

# Test Project

Module A Functionality  
Day 1 P.M.

*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
<b>Pages Demands .....</b>	<b>4</b>
Home Page.....	4
<i>Elements include: .....</i>	<i>4</i>
<i>Functional requirements: .....</i>	<i>5</i>
Travel page.....	6
<i>Elements include: .....</i>	<i>6</i>
<i>Functional requirements: .....</i>	<i>6</i>
Account page (before sign in).....	7
<i>Elements include: .....</i>	<i>7</i>
<i>Functional requirements: .....</i>	<i>7</i>
Account page (after sign in).....	8
<i>Elements include: .....</i>	<i>8</i>
<i>Functional requirements: .....</i>	<i>8</i>
<b>Instructions to the Competitor .....</b>	<b>9</b>
Web API documents .....	9

# 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.

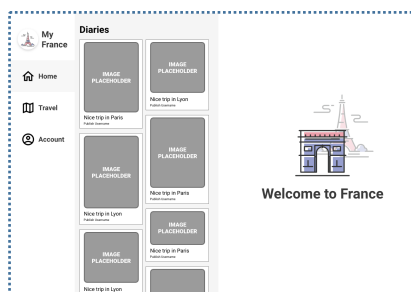
Today, the Product Manager has provided you with product wireframes. In the morning, it is a phone game app about skiing; in the afternoon, it is a tablet app related to a travel diary. As a Mobile Application Developer, your task is to develop corresponding apps based on the given wireframes according to the specific development needs.

## Description of project and tasks

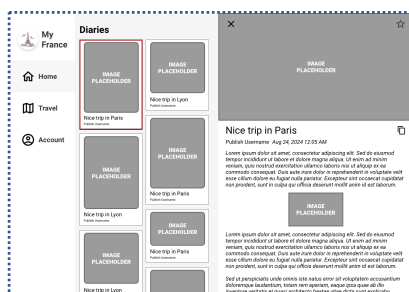
You do not need to consider the program interface's aesthetics and the elements' position. The work can be inconsistent with the given wireframes, but you need to consider the ease of use. The development task is divided into two parts depending on the devices.

	PART	NAME	COMPETITION TIME	MARKING DEVICES
	A.M.	Functionality (phone)	2.5h	Emulator: Pixel 2 or iPhone 8 Plus
✓	P.M.	Functionality (tablet)	2.5h	Emulator: Pixel C or iPad Pro (9.7-inch)

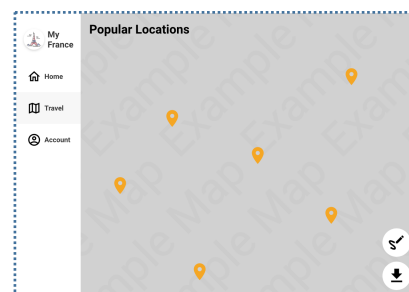
Below is an overview of the wireframes. Please check the **media-files** for details.



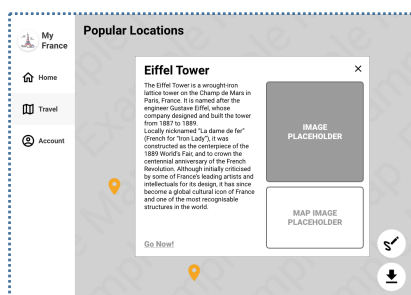
Home page



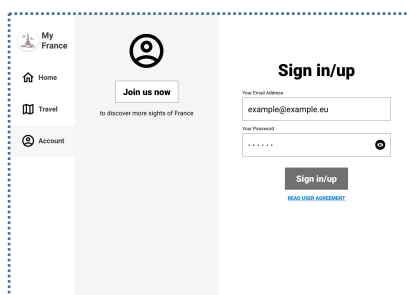
Home page (with a diary)



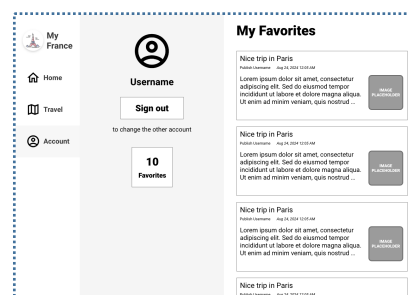
Travel page



Travel page (with a dialog)



Account page (before sign in)



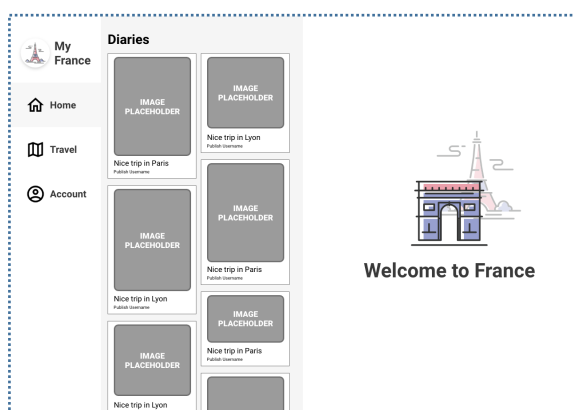
Account page (after sign in & my favorites)

## General Demands

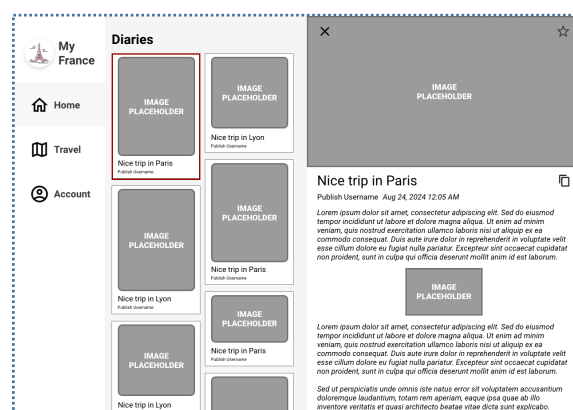
1. Finish the function of this application according to the demands;
2. The application needs to display an application icon in the operating system. The icon needs to use the "[art\\_icon\\_la\\_tour\\_eiffel.png](#)" in the material;
3. The application must display the application name "My France" in the operating system;
4. All of the pages should contain a navigation bar on the left;
5. Left navigation bar:
  - (a) The application icon is on the navigation bar;
  - (b) "Home" button with icon, click to navigate to Home page;
  - (c) "Travel" button with icon, click to navigate to Travel page;
  - (d) "Account" button with icon, click to navigate to Account page;
  - (e) When clicking the navigation button, the button is highlighted.

## Pages Demands

### Home Page



Home page



Home page (with a diary)

#### Elements include:

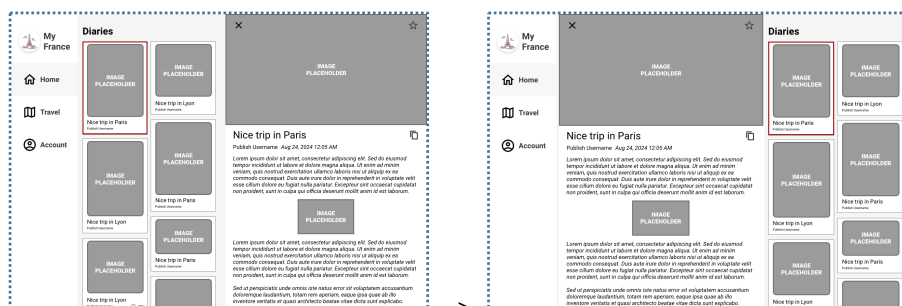
1. "Diaries" Text;
2. Travel diaries summary list, each item:
  - (a) Thumbnail with rounded corners;
  - (b) Title of the diary;
  - (c) Publisher's username.
3. "Welcome to France" Text;
4. Paris landmarks;
5. When a diary item is active, the diary detail area:
  - (a) Diary image which fills the max width of right side;
  - (b) Title of diary;
  - (c) Publish username;
  - (d) Publish date-time;
  - (e) Scrollable main text;
  - (f) Copy icon button (📋);
  - (g) Favorite icon button (★);
  - (h) Close icon button.

## Functional requirements:

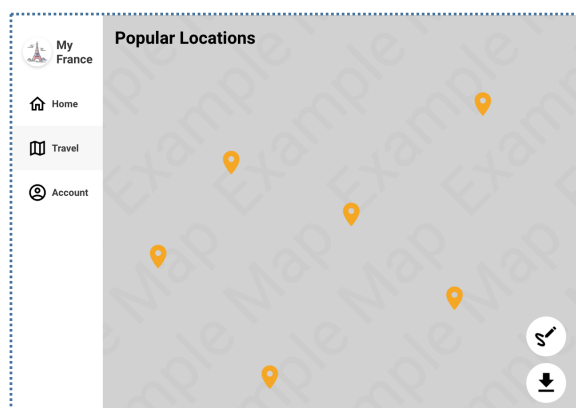
1. The travel diary data is fetched from the corresponding Web API;
2. The travel diary list is displayed in a vertically scrollable waterfall layout (Staggered Grid) list, with content displayed correctly;
3. By default, the right area displays an illustration of Paris landmarks, incorporating the images "art\_icon\_la\_tour\_eiffel.png" and "art\_icon\_la\_tour\_eiffel.png" from media-files, as shown below:



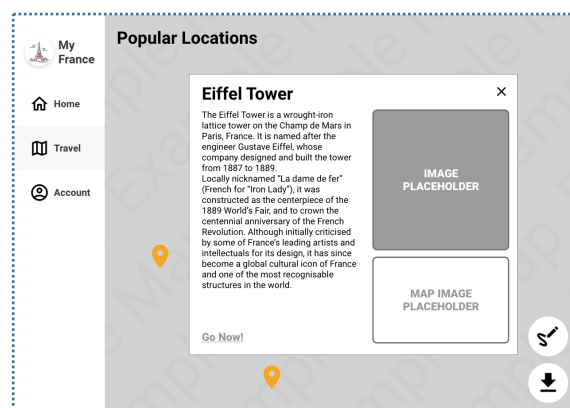
4. Clicking on a travel diary list item highlights the selected item and displays its details in the right area;
5. Clicking the Close button closes the diary details and returns to the default status;
6. The travel diary's main text (in JSON format) returned by the Web API contains image relative paths (e.g., "resources/Efil/2.jpg"). Please replace these image relative paths with displayable images;
7. The publish date-time must be displayed in the specified format-- "Aug 24, 2024 12:05 AM";
8. Clicking the favorite button:
  - (a) If the user is not signed in, a prompt "Please sign in" pops up;
  - (b) If the user is logged in, the favorite button turns yellow (☆->★), and the number of favorites on the Account page increases by one.
9. Clicking the copy button:
  - (a) Copy the travel diary title to the clipboard;
  - (b) After successful copying, a message pops up to prompt that the copy was successful.
10. Swap the positions of the diary detail area and the travel diary summary list area by dragging and dropping:



## Travel page



Travel page



Travel page (with a dialog)

### Elements include:

1. Page title "Popular Locations";
2. Zoom-able map image;
3. Location marker icon (📍);
4. Draw icon button (✍️);
5. Save icon button (💾);
6. Attraction introduction dialog:
  - (a) Attraction name;
  - (b) Attraction introduction main text;
  - (c) Attraction image;
  - (d) Attraction location map;
  - (e) Underlined text button: "Go Now!";
  - (f) Close icon button.

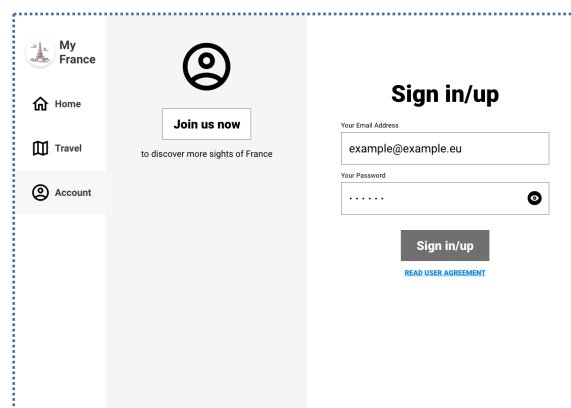
### Functional requirements:

1. The Attractions' information data comes from the JSON file;
2. Map images can be manipulated using finger gestures:
  - (a) Stretch to zoom in;
  - (b) After zooming in, the user can long press and drag with one finger.
3. The location marker icon always maintains the correct position and fixed size on the map;
4. Tapping the draw icon button highlights it, allowing users to freely draw red shapes on the map;
5. Drawn shapes always maintain the correct position and scale on the map;
6. Click the save button to save the map, location markers, and drawn shapes as an image in the system photo gallery;
7. Attraction introduction dialog:
  - (a) Click the location marker icon to open the introduction dialog box;
  - (b) Click the Close button to close the introduction dialog box;
  - (c) Click the "Go Now!" button to open the map app (Google/Apple Maps);
  - (d) Click on the scenic spot picture to open a new window displaying the full picture;
  - (e) The Attraction name can be dragged and dropped into Messages to share in split-screen mode (refer to the media-files' video: [drag\\_and\\_drop\\_demo\\_video\\_android.mp4](#) or [drag\\_and\\_drop\\_demo\\_video\\_ipad\\_os.mp4](#)).

## Account page (before sign in)

### Elements include:

1. User avatar icon;
2. Button "Join us now";
3. Text "to discover more sights of France";
4. Sign in/up area:
  - (a) Title "Sign in/up";
  - (b) Email address input box with the text "Your Email Address";
  - (c) Password input box, the text "Your Password", and an eye-shaped icon button (👁️);
  - (d) Button "Sign in/up";
  - (e) Blue underlined text "READ USER AGREEMENT".



Account page (before sign in)

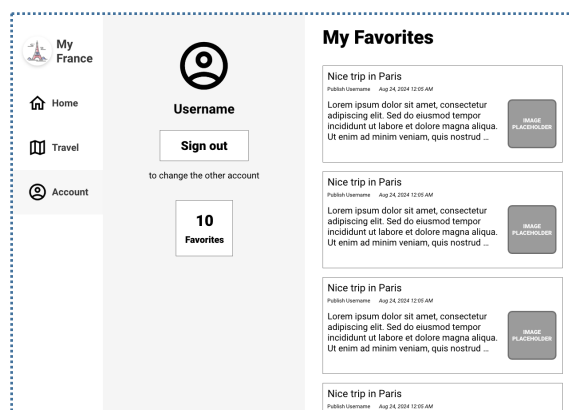
### Functional requirements:

1. Sign in/up requires the use of Web API;
2. By default, the Sign in/up area is not displayed;
3. Click "Join us now" to display the Sign in/up area;
4. The email input box verifies the correctness of the email format in real time:
  - (a) If it does not conform to the correct format, display a red message "Invalid Email";
  - (b) If the format is correct, do not display any prompt.
5. The password input box verifies the correctness of the password format (at least 6 characters including English letters and numbers) in real time:
  - (a) If the password does not conform to the correct format, displays a red message "Invalid Password";
  - (b) If the format is correct, do not display any prompt.
6. Password characters are displayed in ciphertext;
7. In the password input box, pressing and holding the eye-shaped icon button displays the password characters in plain text; releasing your finger returns the password characters to ciphertext;
8. Clicking "READ USER AGREEMENT" opens a WebView dialog in the app and displays the user agreement webpage with the CSS style correctly applied;
9. The WebView dialog can only be closed by clicking the close button on the user agreement webpage;
10. When a user signs in with an incorrect password, a system notification pops up to inform the user of the reason for the login failure;
11. When the user successfully signs in with the correct password, a prompt tone (*sign\_in\_success.mp3*) is played
12. Upon successful sign-in with correct information, the page displays the content of the Account page (after sign in).

## Account page (after sign in)

### Elements include:

1. User avatar icon;
2. The username of the currently signed in user;
3. Button "Sign out";
4. Text "to change the other account";
5. Box of quantity of favorites "XX Favorites";
6. Diary list content area;
7. Title "My Favorites";
8. Items of list:
  - (a) Thumbnail;
  - (b) Title;
  - (c) Publisher's username;
  - (d) Datetime;
  - (e) Main text.



Account page (after sign in & my favorites)

### Functional requirements:

1. The username of the currently signed-in user is derived from the string before @ in the signed-in user's email address;
2. The quantity of favorites should be displayed correctly;
3. The My Favorites area shows the user's favorite diary list;
4. Newly favorited diary items are displayed at the top of the favorites list;
5. Each diary item can display up to 4 lines of text. Text exceeding 4 lines is truncated and displayed as "...";
6. Clicking the "Sign out" button signs out the current user, and the page switches to the Account page (before sign in).



## Instructions to the Competitor

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

## Web API documents

[USER] SIGN IN OR SIGN UP					
No.	1	Method	POST	URL	http://{{BASE_URL}}:{{SERVICE_PORT}}/api/users/signin
Request Body:			Type	Application/Json	
Key		Value (Example)		Description	
userEmailAddress		username@email.com		Vaild Email Address	
userPassword		123abc		No less than six characters and include English letters and numbers	
<pre>{   "userEmailAddress" : "username@email.com",   "userPassword" : "123abc " }</pre>					
Successful Response Body (Example):				Type:	Application/Json
<pre>{   "msg": "Sign in successful",   "data": {     "auth_token": "XHJ0VF1GJXHQMTP2PVWP6EXZS"   } }</pre>					

[DIARY] GET ALL DIARIES					
No.	2	Method	GET	URL	http://{{BASE_URL}}:{{SERVICE_PORT}}/api/diary
Successful Response Body (Example):				Type:	Application/Json
<pre>{   "msg": "Success",   "data": [     {       "diary_id": "BA23617D-42DA-8C4A-F569-C1915B9B55B1",       "diary_title": "Brushstrokes of Time",       "diary_main_text": "resources/Musee d Orsay/d1.json",       "diary_upload_datetime": "2023-03-01 12:30:30",       "diary_image": "resources/Musee d Orsay/1.jpg",       "diary_upload_username": "Zachary Butler"     },     {       "diary_id": "17B9C94E-F829-6088-FA92-9A07EEA00A8E",       "diary_title": "Journey Through the Splendor of French Art",       "diary_main_text": "resources/Musee d Orsay/d2.json",       "diary_upload_datetime": "2023-04-25 09:50:30",       "diary_image": "resources/Musee d Orsay/3.jpg",       "diary_upload_username": "Jake Clarke"     }   ] }</pre>					

[FAVORITE] INSERT NEW FAVORITE					
No.	3	Method	PUT	URL	http://{{BASE_URL}}:{{SERVICE_PORT}}/api/diary/collection
Request Headers:					
Key		Value (Example)		Description	
auth_token		VS0Z7OR9ZTGYQTTJ4A5RCR456		The token which was responded after successfully sign in	
Request Body:			Type	Form-data	
Key		Value (Example)		Description	
diary_id		BA23617D-42DA-8C4A-F569-C1915B9B55B1		The ID of diary	
Successful Response Body (Example):				Type:	Application/Json
{  "msg": "Success",					

```

    "data": {
      "diary_id": "BA23617D-42DA-8C4A-F569-C1915B9B55B1",
      "favorite_datetime": "2024-06-20 02:44:10"
    }
  }
}

```

#### [FAVORITE] GET MY ALLFAVORITES

No.	4	Method	GET	URL	http://{{BASE_URL}}:{{SERVICE_PORT}}/api/diary/collection
Request Headers:					
Key		Value (Example)		Description	
auth_token		VS0Z7OR9ZTGYQTTJ4A5RCR4S6		The token which is responded after sign in successfully	
Successful Response Body (Example):				Type	Application/Json
<pre>{   "msg": "Success",   "data": [     {       "diary_id": "BA23617D-42DA-8C4A-F569-C1915B9B55B1",       "favorite_datetime": "2024-09-10 02:44:10"     },     {       "diary_id": "17B9C94E-F829-6088-FA92-9A07EEA00A8E",       "favorite_datetime": "2024-09-10 02:44:10"     }   ] }</pre>					

#### [RESOURCES] GET RESOURCE FILE

No.	5	Method	GET	URL	http://{{BASE_URL}}:{{SERVICE_PORT}}/api/{{path}}
<b>Path Parameter:</b>					
Key	Value (Example)		Description		
Relative path	resources/Musee d Orsay/d1.json		The relative path of resource files which the diary's image or diary main text		

#### [RESOURCES] VISIT USER AGREEMENT (WEBSITE)

No.	6	URL	http://{{BASE_URL}}:{{SERVICE_PORT}}/api/user-agreement
-----	---	-----	---