通用

一張含有 螢幕擷取畫面, 圖形, 行, 電子藍 的圖片

自動產生的描述

**開發指南**

重韻技術開發部 第一小隊

第一版

# 目錄

[目錄 2](#_Toc158971904)

[宗旨 3](#_Toc158971905)

[常規流程 4](#_Toc158971906)

[設計師 4](#_Toc158971907)

[產品經理 4](#_Toc158971908)

[技術負任人 4](#_Toc158971909)

[UI 開發 5](#_Toc158971910)

[編程語言 6](#_Toc158971911)

[主要 6](#_Toc158971912)

[副要 6](#_Toc158971913)

[框架 8](#_Toc158971914)

[主要 8](#_Toc158971915)

[副要 8](#_Toc158971916)

[組件庫 9](#_Toc158971917)

[主要 9](#_Toc158971918)

[副要 9](#_Toc158971919)

[後端 10](#_Toc158971920)

[API 格式 11](#_Toc158971921)

[包管理 13](#_Toc158971922)

[交付 14](#_Toc158971923)

本指南為 AnsonCar 為管理自己之用，其他人非必要跟隨。

## 宗旨

通用性

複用性

1. 統一
2. 簡單
3. 快速

* 所有項目要統一開發語言，框架，組件庫，增加代碼複用性。
* 所有文字說明需簡單，不要用複雜眼字。
* 所有項目需拆分不同小任務，以日為單位，任務不論完成與否，都需當天交付，再作安排。

注：盡量使用docker開啟服務

## 常規流程

1. 提出概念（idea）
2. ⁠列出需求
3. ⁠設計架構
4. ⁠定義功能
5. ⁠UI設計
6. ⁠開發
7. ⁠QA測試
8. ⁠上架

### 設計師

1. 交互設計
2. 視覺設計
3. 設計評審
4. 交付

### 產品經理

1. 需求定義
2. 需求評審
3. 綫框原型
4. 交互原型

### 技術負任人

1. 需求評審
2. 技術評審
3. 設計評審
4. 頁面開發

## UI 開發

* Adobe XD
* Figma

## UI尺寸

按钮的高度可以不同

按钮可以使全屏宽度

最好的字体大小16px

字体不需要随着按钮的大小而变化

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型, 設計 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 軟體, 網站 的圖片

自動產生的描述

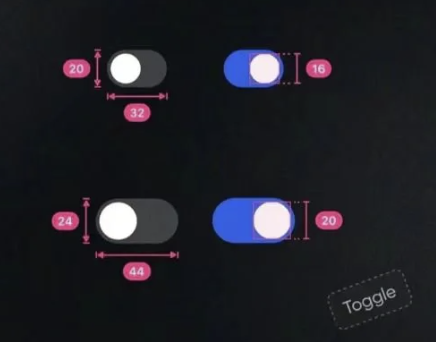


一張含有 文字, 螢幕擷取畫面, 字型, 標誌 的圖片

自動產生的描述

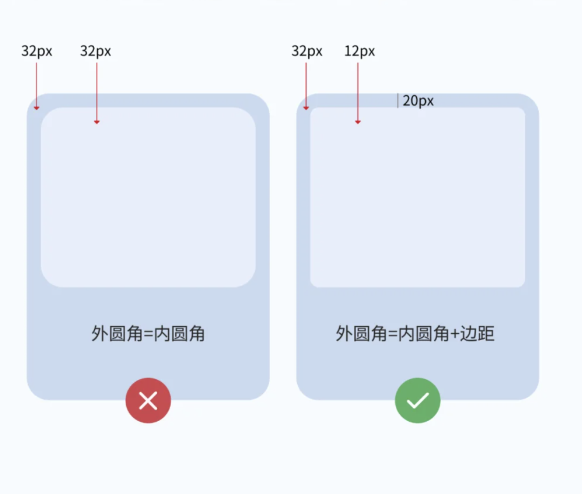






一張含有 文字, 螢幕擷取畫面, 字型, 設計 的圖片

自動產生的描述



* Button

## 編程語言

### 主要

* Python
  + 版本 3.9 以上，不可使用最新版本
  + 管理器 pyenv
* Javascript
  + 版本Node 20.7.0. 以上，不可使用最新版本
  + 管理器 nvm

### 副要

* Golang
* Rust
* C
* C++
* Dart (只用於Flutter)
* R

注：可自行選擇管理器

## 框架

### 主要

* SvelteKit
* Flask
* Fastapi
* Tailwind css

### 副要

* Gin
* Flutter

注：生曰：邊個用React，我同佢死過。

注：只可以使用 Tailwind Css。

注：非特殊/個別情況，只可使用以上框架。

## 組件庫

### 主要

* Daisy UI

### 副要

* AT（開發中）
* TP（開發中）

注：非特殊/個別情況，只可以使用Tailwind Css Component

注：制作組件，只可以使用 Web Component或 Svelte，以及用vite打包。

## 後端

* Appwrite
* Supabase
* Mongodb

## API 格式

鼓勵使用REST / RESTFUL API 格式，

但可使用其他格式。

所有項目文檔需包含以下表格：

Name API

|  |  |  |  |
| --- | --- | --- | --- |
|  | 根URL |  | |
| ID | URL | HTTP方法 | 用途 |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |

|  |  |
| --- | --- |
| ID |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |

注：可加HTTP 狀態碼

例子：

Book API

|  |  |  |  |
| --- | --- | --- | --- |
|  | 根URL | https://example.com/api/ | |
| ID | URL | HTTP方法 | 用途 |
| 1 | books | GET | 取得所有書籍的資訊 |
| 2 | books/{book\_id} | GET | 取得特定書籍的資訊 |
| 3 | books/{book\_id} | POST | 新增一本書籍 |
| 4 | books | PUT |  |
| 5 | Books/{book\_id} | DELETE | 更新特定書籍的資訊 |

傳遞格式

|  |  |  |
| --- | --- | --- |
| ID | request | response |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |

## 包管理

記錄所有使用到的包（不計算依賴包）：

|  |  |  |  |
| --- | --- | --- | --- |
| 名稱 | 版本 | 平台 | 功用 |
|  |  |  |  |
|  |  |  |  |

注：非必須記錄的

## 交付

必須包括：

* 源碼
* 說明文檔
* docker-compose

文件格式為：

[project\_name]\_[version]

注：python pip 沒有requirements.txt 文件，必須拒收

## 測試

所以 測試用的，都建議加上test字眼，例如

Domain:

test\_projectname.star.com