



UNIVERSITI TEKNOLOGI MARA
(UiTM) KEDAH, KAMPUS SUNGAI PETANI
SCHOOL OF INFORMATION SCIENCE
COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN INFORMATICS LIBRARY

[CDIM144]

PROGRAMMING FOR LIBRARIES [IML208]

GROUP ASSIGNMENT: TRAVEL AND TOUR MANAGEMENT SYSTEM

PREPARED BY:

NAME	STUDENT ID
NURAINA HANANI BINTI MOHD ZAMRI	2023483098
SITI NUR AMELIA BINTI MOHD KHALID RASIDI	2023855608
NUR FIDA FARHANA BINTI ROSLI	2023615074
NUR I'ZZAH RAIHANAH BINTI MOHD AMIN	2023840284
NUR IDAYU BINTI MD SANI	2023486364

CLASS: KCDIM144 3B

PREPARED FOR:

ENCIK MOHD FIRDAUS BIN MOHD HILMI

SUBMISSION DATE: TODAY2025

GROUP ASSIGNMENT: TRAVEL AND TOUR MANAGEMENT SYSTEM

NURAINA HANANI BINTI MOHD ZAMRI

2023483098

KCDIM144 3B

SITI NUR AMELIA BINTI MOHD KHALID RASIDI

2023855608

KCDIM144 3B

NUR FIDA FARHANA BINTI ROSLI

2023615074

KCDIM144 3B

NUR I'ZZAH RAIHANAH BINTI MOHD AMIN

2023

KCDIM144 3B

NUR IDAYU BINTI MD SANI

2023486364

KCDIM144 3B

UNIVERSITI TEKNOLOGI MARA (UiTM) KEDAH, KAMPUS SUNGAI PETANI

SCHOOL OF INFORMATION SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN LIBRARY INFORMATICS

2025

ACKNOWLEDGEMENT

Assalamualaikum w.b.t,

First, I would like to thank to Allah SWT for always give me strength and bless to complete this assignment, alhamdulillah.

On top of that, I would like to express my special thanks to my lecturer, Sir Mohd Firdaus Bin Mohd Hilmi for his time and energy in order to guide me finish this assessment. Without his encouragement I will not be able to do it myself.

Also, a million thanks for my classmates for all the advice they have given me. I have learnt a lot of things from them especially when I was facing some kind of difficulties regarding to the tasks.

Lastly, do not forget to mention my parents, a truly big appreciation to them for always being supportive on my passion, and believed that I can achieve my best on this assessment. Thank you very much.

TABLE OF CONTENT

CONTENT	PAGES
ACKNOWLEDGEMENT	
TABLE OF CONTENT	
PROJECT NAME & PROBLEM STATEMENT	
BACKGROUND OF THE SYSTEM	
FLOWCHART & PSEUDOCODE	
STRENGTH & KAIZEN (ROOM FOR IMPROVEMENT)	
PYTHON	

LIST OF TABLE AND FIGURE

PROJECT NAME: Tour and Travel Management system

PROBLEM STATEMENT:

1. Customers need go to physical counter to booking, cancelling and reschedule the ticket.
2. There's no other payment method except cash.
3. Customers only can buy the ticket between office hours.
4. Need to fill up handwriting form.
5. High risk of human error.

BACKGROUND OF THE SYSTEM

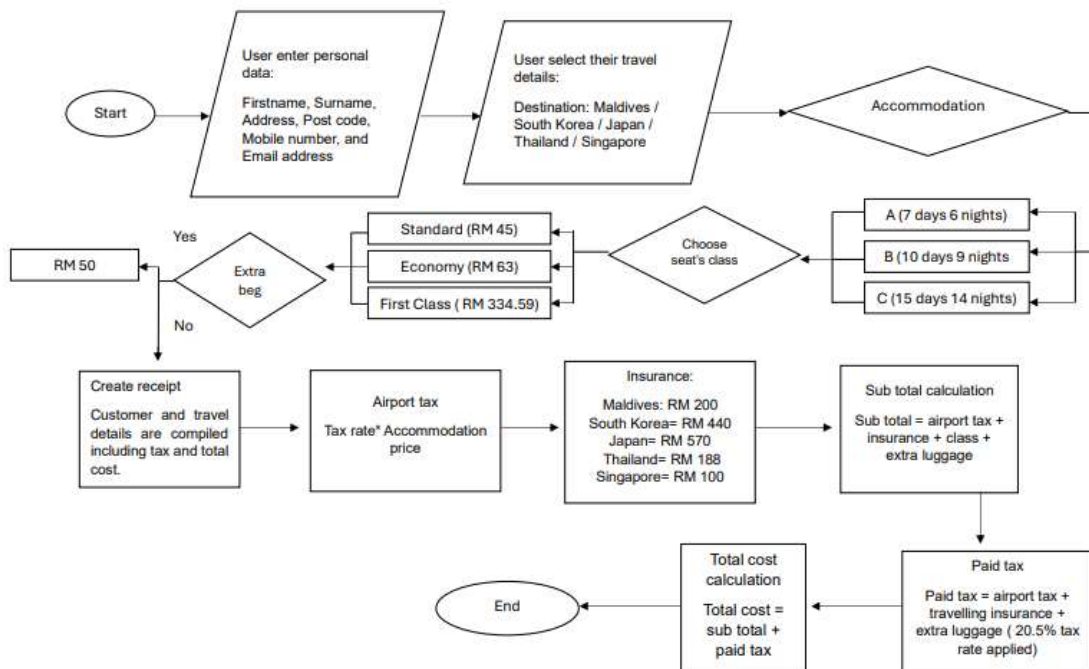
In digital world era, people are using application or website to book or give information for everyone. It can help people to know the information about what their want to search. Other than that, our group are choosing travel and tour management system for our group assignments. Travel and tour management systems is a code that used by any organization to manage service of traveling. It involves a technology for arrange and manage travel and tour from booking ticket until managing schedule for travel. In this system, we include a basic information which is first name, surname, address, post code, telephone number, email address, travel details and receipt.

Besides, travel and tour management system has function that can help users to booking and looking for some information before choosing a package. We can offer a various tourism package such as adventuring package, luxury vacation package and regular visit package for customer who want booking a travel ticket with us. We can make travel package following customers needed. This system can also book a flight ticket and accommodation. It can help customers to booking ticket, flight class, choosing travel date and destination. Thus, travel and tour management system also function to managing cancellation and conversion booking. This system can allow customer to change a date or their destination easily. If needed, customers can cancel their travel booking with a refund.

Furthermore, there are many benefits by using travel and tour management system which is it can help people to optimized cost-efficiency. By using payment method in online system, users can make a payment easily and increase a managing manual financial cost. It can help people to use system which is more effective in the future. Second benefits are it can be a systematic data storage that can help system to storage a user records regularly include travel history and booking information. It can help organizations to offer a better service in the future. Customer management and communication improvement is also a benefit for travel and tour management systems. It is because a system is completely in communication features such as email and notification messages that can help users to asking questions and improve communication and satisfaction.

Moreover, the system has many advantages that can be benefits for customer and users. travel and tour management system can be access 24/7 that can help customer to do a booking in every time and at anywhere. Because of that, we can plan a more accurate schedule without any distractions. It can also decrease human error that can be happened in manual booking that can potentially causing losses in the future.

FLOWCHART & PSEUDOCODE:



1. Start
2. Users need to prompt their personal data includes first name, surname, address, post code, mobile number and email address.
3. Users need to input their travel details involves departure and destination country, and accommodation type
4. Option to pay tax which is airport tax, air miles, travelling insurance and extra luggage
5. Choose seat's class: standard, economy or first class
6. Users can need to pay the fee RM 50 if their luggage more than 15 kg
7. Paid tax Calculation
 - Paid tax= airport tax + travelling insurance + extra luggage (20.5% tax rate applied)
8. Sub Total Calculation
 - Subtotal= airport tax + travelling insurance + class + extra luggage
9. Total cost calculation
 - Total cost= subtotal + paid tax
10. End

STRENGTH:

1. Offers the ability to efficiently manage cancellations and changes to reservations, including refunds when necessary.
2. Provides a variety of tourism packages like adventure, luxury, or regular packages to meet diverse customer needs.
3. Maintains records of travel history and booking information, enhancing organization and user experience.
4. Helps optimize costs by enabling online payments and reducing manual financial management efforts.

KAIZEN (ROOM FOR IMPROVEMENT):

1. Implement advanced security features for payment processing and data protection.
2. Provide real-time updates on booking statuses, availability, and notifications for changes.
3. Add support for multiple languages to accommodate international users.
4. Enable integration with popular travel platforms and tools such as Google Maps.
5. Introduce personalized recommendations based on customer preferences and past bookings.

PYTHON

PROJECT NAME: Travel Tour System

FILE NAME: TRAVELTOURSYSTEM.py

Prompt Data:

- i. First Name
- ii. Surname
- iii. Address
- iv. Post Code
- v. Mobile
- vi. Email

Function:

- i. Create Data

```
C:\> Users > khali > OneDrive - Universiti Teknologi MARA > SEM 3 > IML208-SEM 3 > TOURTRAVELSYSTEM.py > travel > _init_
5 class Travel:
64 def create_widgets(self):
65
66     # Customer Info frame
67     customer_frame = tk.Frame(self.root, bg='white', relief=tk.RIDGE, bd=8)
68     customer_frame.pack(side=tk.TOP, fill=tk.BOTH, expand=True, padx=15, pady=8)
69
70     labels = [
71         ("First Name", self.Firstname),
72         ("Surname", self.Surname),
73         ("Address", self.Address),
74         ("Post code", self.PostCode),
75         ("Mobile", self.Mobile),
76         ("Email", self.Email),
77     ]
78
79     for i, (label_text, variable) in enumerate(labels):
80         tk.Label(
81             customer_frame, text=label_text, font=('Arial', 10), bg='white'
82         ).grid(row=i, column=0, sticky="w", padx=5, pady=5)
83         tk.Entry(
84             customer_frame, textvariable=variable, font=('Arial', 10), width=25
85         ).grid(row=i, column=1, padx=5, pady=5)
86
87     # Travel Info frame
88     travel_frame = tk.Frame(self.root, bg='white', relief=tk.RIDGE, bd=8)
89     travel_frame.pack(side=tk.TOP, fill=tk.BOTH, expand=True, padx=15, pady=8)
90
91     # Destination Country
92     tk.Label(travel_frame, text="Destination Country", font=('Arial', 10), bg='white').grid(
93         row=0, column=0, sticky="w", padx=5, pady=10
94     )
95     destination_options = ["Maldives", "South Korea", "Japan", "Thailand", "Singapore"]
96     self.destination_menu = tk.OptionMenu(travel_frame, self.DestinationCountry, *destination_options, command=self.update_travel_info)
97     self.destination_menu.grid(row=0, column=1, padx=5, pady=10)
98
```

```

5 class Travel:
64     def create_widgets(self):
65
110         # Accommodation Options
111         tk.Label(travel_frame, text="Accommodation", font=('Arial', 10), bg='white').grid(
112             row=1, column=0, sticky="w", padx=5, pady=10
113         )
114         accommodation_options = ["A (7 days 6 nights)", "B (10 days 9 nights)", "C (15 days 14 nights)"]
115         self.accommodation_menu = tk.OptionMenu(travel_frame, self.Accommodation, *accommodation_options)
116         self.accommodation_menu.grid(row=1, column=1, padx=5, pady=10)
117
118         # Seat Class Selection
119         tk.Label(travel_frame, text="Seat Class", font=('Arial', 10), bg='white').grid(row=2, column=0, sticky="w", padx=5, pady=10)
120
121         self.check_standard = tk.Checkbutton(
122             travel_frame,
123             text="Standard (RM45)",
124             variable=self.var_standard,
125             font=('Arial', 10),
126             bg='white',
127         )
128         self.check_standard.grid(row=2, column=1, sticky="w")
129
130         self.check_economy = tk.Checkbutton(
131             travel_frame,
132             text="Economy (RM63)",
133             variable=self.var_economy,
134             font=('Arial', 10),
135             bg='white',
136         )
137         self.check_economy.grid(row=3, column=1, sticky="w")
138

```

```

C:\Users\khali> OneDrive - Universiti Teknologi MARA > SEM 3 > IML208-SEM 3 > TOURTRAVELSYSTEM.py > Travel > _init_
5 class Travel:
64     def create_widgets(self):
65
139         self.check_first_class = tk.Checkbutton(
140             travel_frame,
141             text="First Class (RM334.50)",
142             variable=self.var_first_class,
143             font=('Arial', 10),
144             bg='white',
145         )
146         self.check_first_class.grid(row=4, column=1, sticky="w")
147
148         # Extra Luggage Selection
149         tk.Label(travel_frame, text="Extra Luggage (More than 15KG)", font=('Arial', 10), bg='white').grid(
150             row=5, column=0, sticky="w", padx=5, pady=10
151         )
152         self.extra_luggage_check = tk.Checkbutton(
153             travel_frame,
154             text="Yes (RM50)",
155             variable=self.extra_luggage,
156             font=('Arial', 10),
157             bg='white',
158         )
159         self.extra_luggage_check.grid(row=5, column=1, sticky="w")
160
161         # Button Frame
162         button_frame = tk.Frame(travel_frame, bg='white')
163         button_frame.grid(row=6, column=0, colspan=2, pady=10)
164
165         tk.Button(
166             button_frame,
167             text="Create",
168             command=self.create_entry,
169             font=('Arial', 10),
170             width=8,
171         ).grid(row=6, column=0, padx=10)

```

ii. Read Data

```

C:\Users\khali> OneDrive - Universiti Teknologi MARA > SEM 3 > IML208-SEM 3 > TOURTRAVELSYSTEM.py > Travel > create_widgets
5 class Travel:
227     def create_entry(self):
228
238         receipt = (
239             f"First Name: {self.Firstname.get()}\n"
240             f"Surname: {self.Surname.get()}\n"
241             f"Address: {self.Address.get()}\n"
242             f"Post Code: {self.Postcode.get()}\n"
243             f"Mobile: {self.Mobile.get()}\n"
244             f"Email: {self.Email.get()}\n\n"
245             f"Destination Country: {self.DestinationCountry.get()}\n"
246             f"Accommodation: {self.Accommodation.get()}\n"
247             f"Ticket Class: {self.get_seat_class()}\n"
248             f"Airport Tax: RM{airport_tax}\n"
249             f"Insurance: RM{insurance}\n"
250             f"Extra Luggage Fee: RM{extra_luggage_fee}\n"
251             f"Subtotal: RM{subtotal}\n"
252             f"Paid Tax (20.5%): RM{paid_tax}\n"
253             f"Total Cost: RM{total_cost}\n"
254         )
255
256         self.txtReceipt.delete("1.0", tk.END)
257         self.txtReceipt.insert(tk.END, receipt)
258

```

iii. Update Data

```
C:\Users> khali > OneDrive - Universiti Teknologi MARA > SEM 3 > IML208-SEM 3 > TOURTRAVELSYSTEM.py > Travel > create_widgets

5 class Travel:
294     def update_entry(self):
295         # Implement the update functionality here
296         if len(self.data_list) == 0:
297             messagebox.showerror("No Entry", "No entries to update.")
298             return
299
300         # Updating the last entry for the demonstration purpose
301         entry = self.data_list[-1] # update the most recent entry
302
303         # Update values based on current user input
304         entry["FirstName"] = self.FirstName.get()
305         entry["Surname"] = self.Surname.get()
306         entry["Address"] = self.Address.get()
307         entry["Postcode"] = self.Postcode.get()
308         entry["Mobile"] = self.Mobile.get()
309         entry["Email"] = self.Email.get()
310         entry["Destination"] = self.DestinationCountry.get()
311         entry["Accommodation"] = self.Accommodation.get()
312         entry["SeatClass"] = self.get_seat_class()
313
314         self.create_entry() # To recalculate the cost and display the updated receipt
315
```

iv. Delete Existing Data

```
315
316     def delete_entry(self):
317         # Implement delete functionality
318         if len(self.data_list) == 0:
319             messagebox.showerror("No Entry", "No entries to delete.")
320             return
321
322         # Remove the most recent entry for the demonstration purpose
323         self.data_list.pop()
324
325         self.reset() # Reset the fields after deletion
326         self.txtReceipt.delete("1.0", tk.END) # Clear the receipt area
327         messagebox.showinfo("Delete", "Entry deleted successfully.")
328
```

Conditional Statement:

```
273
274     def calculate_seat_class_price(self):
275         price = 0
276         if self.var_standard.get():
277             price += 45
278         if self.var_economy.get():
279             price += 63
280         if self.var_first_class.get():
281             price += 334.59
282         return price
283
284     def get_seat_class(self):
285         seat_classes = []
286         if self.var_standard.get():
287             seat_classes.append("Standard (RM45)")
288         if self.var_economy.get():
289             seat_classes.append("Economy (RM63)")
290         if self.var_first_class.get():
291             seat_classes.append("First Class (RM334.59)")
292         return ", ".join(seat_classes)
293
294     def update_entry(self):
295         # Implement the update functionality here
296         if len(self.data_list) == 0:
297             messagebox.showerror("No Entry", "No entries to update.")
298             return
299
```

GUI: Yes

The screenshot shows a web browser window titled "Travel Management System". The page has a black header with the title in white. Below the header is a form with two main sections. The top section contains input fields for "First Name", "Surname", "Address", "Post Code", "Mobile", and "Email". The bottom section contains a "Destination Country" dropdown menu, an "Accommodation" dropdown menu, a "Seat Class" section with three radio buttons labeled "Standard (RM45)", "Economy (RM63)", and "First Class (RM334.59)", and an "Extra Luggage (More than 15KG)" section with a checkbox labeled "Yes (RM50)". At the bottom of the form are four buttons: "Create", "Reset", "Update", and "Delete".

Result:

i. Create Data

The screenshot shows the same web browser window as before, but now with data entered into the form. The "First Name" field contains "Amelia", "Surname" contains "Siti", "Address" contains "Imbah Bujang, Merbok Kedah", "Post Code" contains "08100", "Mobile" contains "0147965413", and "Email" contains "Amly@gmail.com". The "Destination Country" dropdown menu is set to "South Korea", and the "Accommodation" dropdown menu is set to "C (15 days 14 nights)". The "Seat Class" section has the "First Class (RM334.59)" radio button selected, and the "Extra Luggage (More than 15KG)" section has the "Yes (RM50)" checkbox checked. The "Create", "Reset", "Update", and "Delete" buttons are still present at the bottom.

- ii. Read Data
- iii. Update Data
- iv. Delete Existing Data

REFERENCES

DJ Oamen. (2018, August 7). *How to Create Travel Management System in Python – Full Tutorial* [Video]. YouTube. <https://youtu.be/x3Syb3N0E6o?si=38okXI3bwkxfNXQ6>