PaymentID);

ALTER TABLE

Booking

ADD

CONSTRAINT FKBooking14350 FOREIGN KEY (ReservationID) REFERENCES Reservation (ReservationID);

ALTER TABLE

Booking

ADD

CONSTRAINT FKBooking71204 FOREIGN KEY (DiscountID) REFERENCES Discount (DiscountID);

ALTER TABLE

Report

ADD

CONSTRAINT FKReport905231 FOREIGN KEY (UserID) REFERENCES TravelAdviser (UserID);

ALTER TABLE

TravelAdviser

ADD

CONSTRAINT FKTravelAdvi396996 FOREIGN KEY (UserID) REFERENCES `User` (UserID);

ALTER TABLE

CancellationReport

ADD

CONSTRAINT FKCancellationReport194521 FOREIGN KEY (BookingID) REFERENCES Booking (BookingID);

ALTER TABLE

Services

ADD

CONSTRAINT FKServices290285 FOREIGN KEY (ServiceID) REFERENCES Service (ServiceID);

```
ALTER TABLE Services
ADD CONSTRAINT FKServices892642 FOREIGN KEY (BasketID) REFERENCES Basket (BasketID);

ALTER TABLE Journeys
ADD CONSTRAINT FKJourneys861196 FOREIGN KEY (JourneyJourneyID) REFERENCES Journey (JourneyID);

ALTER TABLE Journeys
ADD CONSTRAINT FKJourneys913122 FOREIGN KEY (BlanksBlankID) REFERENCES Blanks (BlankID);

DML statements (SQL)

-- Drop all tables

DROP TABLE IF EXISTS `Services`;

DROP TABLE IF EXISTS `Basket`;

DROP TABLE IF EXISTS `Journeys`;

DROP TABLE IF EXISTS `Blanks`;
```

```
DROP TABLE IF EXISTS `Service`;  
  
DROP TABLE IF EXISTS `Journey`;  
  
DROP TABLE IF EXISTS `Booking`;  
  
DROP TABLE IF EXISTS `Reservation`;  
  
DROP TABLE IF EXISTS `CancellationReport`;  
  
DROP TABLE IF EXISTS `Payment`;  
  
DROP TABLE IF EXISTS `Ticket`;  
  
DROP TABLE IF EXISTS `Discount`;  
  
DROP TABLE IF EXISTS `Customer`;  
  
DROP TABLE IF EXISTS `User`;  
  
DROP TABLE IF EXISTS `TravelAdviser`;  
  
DROP TABLE IF EXISTS `Report`;
```

3.4 DML Statements (SQL)

```
-- Select  
-- Selects the customer email with name "John Smith"  
SELECT  
    Email
```

```

FROM
    Customer
WHERE
    Forename = "John"
    AND Surname = "Smith";

-- Selects the names and reason for cancellations of all customers
SELECT
    Customer.Forename,
    Customer.Surname,
    CancellationReport.CancellationReason
FROM
    Customer
    INNER JOIN Basket ON Customer.UserID = Basket.UserID
    INNER JOIN Booking ON Basket.BasketID = Booking.BasketID
    INNER JOIN CancellationReport ON Booking.BookingID =
CancellationReport.BookingID;

-- Insert
-- Inserts some discount rates
INSERT INTO
    Discount (Rate)
VALUES
    (0.8),
    (0.9),
    (0.75);

-- Inserts some payments
INSERT INTO
    Payment (PaymentType, Amount, Timestamp)
VALUES
    ("Card", 120.0, NOW()),
    ("Cash", 40, NOW()),
    ("Card", 189, NOW());

-- Update
-- Updates the customer with UserID 1 to include a contactNumber
UPDATE
    Customer
SET
    contactNumber = "07624 560472"
WHERE
    UserID = 1;

-- Updates the password (sudo reset) for the TravelAdviser "QuickTrips"
UPDATE
    User
    INNER JOIN TravelAdviser ON User.UserID = TravelAdviser.UserID
SET
    User.Password = "1234"
WHERE
    TravelAdviser.Name = "QuickTrips";

-- Delete
-- Deletes user with UserID 1 and will also delete customer with UserID
1 (if exists)

```

```

DELETE FROM
    User
WHERE
    UserID = 1;

-- Deletes both the User and TravelAdviser with UserID 2 as well as any
assosiated reports
DELETE Report,
TravelAdviser,
User
FROM
    User
INNER JOIN TravelAdviser ON User.UserID = TravelAdviser.UserID
INNER JOIN Report ON TravelAdviser.UserID = Report.UserID
WHERE
    User.UserID = 2;

```

3.5 GUI Design

GUI

Home Page

- This is the homepage without logging in
- When the user opens the ap, they can buy blanks, or they can login
- If the user buys blanks without logging in, they can go to the basket and purchase blanks via card or cash payments
- On every page other than the home page, there will be a back button to go back to the previous page

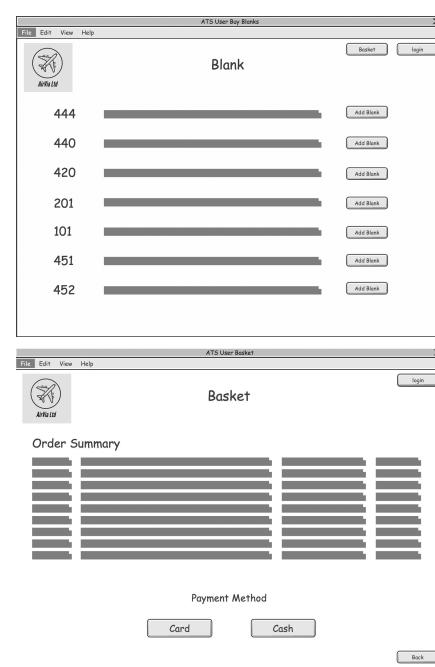


FIGURE 4 GUI: HOME PAGE

Sign Up

- A new user can create an account, they have to create a username and password and enter their address

The screenshot shows a window titled "ATS Sign Up". At the top left is a logo for "AirVia Ltd" featuring a stylized airplane. The main title "Sign Up" is centered above four input fields: "Account Type" (with a dropdown arrow), "Username" (text box), "Password" (text box), and "Address" (text box). Below these fields is a "Create Account" button. In the bottom right corner of the window is a "Back" button.

FIGURE 5 GUI: SIGN UP

MANAGER

Manager Menu

- The manager has access to all reports and can set/change discounts
- From the menu, they can view reports or generate reports and choose dates with the dropdown option
- They can also choose a customer, choose the discount type and amount and set the discount
- For an existing discount, they can also choose the customer and change it
- They can allocate blanks to Travel Agents

The screenshot shows a window titled "ATS manager menu". At the top left is a logo for "AirVia Ltd". The main title "Manager Menu" is centered above several sections: "View Report" (with a dropdown for "Month" set to "feb 2022" and buttons for "Interline Sales report", "Domestic Sales Report", "Global Sales Report", "Individual Reports", and "Stock Report"), "Generate Report" (with dropdowns for "Report Type" (set to "Interline sales"), "From", "To", and a "generate report" button), "Discount" (with dropdowns for "Customer" and "Type", a text box for "Add:", and buttons for "set Discount" and "Change Discount"), and "Allocate Blanks" (with dropdowns for "Travel Agent" and "Type of Blank", a text box for "Quantity", and a "Add Blank" button). In the bottom right corner of the window is a "Back" button.

FIGURE 6 GUI: MANAGER MENU

Manager Report

- Based on the options the manager chose, e.g dates or type of report, the manager will be shown the requested report

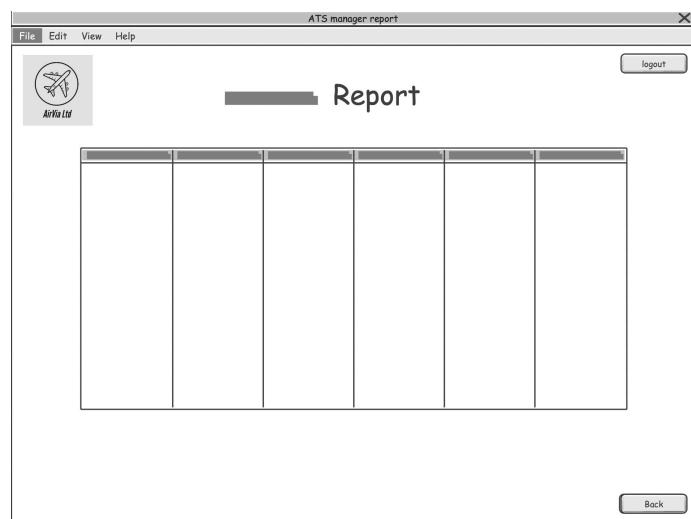


FIGURE 7 GUI: MANAGER REPORT

TRAVEL AGENT

Travel Agent Menu

- A travel agent can look at their sales report, they can generate reports, they can look at their stock, their commission rate and they can also update their account details

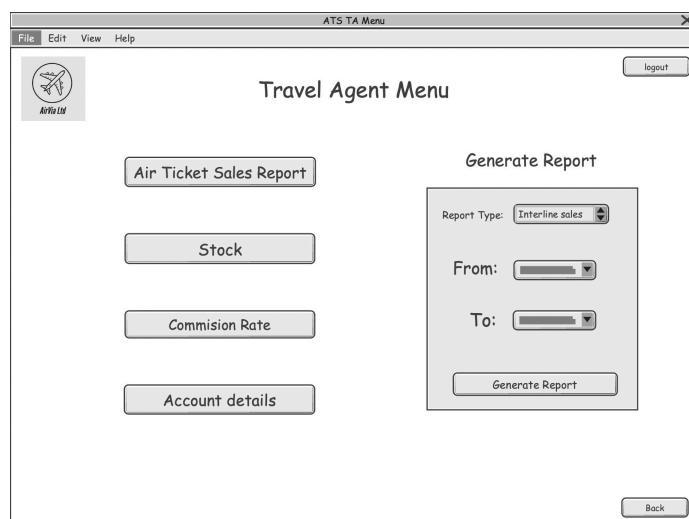


FIGURE 8 GUI: TRAVEL AGENT MENU

Blanks Stock

- Travel agents can see how many blanks they have, and also what types of blanks

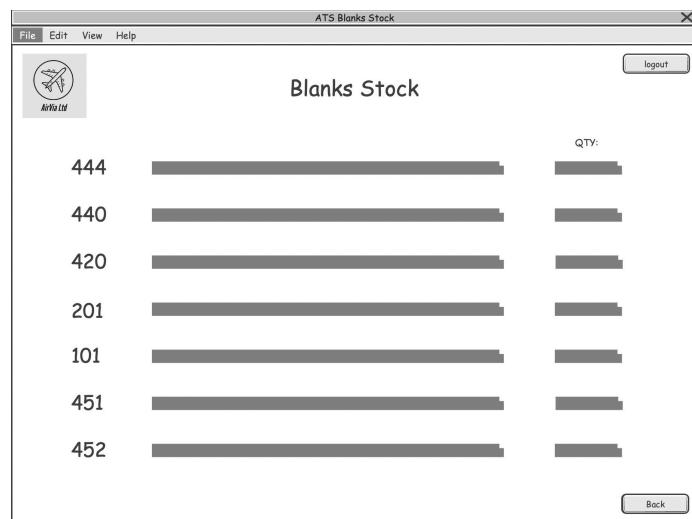


FIGURE 9 GUI: BLANKS STOCK

TA Report

- TA's can generate or view reports, they could also change the rate on the chosen report if there is any mistake

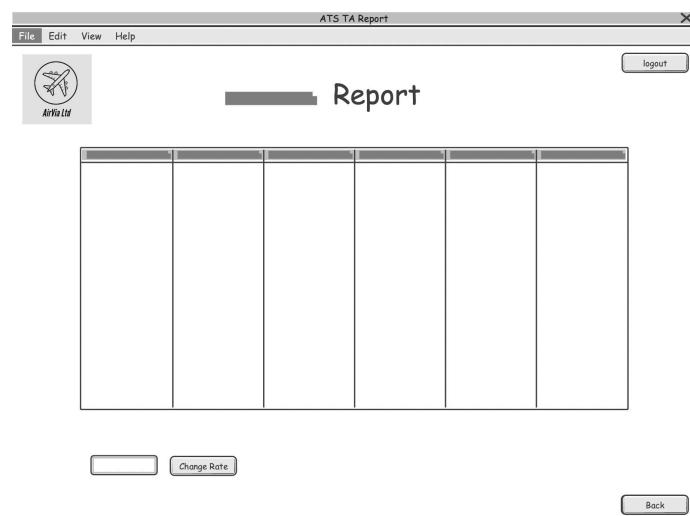


FIGURE 10 GUI: TA REPORT

TA Account Details

- Travel agents can update their account details, their username, passwords or address

The screenshot shows a window titled "ATS TA Details". At the top left is a logo for "AirVita Ltd" featuring a stylized airplane. The main title "TA Account Details" is centered above three input fields: "Username:", "Password:", and "Address:". Below these fields is a "Update" button. In the top right corner, there is a "logout" button, and at the bottom right, a "Back" button.

FIGURE 11 GUI: TA ACCOUNT DETAILS

SYSTEM ADMIN

System Admin Menu

- System admin can view the database, they can maintain the list of travel agents and they can also maintain stock

The screenshot shows a window titled "ATS System Admin Menu". At the top left is a logo for "AirVita Ltd" featuring a stylized airplane. The main title "System Admin Menu" is centered above three buttons: "Database", "view TA's", and "View Stock". In the top right corner, there is a "logout" button, and at the bottom right, a "Back" button.

FIGURE 12 GUI: SYSTEM ADMIN MENU

List of Travel Agents

- System admins can view the travel agents details and maintain this data on the database and they can also view the ticket types used



FIGURE 13 GUI: LIST OF TRAVEL AGENTS

System Admin Stock

- System admin can look at all of the stock on the system

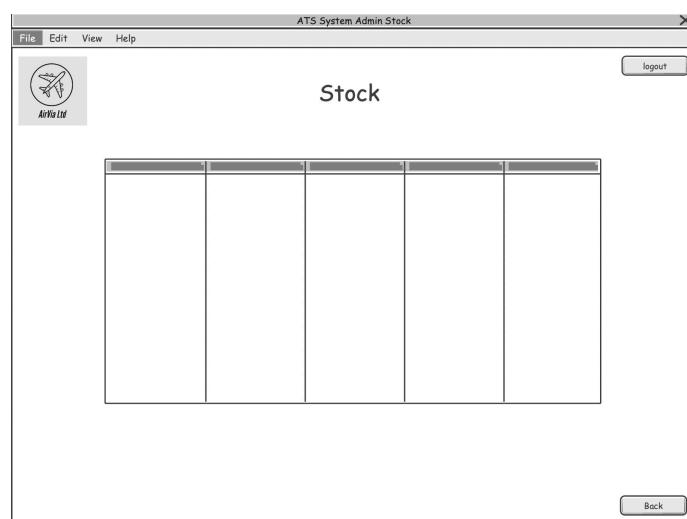


FIGURE 14 GUI: SYSTEM ADMIN STOCK

Database

- System admin can maintain the database, they can also back it up or they can restore the database

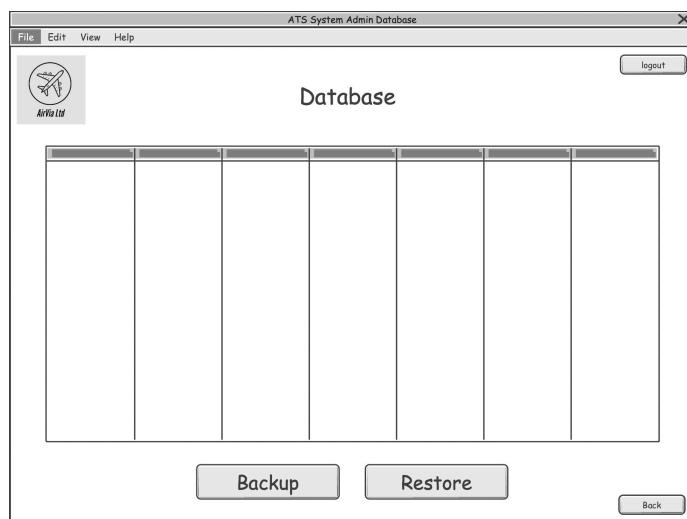


FIGURE 15 GUI: DATABASE

CUSTOMER WITH ACCOUNT

Customer Menu

- From the menu, a customer can buy blanks, they can view their orders if they want a refund or they can update their details
- They could also go to the basket to checkout

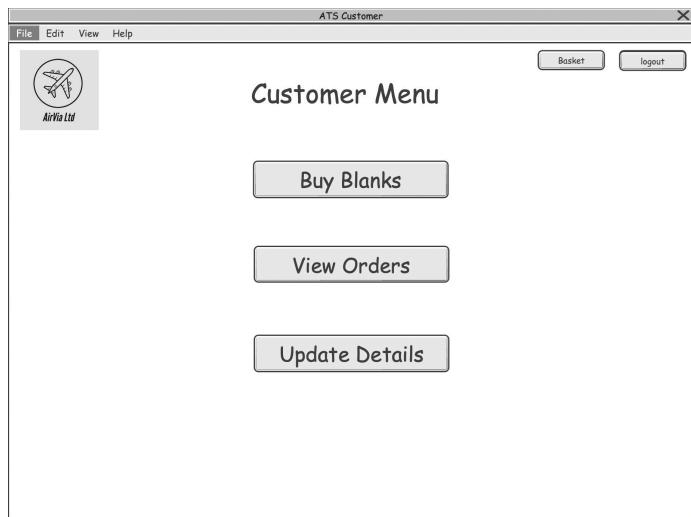


FIGURE 16 GUI: CUSTOMER MENU

Customer Account Details

- Customers with accounts can change/update their details

The screenshot shows a software window titled "ATS Customer: Account Details". At the top, there's a menu bar with "File", "Edit", "View", and "Help". The main area is titled "Account Details" and contains three text input fields: "Username", "Password", and "Address". Below these fields is a "Update" button. In the bottom right corner, there is a "Back" button.

FIGURE 17 GUI: CUSTOMER ACCOUNT DETAILS

Buying Blanks

- Customers can view the different kinds of blanks and they can add them to the basket to purchase
- After they have added blanks, they can go to the basket from the basket button

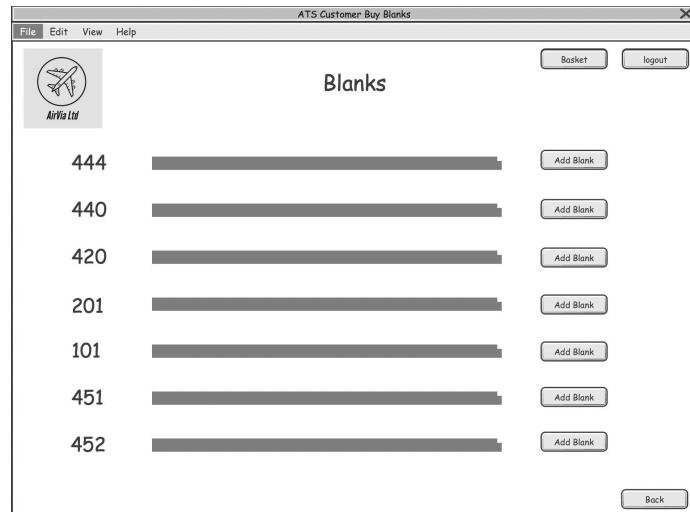


FIGURE 18 GUI: BUYING BLANKS

Customer Basket

- In this page, customers can check what is in their baskets
- The order summary will contain info about: blank types, blank quantity, currency rate, price, discount(if given),and order total
- As they are customers with accounts, they can pay with cash, card, or the pay later option

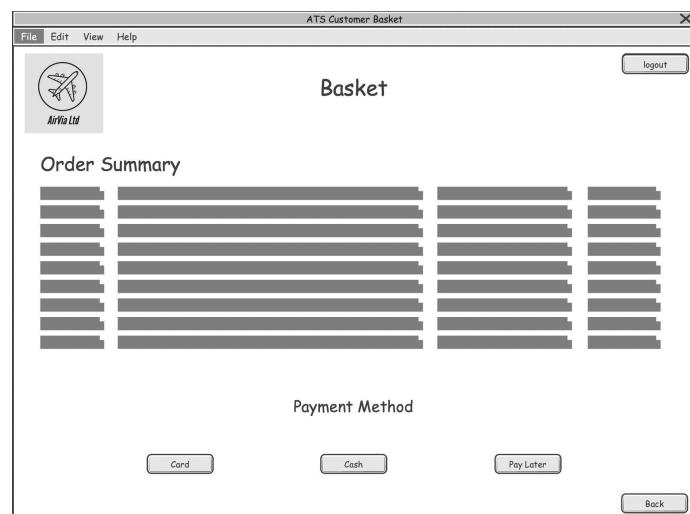


FIGURE 19 GUI: CUSTOMER BASKET

Customer Orders

- Customer can view past orders
- Customers can choose an order and cancel it and they will get a refund



FIGURE 20 GUI: CUSTOMER ORDERS

MAP OF GUI (annotated)

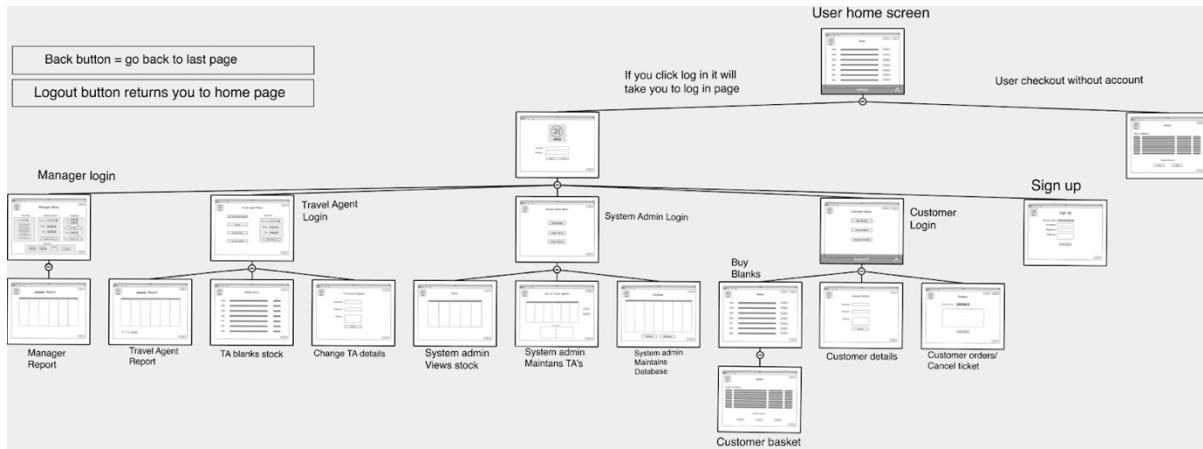


FIGURE 21 MAP OF GUI (ANNOTATED) EXAMPLE

GUI NAVIGATION

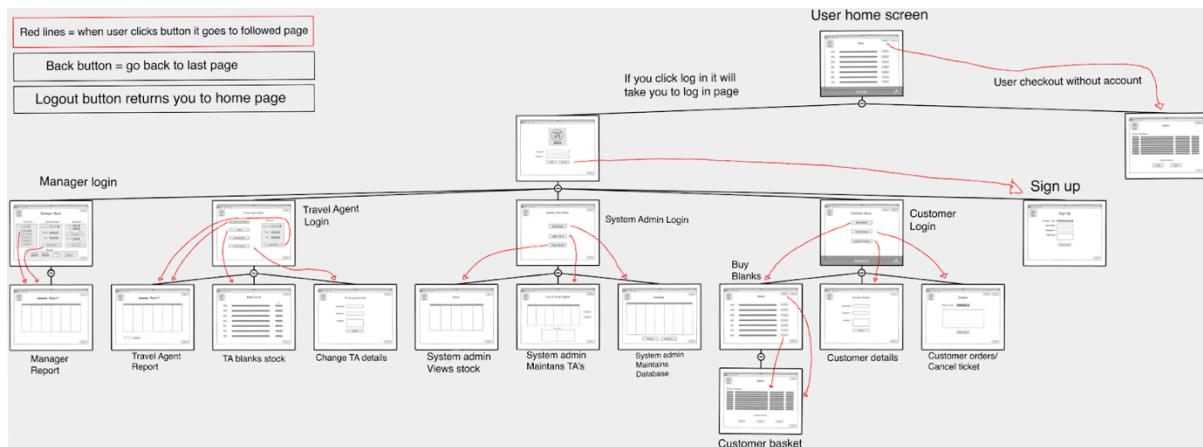


FIGURE 22 GUI NAVIGATION EXAMPLE

4 Appendix

4.1 Table of Figures

FIGURE 1 USE CASE DIAGRAM.....	6
FIGURE 2 DESIGN CLASS DIAGRAM	25
FIGURE 3 ER DIAGRAM EXAMPLE.....	27
FIGURE 4 GUI: HOME PAGE	38
FIGURE 5 GUI: SIGN UP	39
FIGURE 6 GUI: MANAGER MENU.....	39
FIGURE 7 GUI: MANAGER REPORT	40
FIGURE 8 GUI: TRAVEL AGENT MENU	40
FIGURE 9 GUI: BLANKS STOCK	41
FIGURE 10 GUI: TA REPORT	41
FIGURE 11 GUI: TA ACCOUNT DETAILS	42
FIGURE 12 GUI: SYSTEM ADMIN MENU.....	42
FIGURE 13 GUI: LIST OF TRAVEL AGENTS	43
FIGURE 14 GUI: SYSTEM ADMIN STOCK	43
FIGURE 15 GUI: DATABASE	44
FIGURE 16 GUI: CUSTOMER MENU.....	45
FIGURE 17 GUI: CUSTOMER ACCOUNT DETAILS	45
FIGURE 18 GUI: BUYING BLANKS	46
FIGURE 19 GUI: CUSTOMER BASKET.....	46
FIGURE 20 GUI: CUSTOMER ORDERS	47
FIGURE 21 MAP OF GUI (ANNOTATED) EXAMPLE	48
FIGURE 22 GUI NAVIGATION EXAMPLE.....	48

4.2 Table of Tables

TABLE 1 ID:01 USE CASE: FILLBLANK.....	7
TABLE 2 ID: 01.1 ALTERNATIVE FLOW: FILLBLANK:MODIFYSEARCH.....	8
TABLE 3 ID: 02 USE CASE: SELLBLANK.....	9
TABLE 4 ID: 02.1 ALTERNATIVE FLOW: SELLBLANK:NOUNASSIGNEDBLANKS	9
TABLE 5 ID: 03 USE CASE: CALCULATEBLANKPRICE	10
TABLE 6 ID: 03.1 EXTENSION USE CASE: CALCULATEBLANKPRICE.....	11
TABLE 7 ID:04 USE CASE: CHANGERECORD	12
TABLE 8 ID: 04.1 EXTENSION USE CASE: CHANGERECORD	12
TABLE 9 ID: 05 USE CASE: SENDEMAIL.....	13
TABLE 10 ID: 05.1 ALTERNATIVE FLOW: SENDEMAIL:SMTPEERROR	13
TABLE 11 ID: 06 USE CASE: REALLOCATEBLANK	14
TABLE 12 ID: 06.1 ALTERNATIVE FLOW: REALLOCATEBLANK:NoAVAILABLEBLANKS.....	15
TABLE 13 ID: 07 USE CASE: ALLOCATEBLANK	15
TABLE 14 ID: 07.1 ALTERNATIVE FLOW: ALLOCATEBLANK:NoAVAILABLEBLANKS	16
TABLE 15 ID: 08 USE CASE: GENERATEREPORT	17

TABLE 16 ID: 09 USE CASE: MAINTAINDATA	17
TABLE 17 ID: 10 USE CASE: CREATECUSTOMERACCOUNT	18
TABLE 18 ID:10.1 ALTERNATIVE FLOW: CREATECUSTOMERACCOUNT:UNAVAILABLEEMAIL	19
TABLE 19 USE CASE INDEXED LIST	24