

Design Specification of online ordering platform		
Doc #	Version: 2024	Page 1 / 22

TABLE OF CONTENTS

1	Introduction	1
1.1	Document conventions	1
2	Software architecture overview	2
3	Software design specification	3
3.1	Frontend	3
3.1.1	Component design description	3
3.1.2	Workflows and algorithms	4
3.1.4	Component secure design description	10
3.1.6	Software requirements mapping	11
3.2	Backend	11
3.3	Database	19
4	Responsibilities table	21

1 Introduction

線上點單平台的出現，徹底改變了餐飲業中商家與顧客的互動方式。平台透過簡化點單流程，為顧客提供便利且易用的體驗，同時幫助商家更有效率地管理營運。

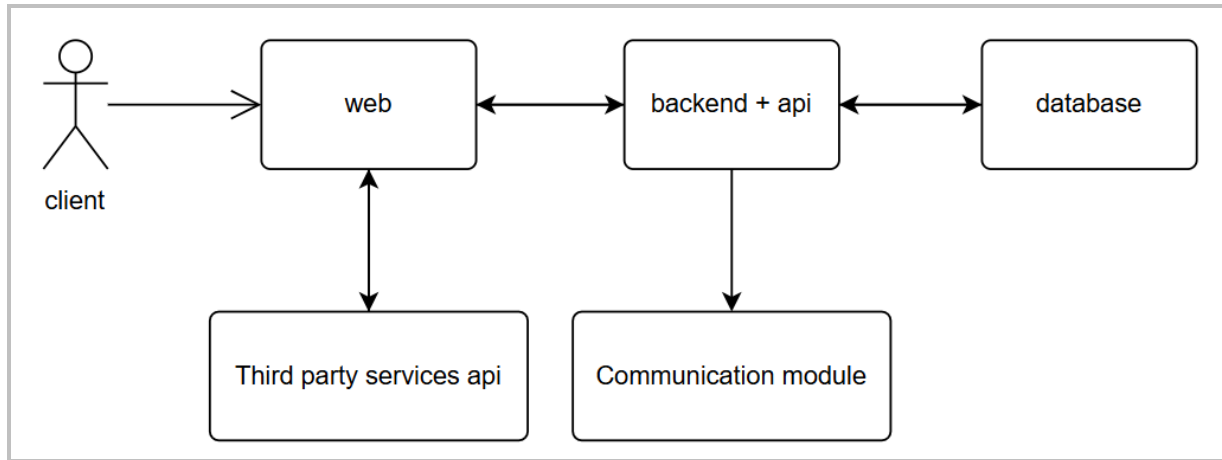
我們的線上點單平台旨在為顧客與商家之間搭建一座無縫的橋樑，提供流暢且高效的點餐流程。此系統專為現代消費者而設計，滿足他們對便利性、速度和靈活性的需求。平台功能包括即時菜單更新、安全的支付處理和訂單追蹤，旨在提升使用者滿意度，並改善商家營運效率。

本系統適用於所有類型的餐廳商家，並提供可自訂的功能以符合不同商家的需求。平台支援桌面和行動裝置，確保跨設備的便利性，為終端用戶提供穩定的使用體驗。

1.1 Document conventions

本文件統一採用Cambria字體撰寫，標題字型大小為12px，子標題及內文皆為11px，大小標題皆保持粗體。

2 Software architecture overview



1. 前端

- 技術: 採用 Nuxt3 + Vue3.js 框架開發, 確保響應迅速且互動性強的使用者介面。
- 元件:
 - 使用者介面: 直觀的設計, 讓顧客瀏覽菜單、選擇商品並完成訂單。
 - 即時更新: 透過 WebSockets 或輪詢機制, 實現訂單狀態與菜單更新的即時同步。
 - 跨平台相容性: 響應式設計, 支援網頁瀏覽器和行動裝置

2. 後端

- 技術: 基於穩定的伺服器框架golang, 提供高效的請求處理與資料處理功能。
- 核心功能:
 - 訂單管理: 處理訂單的建立、更新與追蹤。
 - 菜單管理: 動態更新菜單、檢查商品庫存, 並支援自訂選項。
 - 認證與授權: 提供安全的使用者與商家登入機制。
 - **API**: 使用 RESTful API, 促進前後端之間的溝通。

3. 資料庫

- 技術: 使用NoSQL資料庫firestroe, 它提供更靈活的數據儲存和檢索方式, 通常用於需要高性能、可擴展性或處理非結構化資料的應用場景。
- 實體:
 - 使用者資料(顧客與商家)
 - 菜單與商品
 - 訂單與支付記錄
 - 交易日誌與分析數據

4. 第三方服務api

- 支付api: 支援第三方支付工具(如 Stripe、PayPal 或本地支付解決方案), 確保交易安全便捷。

- 地圖定位api:支援google map 定位使用者及餐廳所在位置，整合導航功能以保障使用者體驗。

5. 通訊模組

- 通知:透過電子郵件、簡訊或推送通知，即時提醒使用者和餐廳訂單狀態。

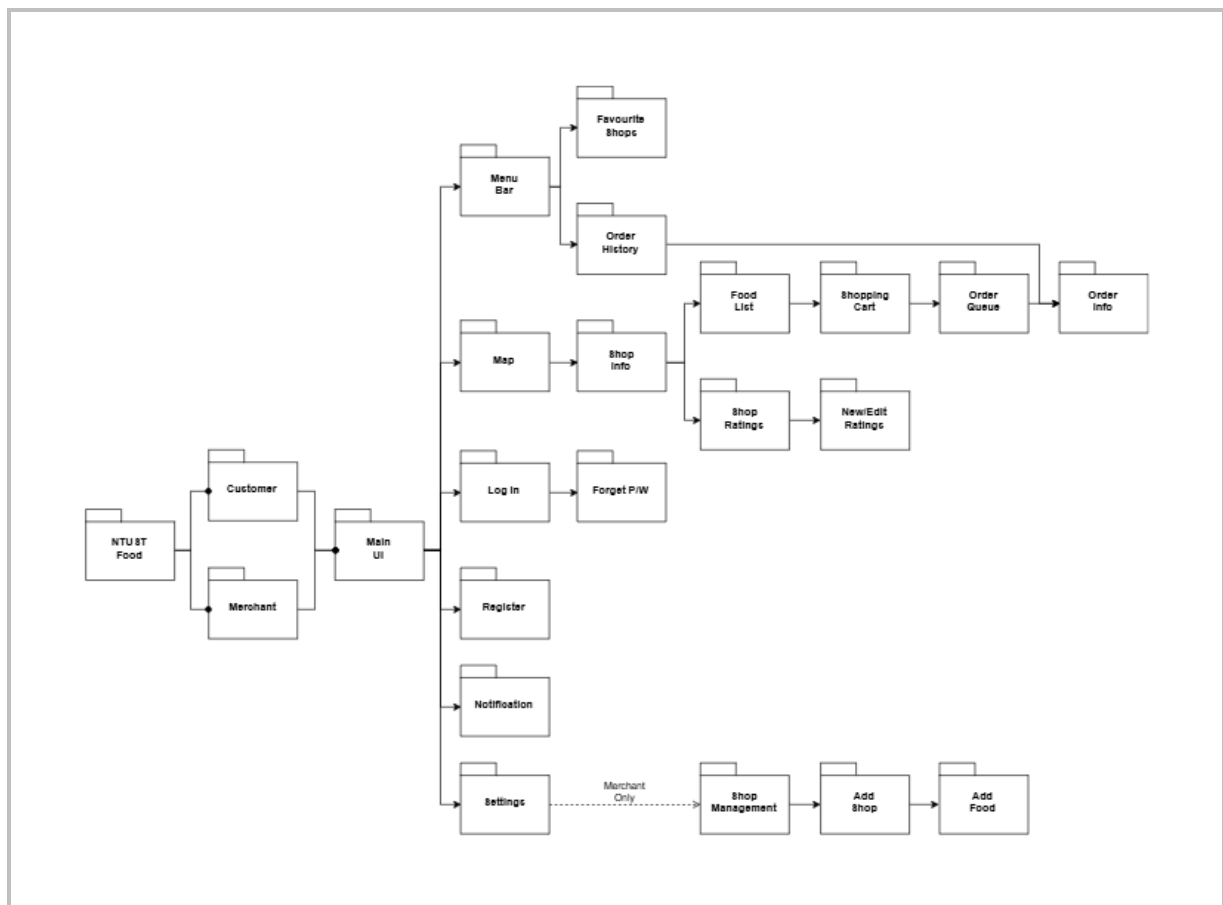
6. 部署與主機

- 雲端架構:部署於 AWS 平台，確保擴展性與穩定性。
- 內容傳遞網路 (CDN):加速內容傳遞並降低延遲。

3 Software design specification

3.1 Frontend

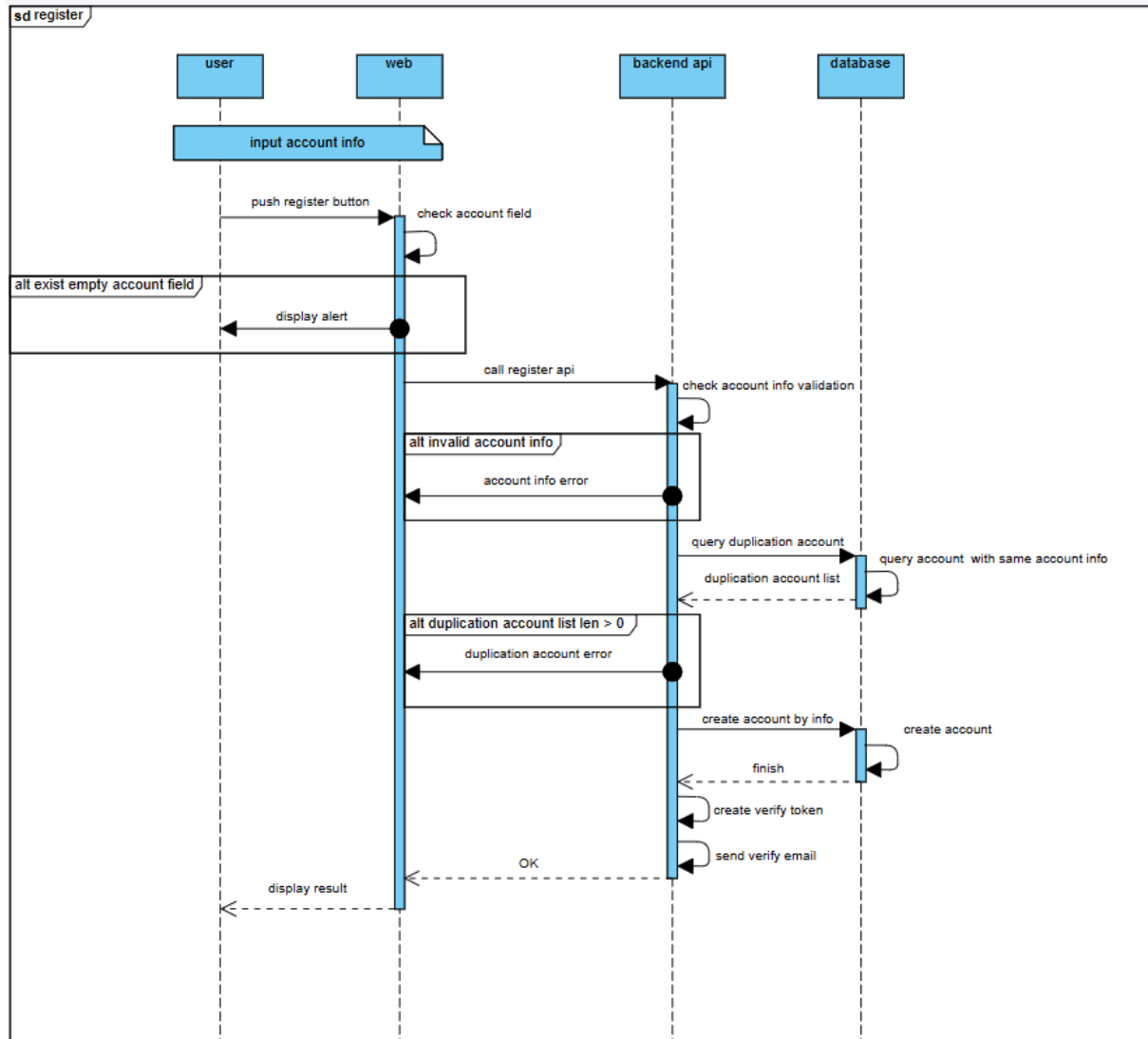
3.1.1 Component design description



上圖展示了前端UI畫面的導向關係，箭頭左至右代表左方UI畫面中將有元件可讓使用者交互並重新導向至右方畫面。

3.1.2 Workflows and algorithms

Register



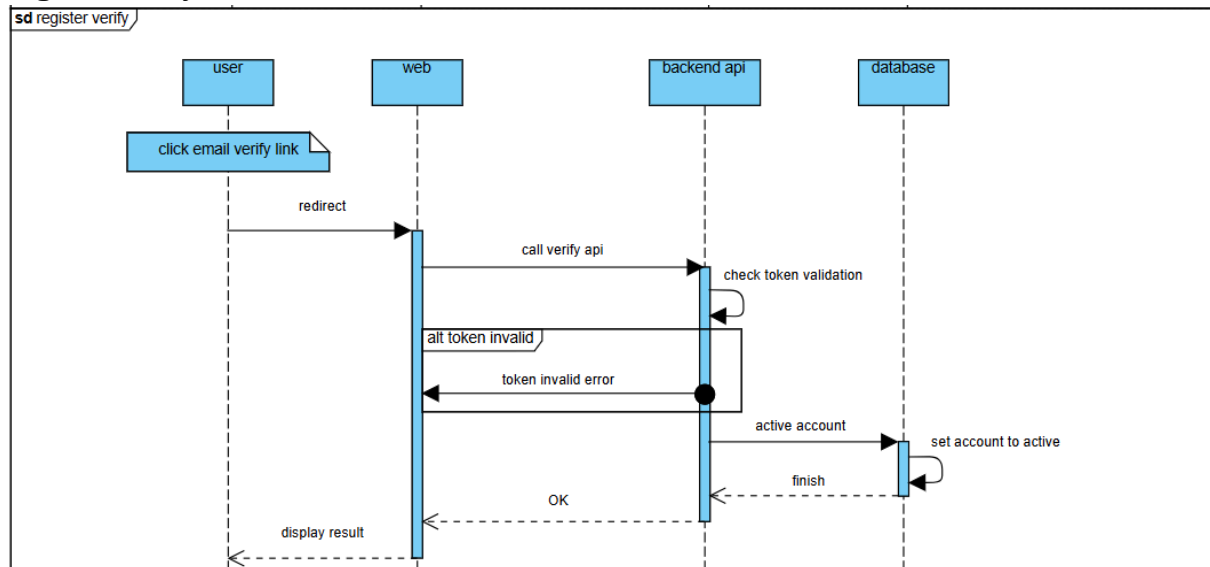
Design Specification of online ordering platform

Doc #

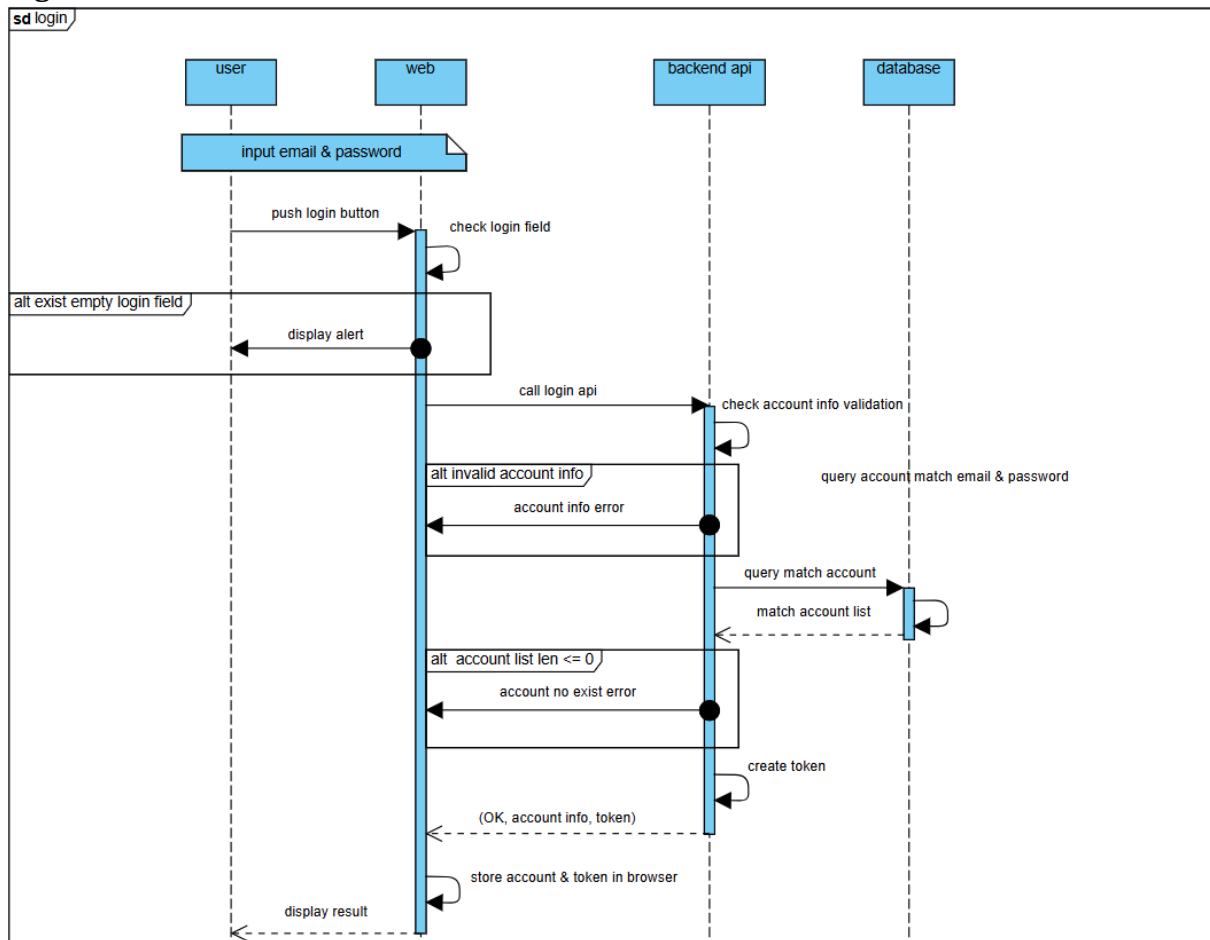
Version: 2024

Page 5 / 22

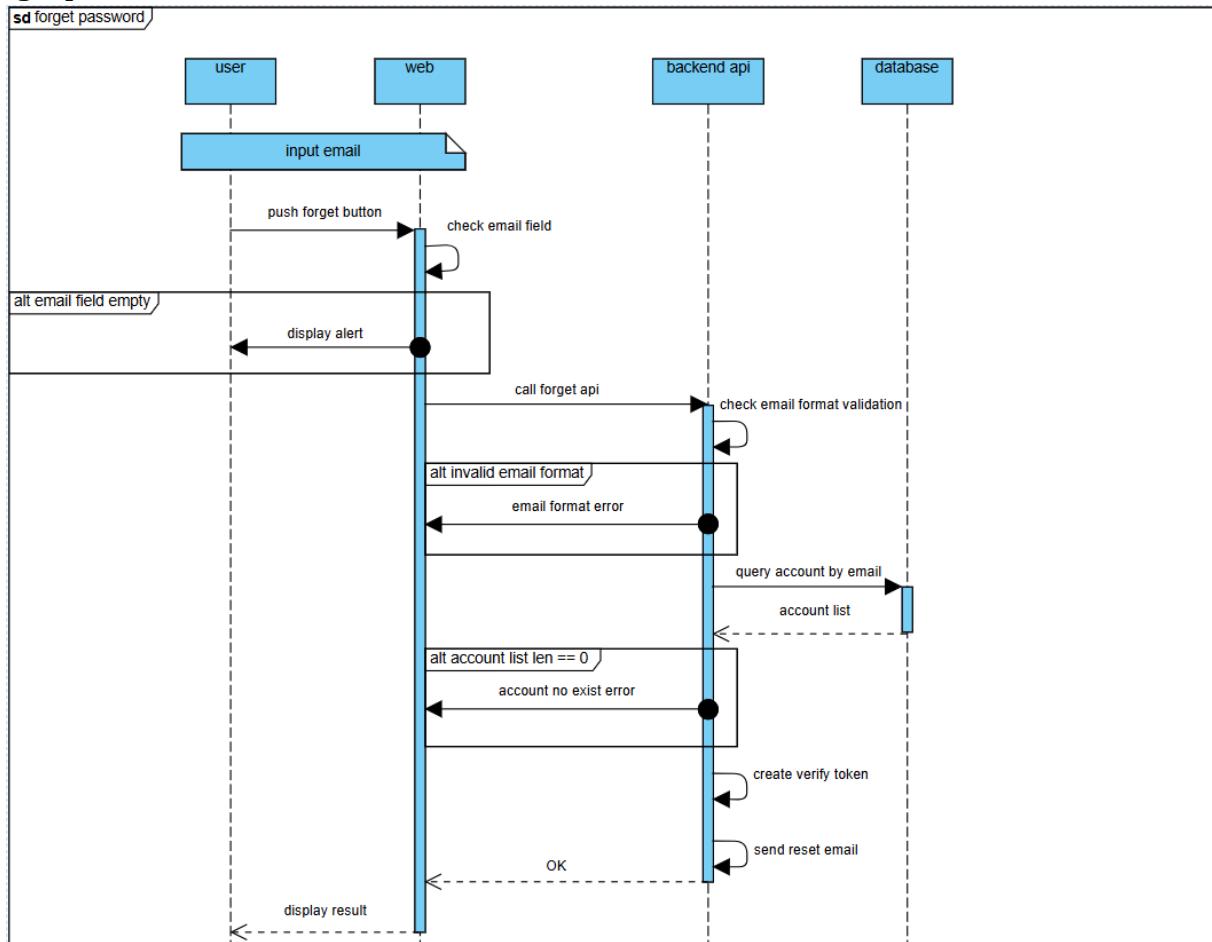
Register verify



Login



Forget password



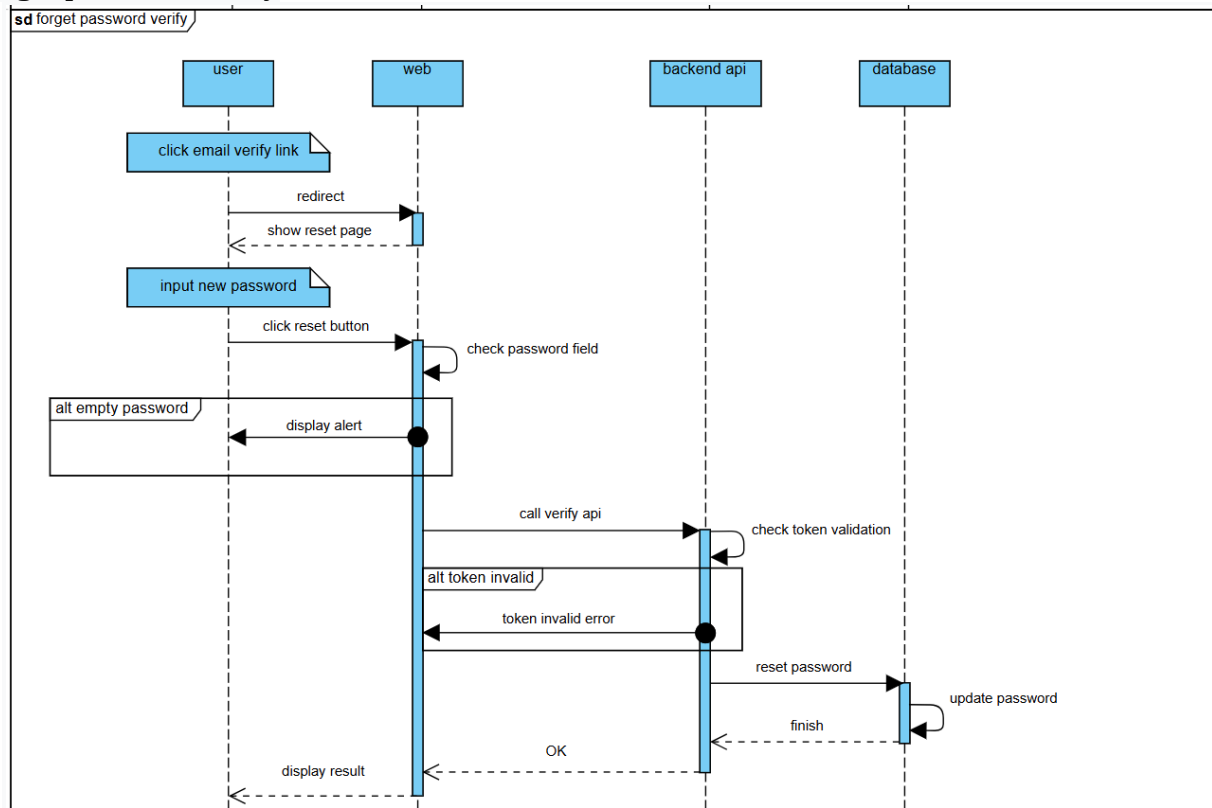
Design Specification of online ordering platform

Doc #

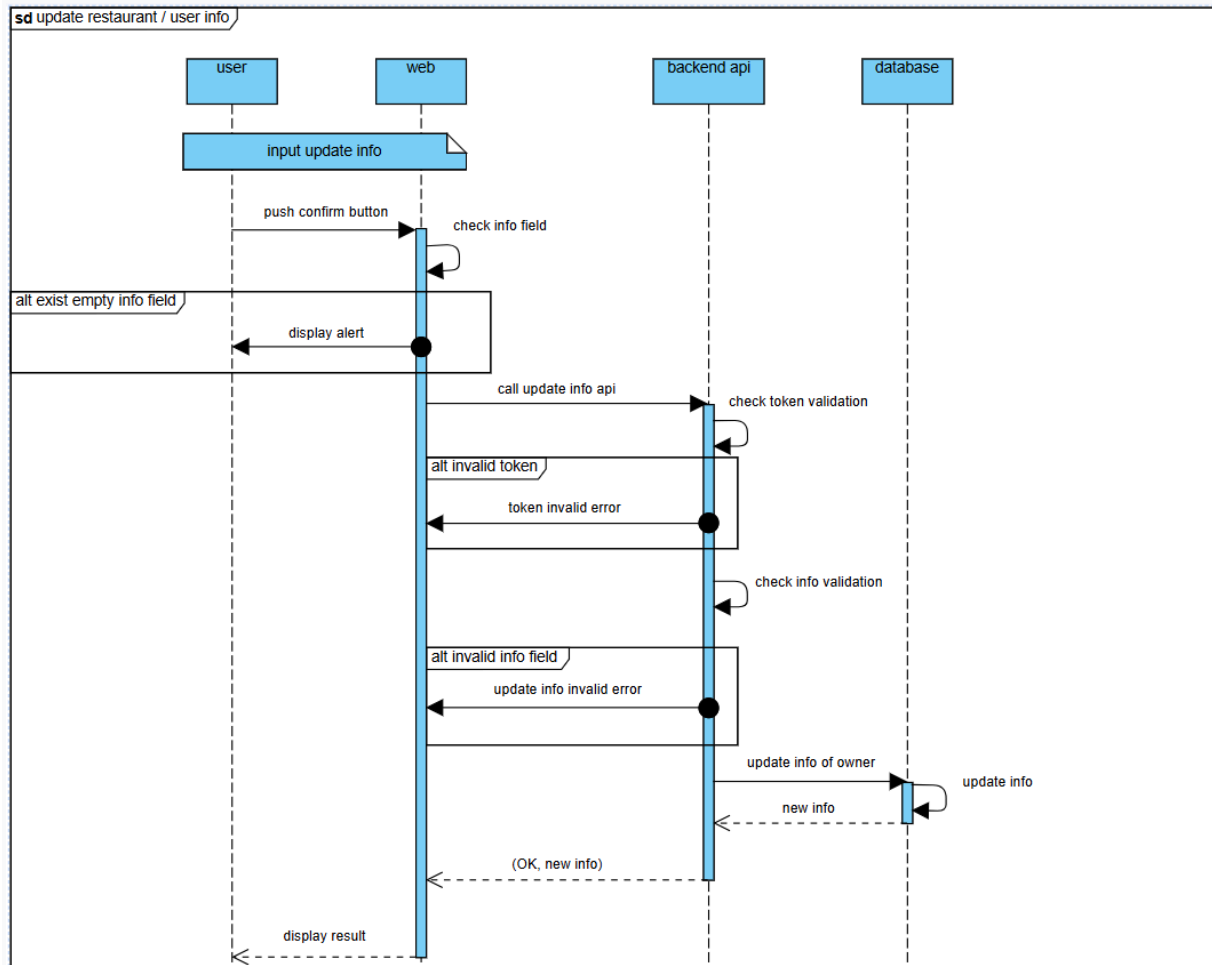
Version: 2024

Page 7 / 22

Forget password verify



Update restaurant & user info



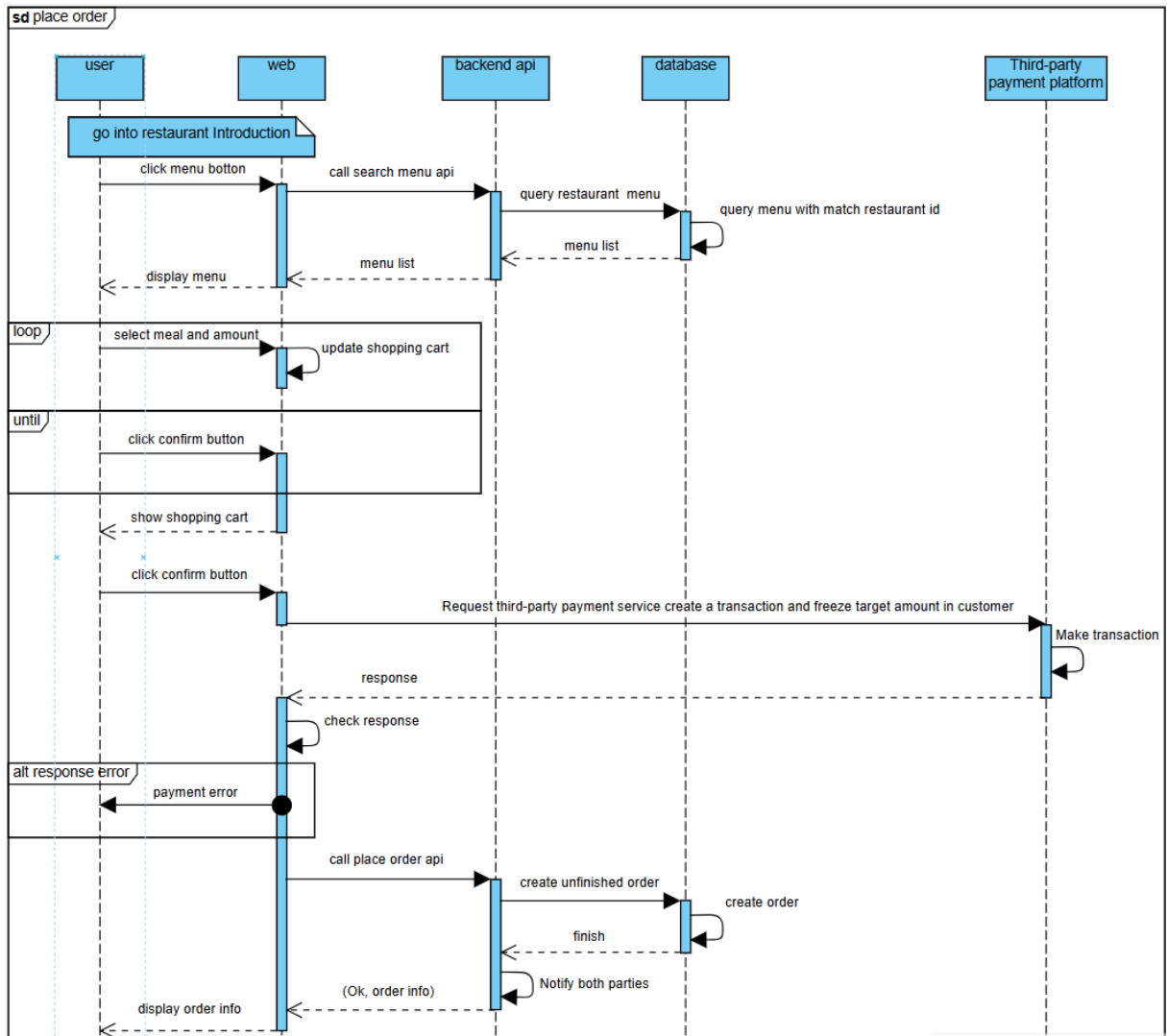
Design Specification of online ordering platform

Doc #

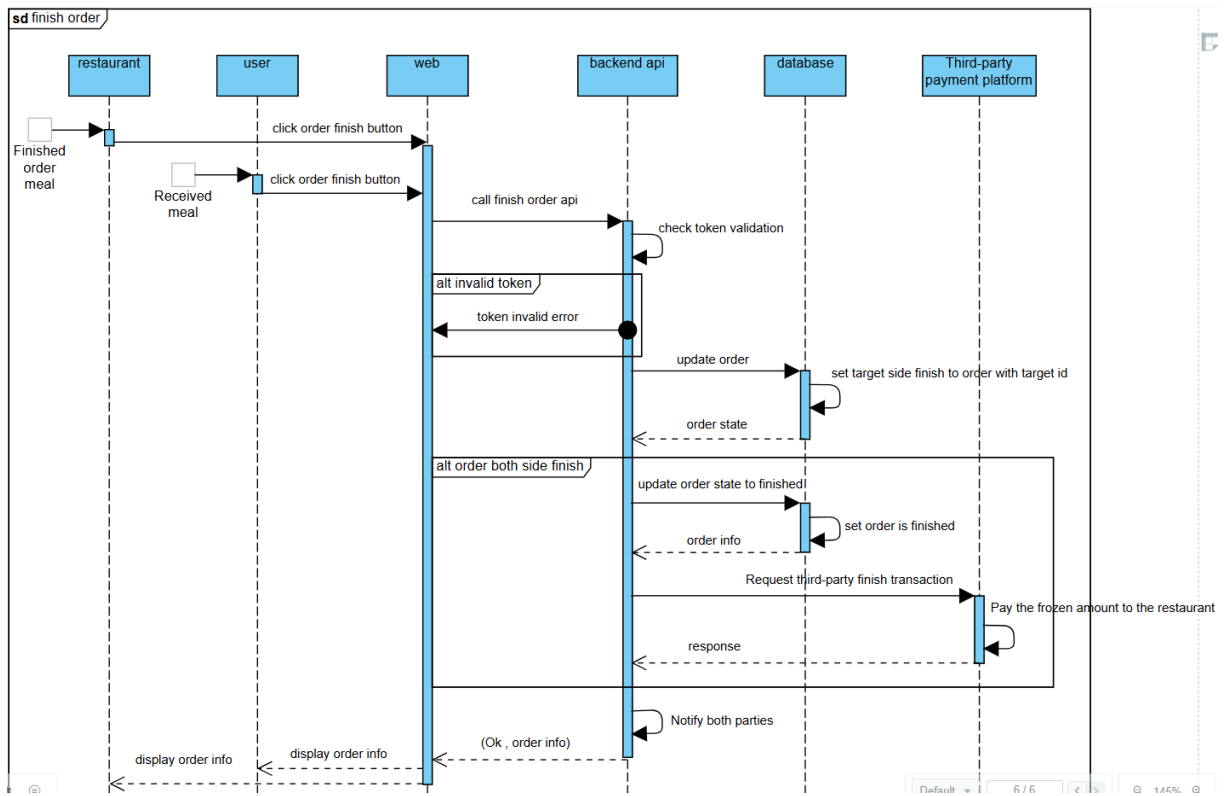
Version: 2024

Page 9 / 22

Place order



Finish order



3.1.3 Component secure design description

前端網頁共有三種權限設定：

- 未登入：只能查詢餐廳。
- User：允許查詢餐廳、下訂單、查詢訂單狀態、設定帳號資訊。
- Restaurant：允許查詢餐廳、查詢訂單狀態、設定餐廳資訊、設定菜單資訊。

有權限限制之頁面會檢查登入帳號的權限設定，未符合權限要求則會拒絕其存取並跳轉至主頁面。

Design Specification of online ordering platform		
Doc #	Version: 2024	Page 11 / 22

3.1.4 Software requirements mapping

需求	UI畫面	說明
菜單瀏覽	SEFE-119: 餐廳資訊畫面 SEFE-131: 菜單瀏覽畫面	菜單應包含所有品項的詳細資訊，並且按分類排列，讓顧客快速找到所需的餐點。
餐點加選	SEFE-120: 點菜畫面	顧客應該能夠在選擇菜品後指定加料、配料等偏好，並看到最終價格的變動。
即時價格更新	SEFE-133: 購物車畫面	每次新增或刪除餐點，總金額會自動更新，包括運費。
訂單確認與支付	SEFE-120: 點菜畫面 SEFE-133: 購物車畫面 SEFE-121: 支付畫面	訂單確認頁應顯示所有點餐項目、數量、總價、稅費及地址。顧客能夠選擇支付方式並完成付款。
訂單追蹤	SEFE-134: 訂單查詢畫面	系統應該提供訂單狀態更新，並在訂單完成送達時提醒顧客。
菜單管理	SEFE-116: 餐廳資訊管理畫面 SEFE-117: 菜單設定畫面	後台應提供簡單的編輯介面，允許管理員更新餐點、圖片、描述及價格，且能立即在前端顯示更新。

3.2 Backend

使用者功能:

1. 忘記密碼

- 網址: /user/forget
- 方法: POST
- 功能: 用戶通過提供 email 信箱來進行忘記密碼的操作。
- 參數:
 - email (類型: string, 位置: query, 描述: 用戶的 email 信箱)
- 回應:
 - 狀態碼: 200
 - 回應內容包含: code 和 message (成功或失敗的狀態碼及消息)

2. 獲取用戶信息

- 網址: /user/info

- 方法:GET
 - 功能:獲取當前用戶的詳細信息。
 - 參數:無
 - 回應:
 - 狀態碼:200
 - 回應內容包含:code, message, 以及 user(用戶的詳細信息)
-

3. 登入帳號

- 網址:/user/login
 - 方法:POST
 - 功能:用戶提供 email 和密碼進行登入。
 - 參數:
 - user(類型:object, 位置:body, 包含以下屬性)
 - email(類型:string, 用戶的 email)
 - password(類型:string, 用戶的密碼)
 - 回應:
 - 狀態碼:200
 - 回應內容包含:code, message, 和 token(用於後續認證的令牌)
-

4. 註冊新用戶

- 網址:/user/register
 - 方法:POST
 - 功能:用戶通過提供 email、姓名、密碼、電話和角色進行註冊。
 - 參數:
 - user(類型:object, 位置:body, 包含以下屬性)
 - email(類型:string, 用戶的 email)
 - name(類型:string, 用戶的姓名)
 - password(類型:string, 用戶的密碼)
 - phone(類型:string, 用戶的電話)
 - roles(類型:string, 用戶的角色)
 - 回應:
 - 狀態碼:200
 - 回應內容包含:code 和 message(註冊結果的狀態碼和消息)
-

5. 驗證帳號

- 網址:/user/verify
- 方法:POST
- 功能:用戶提供驗證令牌來驗證帳號。
- 參數:
 - token(類型:string, 位置:query, 描述:用戶的驗證令牌)

- 回應:
 - 狀態碼:200
 - 回應內容包含:code 和 message (驗證結果的狀態碼和消息)

餐廳功能

1. 新增最愛餐廳

- 網址:/restaurant/addFavoriteRestaurant
- 方法:POST
- 功能:將指定餐廳新增到用戶的最愛餐廳列表中。
- 參數:
 - token(類型:string, 位置:body, 用戶的認證 token)
 - restaurant_id(類型:string, 位置:body, 餐廳的 ID)
- 回應:
 - 狀態碼:200
 - 回應內容包含:code 和 message (成功或失敗的狀態碼及消息)

2. 新增餐廳

- 網址:/restaurant/addRestaurant
- 方法:POST
- 功能:新增一個餐廳。
- 參數:
 - token(類型:string, 位置:body, 用戶的認證 token)
 - restaurant_id(類型:string, 位置:body, 餐廳的 ID)
 - name(類型:string, 位置:body, 餐廳名稱)
 - business_hours(類型:string, 位置:body, 餐廳營業時間)
 - address(類型:string, 位置:body, 餐廳地址)
 - phone(類型:string, 位置:body, 餐廳電話)
- 回應:
 - 狀態碼:200
 - 回應內容包含:code 和 message (成功或失敗的狀態碼及消息)

3. 刪除餐廳

- 網址:/restaurant/deleteRestaurant
- 方法:DELETE
- 功能:刪除指定的餐廳。
- 參數:
 - token(類型:string, 位置:body, 用戶的認證 token)

- `restaurant_id`(類型:string, 位置:body, 餐廳的 ID)
 - 回應:
 - 狀態碼:200
 - 回應內容包含:`code` 和 `message`(成功或失敗的狀態碼及消息)
-

4. 查詢最愛餐廳列表

- 網址:`/restaurant/findFavoriteRestaurantList`
 - 方法:GET
 - 功能:獲取用戶的最愛餐廳列表。
 - 參數:
 - `token`(類型:string, 位置:body, 用戶的認證 token)
 - 回應:
 - 狀態碼:200
 - 回應內容包含:`code`, `message`, 和 `data`(餐廳列表)
-

5. 透過ID查詢餐廳

- 網址:`/restaurant/findRestaurantByID`
 - 方法:GET
 - 功能:根據餐廳 ID 查詢餐廳詳細信息。
 - 參數:
 - `id`(類型:string, 位置:query, 餐廳的 ID)
 - 回應:
 - 狀態碼:200
 - 回應內容包含:`code`, `message`, 和 `data`(餐廳信息)
-

6. 查詢餐廳列表

- 網址:`/restaurant/findRestaurantList`
 - 方法:GET
 - 功能:根據餐廳名稱查詢餐廳列表。
 - 參數:
 - `name`(類型:string, 位置:query, 餐廳名稱)
 - 回應:
 - 狀態碼:200
 - 回應內容包含:`code`, `message`, 和 `data`(餐廳列表)
-

7. 查詢餐廳評分

- 網址:`/restaurant/findRestaurantRating`

- 方法:GET
- 功能:查詢指定餐廳的評分。
- 參數:
 - id(類型:string, 位置:query, 餐廳的 ID)
- 回應:
 - 狀態碼:200
 - 回應內容包含:code, message, 和 data(餐廳評分)

8. 刪除最愛餐廳

- 網址:/restaurant/removeFavoriteRestaurant
- 方法:DELETE
- 功能:將餐廳從最愛餐廳列表中移除。
- 參數:
 - token(類型:string, 位置:body, 用戶的認證 token)
 - restaurant_id(類型:string, 位置:body, 餐廳的 ID)
- 回應:
 - 狀態碼:200
 - 回應內容包含:code 和 message(成功或失敗的狀態碼及消息)

9. 修改餐廳資訊

- 網址:/restaurant/updateRestaurantInformation
- 方法:POST
- 功能:修改指定餐廳的資料。
- 參數:
 - token(類型:string, 位置:body, 用戶的認證 token)
 - restaurant_id(類型:string, 位置:body, 餐廳的 ID)
 - name(類型:string, 位置:body, 餐廳名稱)
 - business_hours(類型:string, 位置:body, 餐廳營業時間)
 - address(類型:string, 位置:body, 餐廳地址)
 - phone(類型:string, 位置:body, 餐廳電話)
- 回應:
 - 狀態碼:200
 - 回應內容包含:code 和 message(成功或失敗的狀態碼及消息)

評價功能

1. 評價添加/修改

網址:/rate/addRate

方法:POST

功能: 新增或修改餐廳的評價。

參數:

- **rate**(類型:object, 位置:body, 包含以下欄位):
 - **comment**(類型:string, 餐廳的評價內容)
 - **rate**(類型:number, 餐廳的評分)
 - **restaurant_id**(類型:string, 餐廳的 ID)
 - **token**(類型:string, 用戶的認證 token)
- 回應:
- 狀態碼: 200
- 回應內容包含: code 和 message (成功或失敗的狀態碼及消息)

2. 刪除評價

網址: /rate/deleteRate

方法: DELETE

功能: 刪除指定的餐廳評價。

參數:

- **rate**(類型:object, 位置:body, 包含以下欄位):
 - **rate_id**(類型:string, 評價的 ID)
 - 回應:
 - 狀態碼: 200
 - 回應內容包含: code 和 message (成功或失敗的狀態碼及消息)
-

訂單及食物功能

1. 新增/修改菜單

網址: /food/addMenu

方法: POST

功能: 新增或修改餐廳的菜單。

參數:

- token(類型:string, 位置:body, 用戶的認證 token)
 - restaurant_id(類型:string, 位置:body, 餐廳的 ID)
 - id(類型:string, 位置:body, 食物的 ID, 必填)
回應:
 - 狀態碼: 200
 - 回應內容包含: code 和 message (成功或失敗的狀態碼及消息)
-

2. 刪除食物

網址: /menu/deleteFood

方法: DELETE

功能: 刪除指定的食物。

參數:

- token(類型:string, 位置:body, 用戶的認證 token)
 - food_name(類型:string, 位置:body, 食物名稱)
回應:
 - 狀態碼: 200
 - 回應內容包含: code 和 message (成功或失敗的狀態碼及消息)
-

3. 新增訂單

網址: /order/addOrder

方法: POST

功能: 新增一筆訂單。

參數:

- order(類型:object, 位置:body, 包含以下欄位):
 - order_list(類型:object, 食物名稱及數量)
 - restaurant_id(類型:string, 餐廳的 ID)回應:
 - 狀態碼: 200
 - 回應內容包含: code 和 message (成功或失敗的狀態碼及消息)
-

4. 結束訂單

網址: /order/endOrder

方法: POST

功能: 結束指定的訂單。

參數:

- order (類型: object, 位置: body, 包含以下欄位):
 - order_id (類型: string, 訂單的 ID)
 - 回應:
 - 狀態碼: 200
 - 回應內容包含: code 和 message (成功或失敗的狀態碼及消息)
-

5. 取得訂單

網址: /order/getOrder

方法: GET

功能: 取得所有訂單資料。

回應:

- 狀態碼: 200
 - 回應內容包含: code、message 和 data (訂單資料)
-

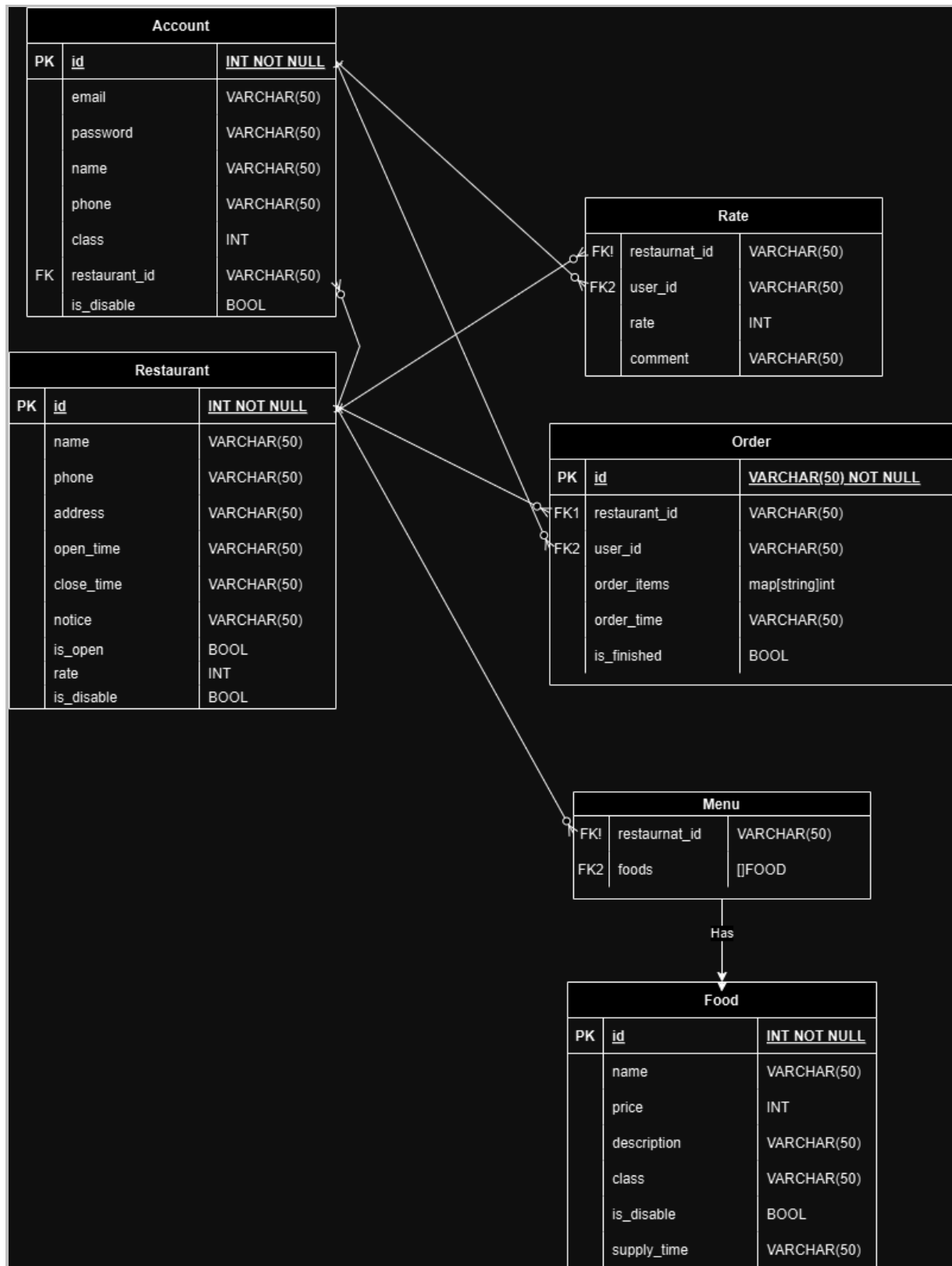
Design Specification of online ordering platform

Doc #

Version: 2024

Page 19 / 22

3.3 Database



Design Specification of online ordering platform		
Doc #	Version: 2024	Page 20 / 22

Account

Account 表用於存儲所有帳戶的相關資訊。此表與 Restaurant、Rate 和 Order 表具有關聯性。

- 屬性：
 - id (PK): 帳戶的唯一識別碼。
 - email: 用戶的電子郵件地址。
 - password: 帳戶的密碼。
 - name: 帳戶名稱。
 - phone: 用戶的聯絡電話號碼。
 - class: 帳戶的分類(如用戶類型)。
 - restaurant_id (FK): 該帳戶所屬餐廳的識別碼。
 - is_disable: 帳戶是否被禁用的標記。

Restaurant

Restaurant 表用於存儲餐廳的相關資訊。此表與 Account、Rate 和 Menu 表具有關聯性。

- 屬性：
 - id (PK): 餐廳的唯一識別碼。
 - name: 餐廳名稱。
 - phone: 餐廳的聯絡電話號碼。
 - address: 餐廳地址。
 - open_time: 營業開始時間。
 - close_time: 營業結束時間。
 - notice: 餐廳的公告訊息。
 - is_open: 餐廳是否營業的標記。
 - rate: 餐廳的平均評分。
 - is_disable: 餐廳是否被禁用的標記。

Rate

Rate 表用於存儲對餐廳的評分與評論。此表與 Restaurant 和 Account 表具有關聯性。

- 屬性：
 - restaurant_id (FK): 被評分餐廳的識別碼。
 - user_id (FK): 評分用戶的識別碼。
 - rate: 評分分數(整數)。
 - comment: 用戶的評論內容。

Order

Design Specification of online ordering platform		
Doc #	Version: 2024	Page 21 / 22

Order 表用於存儲用戶的訂單資訊。此表與 Account 和 Restaurant 表具有關聯性。

- 屬性：
 - id (PK): 訂單的唯一識別碼。
 - restaurant_id (FK): 訂單所屬的餐廳識別碼。
 - user_id (FK): 下訂單用戶的識別碼。
 - order_items: 訂單中的商品與數量(以鍵值對存儲)。
 - order_time: 下訂單的時間。
 - is_finished: 訂單是否完成的標記。

Menu

Menu 表用於存儲餐廳的菜單資訊。此表與 Restaurant 表具有關聯性。

- 屬性：
 - restaurant_id (FK): 菜單所屬餐廳的識別碼。
 - foods: 菜單中的食品列表。

Food

Food 表用於存儲食品的詳細資訊。此表與 Menu 表具有關聯性。

- 屬性：
 - id (PK): 食品的唯一識別碼。
 - name: 食品名稱。
 - price: 食品價格(整數)。
 - description: 食品描述。
 - class: 食品分類。
 - is_disable: 食品是否被禁用的標記。
 - supply_time: 食品的供應時間。

4 Responsibilities table

學號_姓名	工作內容
B11115016_Jin Woojun	Frontend Package diagram
B11115049_黃羿華	SDS Report Frontend Sequence diagram Architecture Overview diagram
B11115055_游騰楓	SDS Report Database Table diagram
B11005121_許菀真	Frontend Sequence diagram

Design Specification of online ordering platform		
Doc #	Version: 2024	Page 22 / 22