

# Test Project

Module D Development and Testing  
Day 4

## *Mobile Applications Development*

Independent Test Project Designer: ITPDs

Independent Test Project Validator: Junbin Fang SCM

# Contents

Introduction .....	3
Description of project and tasks .....	3
General Demands.....	4
<i>Develop by Test Case</i> .....	5
Automated Test.....	10
Instructions to the Competitor .....	13
Automated Test Guide .....	14

# Introduction

X-French Travel plans to develop several apps to help people learn about French tourist attractions, history, and culture and promote the flourishing development of local tourism.

This time, the App is designed to meet tourists' needs for booking hotels in the Alps. It provides comprehensive hotel information and booking services, simplifies the booking process, and improves users' travel experience in the Alps.

In this module, you need to develop the basic App functional logic and write automated test scripts to run the App.

## Description of project and tasks

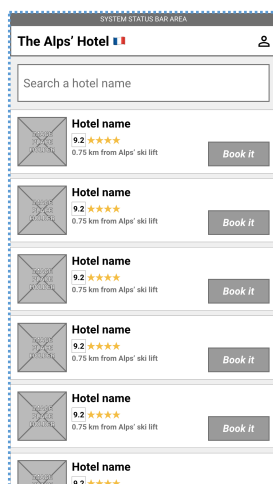
You should push the full project and an installable apk (Android) or app (iOS) package to Git Server.

Competition time: 3 hours

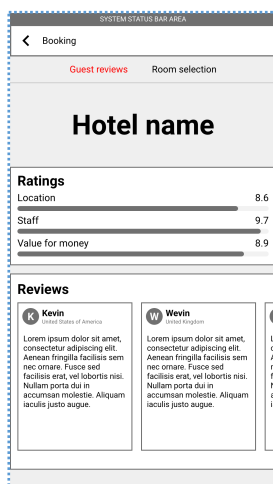
Marking Emulator:

- iOS: iPhone 8 Plus
- Android: Pixel 2

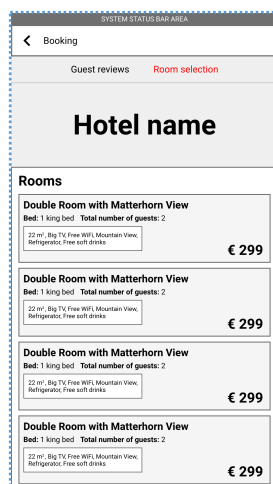
The wireframes are for reference only. You do not need to replicate the exact appearance, colors, decorative icons/images, or elements' positions/styles.



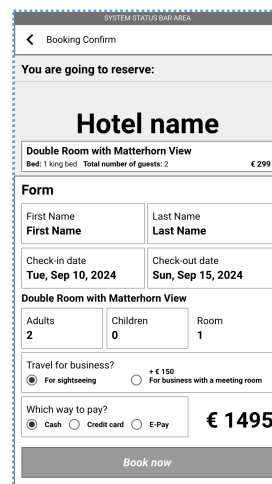
Home page



Booking Details page  
(guest reviews)



Booking Details page  
(room selection)



Booking Confirm page

SYSTEM STATUS BAR AREA	
< My bookings	
List of my bookings	
<div> <div>First Name Last Name</div> <div>Hotel name</div> <div>1</div> <div>Tue, Sep 10, 2024 to Sun, Sep 15, 2024</div> <div>2 Adults, 0 Children, 1 Room</div> <div>For sightseeing Pay with cash</div> <div>€ 299</div> </div>	
<div> <div>First Name Last Name</div> <div>Hotel name</div> <div>2</div> <div>Tue, Sep 10, 2024 to Sun, Sep 15, 2024</div> <div>2 Adults, 0 Children, 1 Room</div> <div>For sightseeing Pay with cash</div> <div>€ 299</div> </div>	
<div> <div>First Name Last Name</div> <div>Hotel name</div> <div>3</div> <div>Tue, Sep 10, 2024 to Sun, Sep 15, 2024</div> <div>2 Adults, 0 Children, 1 Room</div> <div>For sightseeing Pay with cash</div> <div>€ 299</div> </div>	

My Bookings page

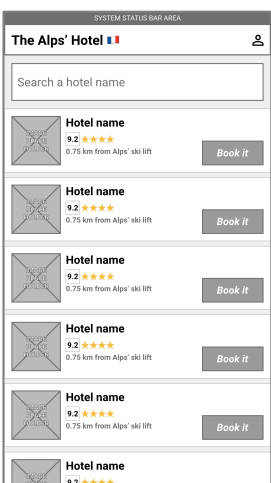
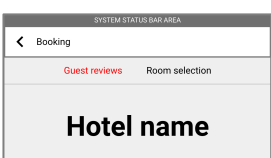
## General Demands

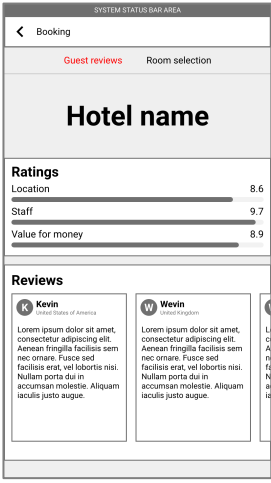

1. All data you need are in **media-files/json-data**.
2. Your application functions should be developed according to all test cases (Form 1.1/1.2) and the automated function test case (Form 2.1).


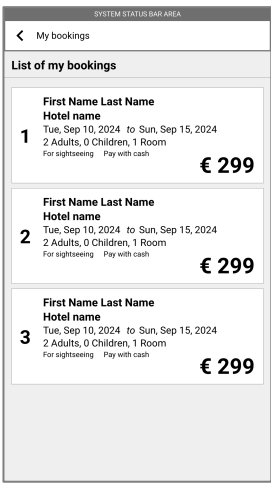
# Develop by Test Case

Develop the application by understanding the test cases.

## FORM 1.1: APPLICATION UI TEST CASE

No.	Area	Demands
1	Home page 	<ol style="list-style-type: none"> <li>1. The title bar contains the app name, the French flag, and an icon button in the top right corner;</li> <li>2. The homepage contains a search bar and a list of bookable hotels;</li> <li>3. The search bar displays the hint text "Search a hotel name";</li> <li>4. The search bar allows finding hotels based on the entered text;</li> <li>5. The Search button should be shown on the system keyboard;</li> <li>6. Each hotel list item includes: <ol style="list-style-type: none"> <li>(a) Hotel name;</li> <li>(b) Hotel ratings and star rating bars;</li> <li>(c) Distance-- "XX km from Alps' ski lift";</li> <li>(d) Book Button;</li> <li>(e) Hotel cover image.</li> </ol> </li> <li>7. Click the booking button of a hotel list item to navigate to the corresponding Booking Details page (Only data with hotel_id 1000 and 1008 need to be implemented);</li> <li>8. Click the icon button in the top right corner to navigate to the My Bookings page.</li> </ol>
2	Booking Details (top bar) 	<ol style="list-style-type: none"> <li>1. The title bar contains a back button and the text "Booking";</li> <li>2. Click the back button to navigate to the previous page;</li> <li>3. The hotel information bar shows the hotel name;</li> <li>4. Use tabs on the subpages (guest reviews, room selection) to navigate to the corresponding pages.</li> </ol>

2.1	<p>Booking Details page (guest reviews)</p> 	<ol style="list-style-type: none"> <li>The Ratings area includes: <ol style="list-style-type: none"> <li>Title-- "Ratings";</li> <li>Several rating items;</li> <li>Each rating item includes progress bars and scores.</li> </ol> </li> <li>The Reviews area includes: <ol style="list-style-type: none"> <li>Title-- "Reviews";</li> <li>Horizontal list of reviews;</li> <li>Each reviews list item includes name, nationality, circle shape and review content;</li> <li>The initial in circle shape represents the first letter of the reviewer's name.</li> </ol> </li> </ol>
2.2	<p>Booking Details page (room selection)</p> 	<ol style="list-style-type: none"> <li>The Rooms area includes: <ol style="list-style-type: none"> <li>Title-- "Rooms";</li> <li>A list of rooms.</li> </ol> </li> <li>Each room includes: <ol style="list-style-type: none"> <li>Room Type;</li> <li>Bed type-- "Bed: XX";</li> <li>"Total number of guests: XX";</li> <li>Room features introduction text;</li> <li>"€" icon with price text.</li> </ol> </li> <li>Click a bookable room list item to navigate to the corresponding Booking Confirm page.</li> </ol>

3	<p>Booking Confirm page</p> 	<ol style="list-style-type: none"> <li>1. The title bar includes a back button, text "Booking Confirm", and the text "You are going to reserve:";</li> <li>2. Click the back button to return to the previous page;</li> <li>3. Hotel information comes from previous selection;</li> <li>4. The Form area includes: <ol style="list-style-type: none"> <li>(a) "First Name" and "Last Name" input boxes;</li> <li>(b) "Check-in date" and "Check-out date" input boxes;</li> <li>(c) Room type to be booked;</li> <li>(d) "Adults" and "Children" input boxes;</li> <li>(e) Automatically calculated number of "Rooms";</li> <li>(f) "Travel for business?" radio button with sub-options-- "For sightseeing" and "+ €150 For business with a meeting room";</li> <li>(g) "Which way to pay?" radio button with sub-options-- "Cash", "Credit card" and "E-Pay";</li> <li>(h) "€" icon with automatically calculated price text;</li> <li>(i) "Book now" button.</li> </ol> </li> </ol>
4	<p>My Bookings page</p> 	<ol style="list-style-type: none"> <li>1. The title bar includes the back button, and text "My Bookings";</li> <li>2. "List of my bookings" subtitle;</li> <li>3. Booking List;</li> <li>4. Each list item includes: <ol style="list-style-type: none"> <li>(a) Numerical sequence (starting from 1);</li> <li>(b) Name of the person who made the reservation;</li> <li>(c) Start and end dates (formatted as: Tue, Sep 10, 2024);</li> <li>(d) Number of adults, children and rooms;</li> <li>(e) Booking type and payment method;</li> <li>(f) Price.</li> </ol> </li> </ol>

## FORM 1.2: APPLICATION TEST CASE

No.	Area	Demands			
1	Information input area		Trigger Timing	Constraints Type	Constraints Description
	<b>Informations</b> <div> <div> <div>Form</div> <div> <div>First Name First Name</div> <div>Last Name Last Name</div> </div> <div> <div>Check-in date Tue, Sep 10, 2024</div> <div>Check-out date Sun, Sep 15, 2024</div> </div> <div> <div>Double Room with Matterhorn View</div> <div> <div>Adults 2</div> <div>Children 0</div> <div>Room 1</div> </div> <div> <div>Travel for business?</div> <div> <input checked="" type="radio"/> For sightseeing <input type="radio"/> For business with a meeting room </div> </div> <div> <div>Which way to pay?</div> <div> <input checked="" type="radio"/> Cash <input type="radio"/> Credit card <input type="radio"/> E-Pay </div> </div> <div>€ 1495</div> <div>Book now</div> </div> </div> </div>	First Name Input and Last Name input	Click "Book now" Button	length	[1, 15]
				validity	Required, only letters can be entered
		Input box for the number of adults and children	Click "Book now" Button	length	Adults: [1, 5] Children: [0, 2 times the number of adults]
				validity	Required, cannot be children only, only numbers can be entered
				defaults	Adults: 1 Children: 0
		Date input box	Click "Book now" Button	format	(i.e. Tue, Sep 10, 2024)
				validity	Required, each value should be a valid date, and the start date should be earlier than the end date.

### Note:

- When the number of occupants changes, the number of rooms is automatically calculated based on the total number of people;
- The price is automatically recalculated based on the number of rooms, the type of reservation and length of stay whenever the number of rooms, the reservation type or the date changes;
- The date input box accepts multiple formats and automatically converts them to the correct format (e.g. "Tue, Sep 10, 2024"). Acceptable input formats only include "09/10/2024", "09-10-2024" and "Sep 10 2024";
- Clicking the "Book now" button triggers data verification.



- (a) If an error is detected, a corresponding prompt will pop up.
- (b) If the verification is successful, a prompt "Are you going to book this room?" will pop up. Clicking "No" closes the dialog box; clicking "Yes" saves the reservation information, close the dialog box, and navigates to the My Bookings page.

## Automated Test

Finish the automated test script according to the Test Project description.

The test script files must be stored in the location specified in the **Automated Test Guide**.

All steps need to be held on for at least 10 seconds, and the "Step No: \$Steps\_No,\$Description of the Action" should be logged in the "output" area, such as terminal, debug area or console.

### FORM 2.1: APPLICATION FUNCTION TEST CASE

Program Name		The Alps’ Hotel		
Test Type		Function Test		
Test Method		Black Box Test, Automated Test		
Test Demand		The workflow of the application functions runs normally.		
Test Execution Steps				
Steps_No	Page	Description of the Action	Input Data	Expected Results
1		Application startup		Application starts normally
2	Home	Scroll to the bottom of the hotel list		The hotel list is successfully scrolled to the bottom
3	Home	Perform a search after entering the search term in the search bar	Méribel	Searching the hotel list results in the listing "Appartement Méribel"
4	Home	Click the listing item "Appartement Méribel"		Go to the Booking Details page for the hotel "Appartement Méribel" and display the Overview subpage
5	Booking Details (overviews)	Click the Guest reviews tab button		Display the Guest reviews subpage
6	Booking Details (guest reviews)	Scroll down the Reviews list to the 4th Review (published by "Miles")		Successfully scroll the Reviews list to the 4th

				Review item (publisher name is "Miles")
7	Booking Details (guest reviews)	Click the Room selection label button		Display the Room selection subpage
8	Booking Details (room selection)	Scroll down the room list to the "Great Family Room" item		Successfully scroll the room list to the "Great Family Room" item
9	Booking Details (room selection)	Click the "Great Family Room" item		Navigate to the Booking Confirm page of the hotel's Great Family Room
10	Booking Confirm	Enter First Name	Taylor	The First Name input box shows "Taylor"
11	Booking Confirm	Enter Last Name	Hutchinson	The Last Name input box displays "Hutchinson"
12	Booking Confirm	Enter check-in date	Oct 15 2024	The check-in date is displayed in the correct format
13	Booking Confirm	Enter check-out date	2024/2/10	The check-out date is displayed as "2024/2/10"
14	Booking Confirm	Enter the number of adults	4	The number of adults is shown as 4 and the number of rooms is shown as 2
15	Booking Confirm	Enter the number of children	3	The number of children is shown as 3 and the number of rooms is shown as 3
16	Booking Confirm	Click the E-Pay radio button		The Cash button is disabled; E-Pay radio button is enabled
17	Booking Confirm	Click the "Book now" button		An error message pop-up window appears

18	Booking Confirm	Change check-out date	10-19-2024	The check-out date is displayed in the correct format
19	Booking Confirm	Click the "Book now" button		A confirmation pop-up window pops up to confirm the reservation
20	Booking Confirm	Click "Yes" in the pop-up window to confirm your reservation.		Navigate to the My Bookings page, and the booking list shows the information of the newly completed booking

## Instructions to the Competitor

1. Please create the project with correct package name (Android)/ Organization Identifier (iOS). The naming convention is edu.ws2024.dXX;
2. You should save the project in the folder XX\_Module\_D;
3. You should rename the generated apk (Android)/ app (iOS) file as XX\_Module\_D.apk (Android)/ XX\_Module\_D.app (iOS) and save it in the root of the XX\_Module\_D folder;
4. The whole XX\_Module\_D folder should be pushed to the provided remote Git repository in time;
5. Note: "XX" is your workstation code.

# Automated Test Guide

Automated test dependencies configuration guide and scripts storage demands:

- If you use native Android platform, you should use "UI Automator" to finish the automated test scripts. You must write your UI test scripts entrance method "startTesting" in ApplicationUITesting class in androidTest directory. You must make sure that UI test in Android Studio can be started by clicking the start run button of the method: "startTesting".

Before building your UI test with UI Automator, make sure to configure your test source code location and project dependencies. In the build.gradle file of your Android app module, you must set a dependency reference to the UI Automator library:

```
dependencies {
    ...
    androidTestImplementation 'androidx.test.uiautomator:uiautomator:2.2.0'
}
```

- If you use Flutter platforms, you should use 'flutter\_test' and 'integration\_test' packages to finish the automated test scripts. You must write your UI test scripts in integration\_test/app\_test.dart file. You must make sure that the UI test can be started by running the console command:

1. "flutter pub get --offline";
2. "flutter test --no-pub integration\_test/app\_test.dart".

Before building your UI test with flutter\_test and integration\_test packages, make sure to configure your project dependencies. In the pubspec.yaml file of your flutter module, you must set dependencies reference to the flutter\_test and integration\_test packages:

```
dev_dependencies:
  flutter_test:
    sdk: flutter
  integration_test:
    sdk: flutter
```

- If you use native iOS platforms, you should finish the automated test scripts. During the marking process, the Experts will mark your UI test with the Xcode. You must make sure that the UI test in Xcode can be started by clicking the start run button of the method: "testStart" in "[Project Name]UITests.swift" or "[Project Name]UITests.m".

When you create a project, select the Include Unit Tests and Include UI Tests checkboxes that appear under the Language pop-up menu on the first sheet. These options preconfigure your project with [Project Name]Tests and [Project Name]UITests targets. In the Test navigator, you can view and edit the code for these tests, and add additional tests. Then run the UI tests and unit tests on simulated and connected devices.