



## CS426 - Mobile Device Application Development

# MIDTERM PROJECT

## Report

### Student Information

- Name: Bùi Danh Nghệ
- Student ID: 22125063

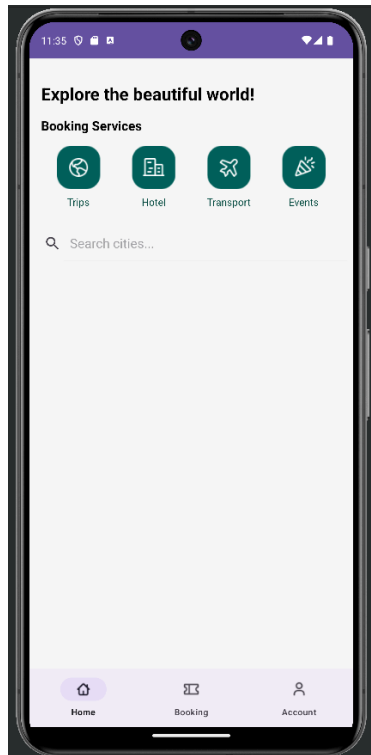
### Project Self Assessment

Link to video:

[https://drive.google.com/drive/folders/1ceJAeLivhW6--x07MTGUjF3wRNm\\_Ucmw?usp=sharing](https://drive.google.com/drive/folders/1ceJAeLivhW6--x07MTGUjF3wRNm_Ucmw?usp=sharing)

- Self-assess your score in the evaluation column
- Replace the app design images with your app screenshots

## Home



Basic layout

2

2

Create a search bar that allows users to input text and click a search button to trigger an action. (such as displaying a dialog with the search text.)

2

2

Bottom navigation

5

5

Booking services

5

5

When clicking on one of the booking services in the list -> switch to the corresponding booking selection screen. Currently only supports Transport Booking.

2

2

Booking			



The layout features 4 booking services in a card view. You can find all the card view resources in Figma. Currently only supports **Transport Booking**

5

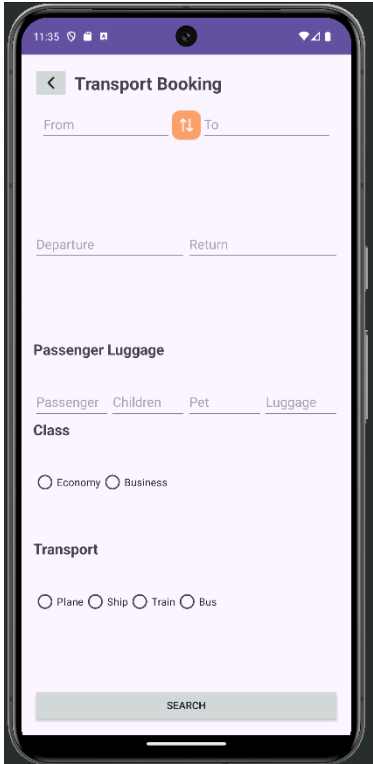
5

When clicking on one of the booking services in the list -> switch to the corresponding booking selection screen.

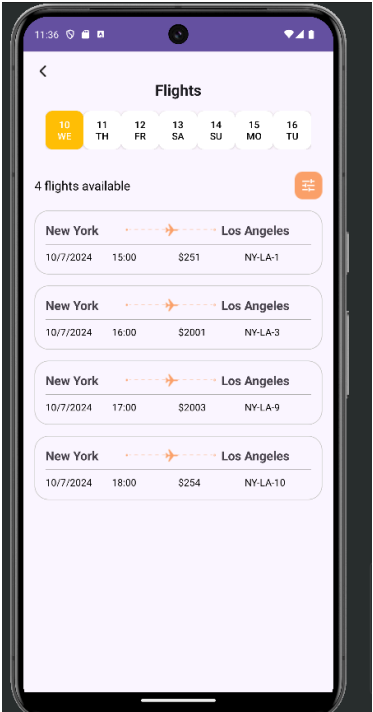
2

2

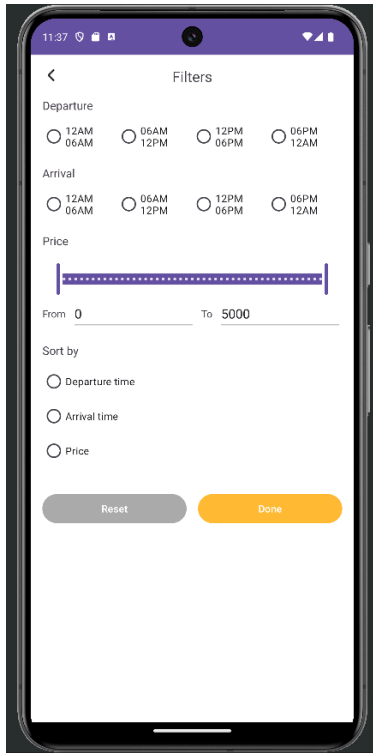
Transport Booking



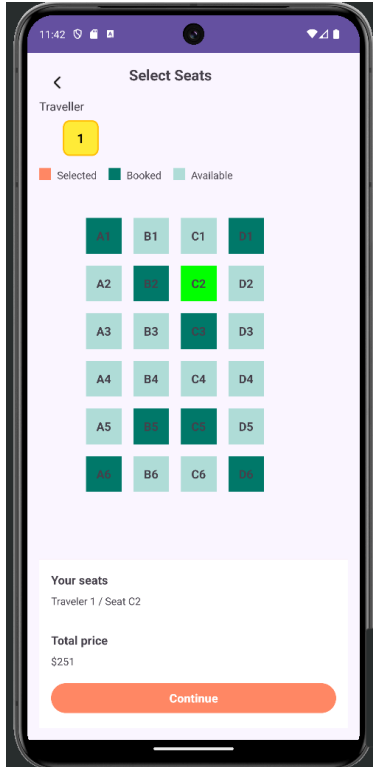
Basic layout	2	2
Choose the departure city and arrival city. Allow swap the departure and arrival city when hit the swap button.	5	5
Allows users to choose the date of departure and return flights. The return flight date must be after the departure date	5	5
Design and implement an friendly UI that allow users to input the number of Passenger & Luggage	5	5
Allows users to choose a transportation method. Currently only supports flight selection	2	2
Click Search to switch to the <b>Transport Flights</b> screen	2	2
Transport Flights		



Basic layout	2	2
Click to change flight date, displaying a list of corresponding flights from the departure and arrival cities	5	5
Display detailed flight list information	5	5
Click on filter icon, switch to <b>Filters</b> screen	2	2
Click on flight details, switch to <b>Select Seats</b> screen	2	2
Transport - Filters		



Basic layout	2	2
Allows users to choose departure and arrival times	3	3
Allows users to choose the amount to filter by text input field or slider	5	5
Users can choose specific information fields for sorting.	2	2
Click Reset to reset the default filter information	2	2
Click Done to return to the <b>Transport Flights</b> screen with flights that meet the filter information	5	5
Transport - Select Seats		



Basic layout	2	2
Allows you to select seats for each person in the list	5	5
Displays the same layout as in the design with selected, booked, available seats	5	5
When selecting a seat, the corresponding selected seat information is displayed at the bottom of the screen	5	5
Click continue to move to the <b>Boarding pass</b> screen	2	2
Transport - Boarding Pass		





Basic layout

Design a user interface that visually represents tickets based on the number of people and the selected seats.

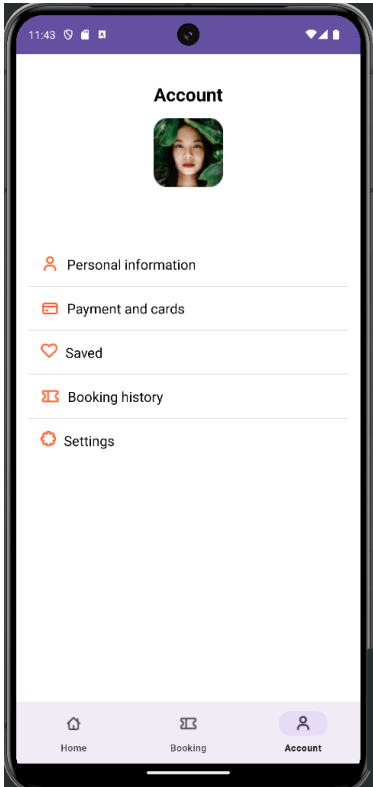
5

5

5

5

Account



Basic layout

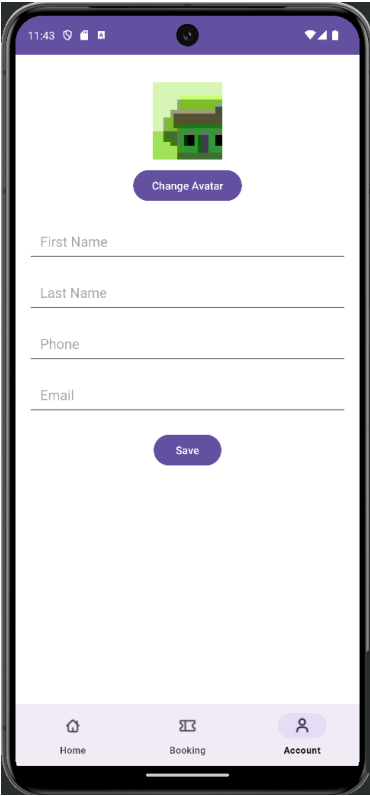
Click on *personal information* to switch to the ***personal information*** screen, other cases notify the user that this function will be developed

2

2

5

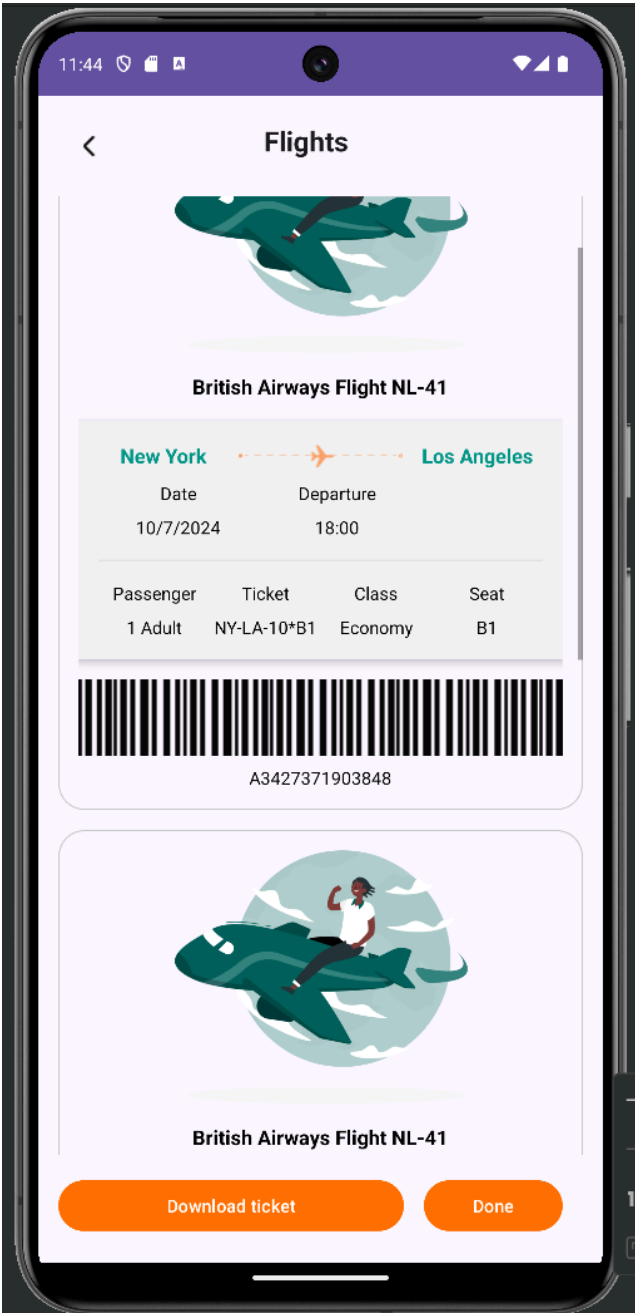
5

<b>Account - Personal Information</b>			
	Basic layout	2	2
	Allow user to update their profile picture	3	3
	Allow user to edit and update their information	5	5
<b>General application requirements</b>			
	Manage application lifecycle, application state	20	20
	Initialize data for the application	10	10
	New features/requirements added by yourself	15	10
<b>Total</b>	<b>Min(150, Total score of all requirements above)</b>	<b>150</b>	<b>145</b>
<b>Additional Criteria</b>	<b>Source code + report + demo lean and clear</b>	<b>[-15,+15]</b>	<b>10</b>

Your score: $\text{Min}(10, (\text{Total} + \text{Additional Criteria}) / 15)$	10
--	----

## Insights

- **Content of New Features:** Briefly describe the functionality and purpose of each new feature (if any).  
New feature: Booking History allow user to view their history booking ticket. After click download ticket, the ticket is saved then can be opened again in this section. Even if the app is turn off, the ticket is still can be opened again  
-> Purpose of this feature is allowing user to view their history ticket right in the app even after closing it, save user time from finding their purchase ticket. Only latest ticket will be saved.



- **Techniques Applied:** Discuss the specific technical approaches used to implement the application features. Did you encounter any challenges or require new skills?

There are a lot of challenges: Design complex UI layout, manage flight data, including storage, retrieval, and displaying it dynamically, managing asynchronous tasks, or handling user input validation, using gradle,... I have learn many new skills to complete this project: gradle, searching, apply many android studio tutorials,...

- **Implementation Experience:** Share your personal experience with implementing this project. What aspects did you find most enjoyable or challenging? What did you learn from the process?

The most enjoyable part of this project is that i must plan what to design (UI+backend) after start implement each activity, UIUX. The most challenging part is handle seat activity, data handling, filter activity, the logic between activities,... I learned how to search effectively, using chatGPT to do some part of the work, plan detailed to apply, some complex data handling between activities

## References

You should list all source codes and/or references that you use in your project in this section.

Reference: chatGPT, i asked chatGPT a lot but most of codes i must modify before putting in my project.

Any code fragments or libraries that are not yours **MUST be explicitly declared** in this section.

```
implementation("com.google.code.gson:gson:2.8.8")
```

gson : to pass list array to sharepreferenced

If you fail/forget to declare those “inherited” resources, you will be considered “**cheating**”!