A Real Time and Interactive Dashboard for Tourism Industry

**Design Document**

By

Junyu Zhou 592115508

Yawei Li 592115518

Department of Software Engineering,

College of Arts, Media and Technology,

ChiangMai University

Project Advisor



Table of Contents

[1. Document History 5](#_Toc19797158)

[2. Introduction 6](#_Toc19797159)

[2.1 Purpose 6](#_Toc19797160)

[2.2 Intended Audience and Reading Suggestions 6](#_Toc19797161)

[2.3 Project Scope 7](#_Toc19797162)

[2.4 Acronyms 7](#_Toc19797163)

[3. System Architecture 8](#_Toc19797164)

[4. Detail Design (UML Diagram) 10](#_Toc19797165)

[4.1 Component Diagram 10](#_Toc19797166)

[4.2 Class Diagram 12](#_Toc19797168)

[Class diagram overview of whole system 12](#_Toc19797169)

[CD-01 App 13](#_Toc19797170)

[CD-02 Home 14](#_Toc19797171)

[CD-03 Register 16](#_Toc19797172)

[CD-04 Login 18](#_Toc19797173)

[CD-05 AdminLogin 21](#_Toc19797174)

[CD-06 Admin 22](#_Toc19797175)

[CD-07 Accounts 25](#_Toc19797176)

[CD-08 Dashboard 27](#_Toc19797177)

[CD-09 User Profile 29](#_Toc19797178)

[CD-10 EditProfile 32](#_Toc19797179)

[CD-11 Summary 36](#_Toc19797180)

[CD-12 SparkLine 37](#_Toc19797181)

[CD-13 LineChart 38](#_Toc19797182)

[CD-14 BarChart 39](#_Toc19797183)

[CD-15 StackedChart 40](#_Toc19797184)

[Class diagram for web services 42](#_Toc19797185)

[CD-16 DataService 42](#_Toc19797186)

[CD-17 UserService 44](#_Toc19797187)

[Class diagram of Backend 48](#_Toc19797188)

[CD-18 Mongo 48](#_Toc19797189)

[4.3 ER Diagram 52](#_Toc19797190)

[4.4 Sequence Diagram 53](#_Toc19797191)

[5. User Interface Design 57](#_Toc19797192)

[5.1 Web Application UI 57](#_Toc19797193)

[5.1.1 W-UI01-Homepage 57](#_Toc19797194)

[5.1.2 W-UI02-Super admin login 58](#_Toc19797195)

[5.1.3 W-UI03-Register 59](#_Toc19797196)

[5.1.4 W-UI04-Admin Login 60](#_Toc19797197)

[5.1.5 W-UI05-Add accounts 61](#_Toc19797198)

[5.1.5 W-UI06-Dashboard 62](#_Toc19797199)

[5.1.7 W-UI07-User profile page 63](#_Toc19797200)

[5.1.8 W-UI08-Edit username page 64](#_Toc19797201)

[5.1.9 W-UI09-Edit password page 65](#_Toc19797202)

[6. List of Figures 66](#_Toc19797203)

# 1. Document History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| History | Status | Date | Viewable | Editable | Responsible |
| Project-DesignDocument\_v1.docx  Create:  - Document History  - Introduction  - System Architecture  - Detail Design  - User Interface Design | Draft | 17, July 2019 | ZJY, LYW,  AJP | ZJY, LYW | ZJY,  LYW |
| Project-DesignDocument\_v2.docx  Modify:  - Detail Design  - User Interface Design | Final | 22, Aug 2019 | ZJY, LYW,  AJP | ZJY, LYW | ZJY,  LYW |

ZJY = Junyu Zhou

LYW = Yawei Li

AJP = Dr. Pree Thiengburanathum

# 2. Introduction

## 2.1 Purpose

The purpose of the software design document (SDD) is to design the detailed structure of the system according with the software requirement specification (SRS). It intended to help stakeholders in the project understands the detailed design of the system using class diagrams, sequence diagrams, and user interface designs.

## 2.2 Intended Audience and Reading Suggestions

**2.2.1 Intended Audience**

This software requirement specification will benefit to audiences as followed:

**Development Team**

The software requirement specification will help everyone in developer team has the same understanding or overall scope of the project.

The software requirement specification will help the developer to verify and validate the software product that match with the requirements in this document or not.

The software requirement specification will help developers to develop project more efficiency.

The software requirement specification will be required when some changing occur. Developer will know where get effect and know where to change.

**User**

The software requirement specification will help the user to know the process of the software product.

User can check that the requirements in software requirement specification match with their requirement or not.

The software requirement specification can help the user to discuss with developers easier.

## 2.3 Project Scope

A real time and interactive dashboard in tourism industry is a web-based application. It is for decision maker to do the easy statistics of mess data in a real time way. For decision maker to manage and keep tracks all the data.

## 2.4 Acronyms

SRS = Software Requirement Specification SDD Software Design Document

UI = User Interface

DD = Development Diagram

CD= Class Diagram

ER= Entity Relationship Diagram

# 3. System Architecture

(On the next page)

Figure 1: System Architecture

The system consists of clients and server. For the client, there is a part, the web application. The web application that used by users will send and receive data from web-server in JSON format. The server itself store and retrieve data from the database.

We are using pusher components, which could transfer the data two-way duple

**图片包含 屏幕截图

描述已自动生成**

# 4. Detail Design (UML Diagram)

## 4.1 Component Diagram

(On the next page)

## 

Figure : Component Diagram

## 4.2 Class Diagram

## Class diagram overview of whole system

图片包含 文字, 屏幕截图

描述已自动生成

Figure 3: Class Diagram Overview of whole system

## CD-01 App

**图片包含 屏幕截图

描述已自动生成**

Figure 4: App class

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | render () |
| **Parameters** | - |
| **Description** | It is the main class of system |
| **Returns** | <App> |

## CD-02 Home

图片包含 屏幕截图

描述已自动生成

Figure 5: Home class

**Field Summary:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | boolean | isLoginOpen | Check if system displays login interface. |
| 2 | Default | boolean | isRegisterOpen | Check if system displays register interface. |
| 3 | Default | boolean | isAdminOpen | Check if system displays admin interface. |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | showLoginBox () |
| **Parameters** | - |
| **Description** | It is used to show login box. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | render () |
| **Parameters** | - |
| **Description** | It is used to render component |
| **Returns** | <Home> |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | showRegisterBox () |
| **Parameters** | - |
| **Description** | It is used to show register box. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 4 |
| **Name** | showAdminBox () |
| **Parameters** | - |
| **Description** | It is used to show admin box. |
| **Returns** | - |

## CD-03 Register

图片包含 屏幕截图, 文字

描述已自动生成

Figure 6: Register class

**Field Summary:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | String | username | The username of account |
| 2 | Default | String | password | The password of account |
| 3 | Default | array | errors | The array of errors |
| 4 | Default | boolean | isRegistered | The boolean to judge if register |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | showValidationErr () |
| **Parameters** | elm: event, msg: string |
| **Description** | It is used to show validation message |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | clearValidation () |
| **Parameters** | elm: event |
| **Description** | It is used to remove validation message |
| **Returns** | array |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | onUsernameChanged () |
| **Parameters** | e: event |
| **Description** | It is used to listen username change. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 4 |
| **Name** | onPasswordChanged () |
| **Parameters** | e: event |
| **Description** | It is used to listen password change. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 5 |
| **Name** | submitRegister () |
| **Parameters** | e: event |
| **Description** | It is used to submit register form. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 6 |
| **Name** | render () |
| **Parameters** | elm: event, msg: string |
| **Description** | It is used to render component. |
| **Returns** | <RegisterBox> |

## CD-04 Login

图片包含 屏幕截图

描述已自动生成

Figure 7: Login class

**Field Summary:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | String | username | The username of account |
| 2 | Default | String | password | The password of account |
| 3 | Default | array | errors | The array of errors |
| 4 | Default | boolean | isLoggedln | The boolean to judge if login |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | showValidation () |
| **Parameters** | elm: event, msg: string |
| **Description** | It is used to show validation |
| **Returns** |  |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | clearValidationt () |
| **Parameters** | elm: |
| **Description** | It is used to remove validation |
| **Returns** | array |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | onUsernameChanged () |
| **Parameters** | e: event |
| **Description** | It is used to listen username change. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 4 |
| **Name** | onPasswordChanged () |
| **Parameters** | - |
| **Description** | It is used to listen password change. |
| **Returns** | <LoginBox> |

|  |  |
| --- | --- |
| **ID** | 5 |
| **Name** | submitLogin () |
| **Parameters** | e: event |
| **Description** | It is used to submit login request |
| **Returns** |  |

|  |  |
| --- | --- |
| **ID** | 6 |
| **Name** | render () |
| **Parameters** | e: event |
| **Description** | It is used to render component. |
| **Returns** |  |

## CD-05 AdminLogin

图片包含 屏幕截图

描述已自动生成

Figure 8: AdminLogin class

**Field Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | string | username | The username of account |
| 2 | Default | string | password | The password of account |
| 3 | Default | array | errors | The array of errors |
| 4 | Default | boolean | isLoggedln | The boolean to judge if login |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | render () |
| **Parameters** | - |
| **Description** | It is used to render component. |
| **Returns** | <AdminLogin> |

## CD-06 Admin

图片包含 屏幕截图

描述已自动生成

Figure 9: Admin class

**Field Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | boolean | isAccountShow | The boolean to judge if account is showing |
| 2 | Default | boolean | isWaitingListShow | The boolean to judge if waiting list is showing |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | componentDidMount () |
| **Parameters** | - |
| **Description** | It is used to load data from local storage after all components are rendered. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | showDashboard () |
| **Parameters** | - |
| **Description** | It is used to show dashboard. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | showAccount () |
| **Parameters** | - |
| **Description** | It is used to show account. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 4 |
| **Name** | showWaitingList () |
| **Parameters** | - |
| **Description** | It is used to show waiting list. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 5 |
| **Name** | logout () |
| **Parameters** | e: event |
| **Description** | It is used to logout from system. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 6 |
| **Name** | render () |
| **Parameters** | - |
| **Description** | It is used to render component. |
| **Returns** | <Admin> |

## CD-07 Accounts

图片包含 屏幕截图

描述已自动生成



Figure 10: Accounts class

**Field Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | array | accounts | The array of accounts |
| 2 | Default | array | columns | The array of columns |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | componentDidMount () |
| **Parameters** | - |
| **Description** | It is used to load data from local storage after all components are rendered. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | getAll () |
| **Parameters** | - |
| **Description** | It is used to get all accounts |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | onAdd () |
| **Parameters** | newData: userObject |
| **Description** | It is used to listen add change. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 4 |
| **Name** | onDelete () |
| **Parameters** | oldData: userObject |
| **Description** | It is used to listen delete change. |
| **Returns** |  |

|  |  |
| --- | --- |
| **ID** | 5 |
| **Name** | onEdit () |
| **Parameters** | oldData: userObject, newData: userObject |
| **Description** | It is used to listen edit change. |
| **Returns** |  |

|  |  |
| --- | --- |
| **ID** | 6 |
| **Name** | render () |
| **Parameters** | - |
| **Description** | It is used to render component. |
| **Returns** | <Admin> |

## CD-08 Dashboard

图片包含 屏幕截图

描述已自动生成

Figure 11: Dashboard class

**Field Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | string | username | The username of account |
| 2 | Default | string | password | The password of account |
| 3 | Default | string | triral\_time | The trial time of account |
| 4 | Default | boolean | isSummaryOpen | The boolean to judge if summary page is opening. |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | componentDidMount () |
| **Parameters** | - |
| **Description** | It is used to load data from local storage after all components are rendered. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | onBack () |
| **Parameters** | e: event |
| **Description** | It is used to listen back button is clicked. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | showSummary () |
| **Parameters** | - |
| **Description** | It is used to show summary. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 4 |
| **Name** | logout () |
| **Parameters** | event |
| **Description** | It is used to log out and remove data from local storage. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 5 |
| **Name** | render () |
| **Parameters** | - |
| **Description** | It is used to render component. |
| **Returns** | <Dashboard> |

## CD-09 User Profile

图片包含 屏幕截图, 文字

描述已自动生成

Figure 12: User Profile class

**Field Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | string | username | The username of account |
| 2 | Default | string | status | The status of account |
| 3 | Default | string | trial\_time | The trial time of account |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | componentDidMount () |
| **Parameters** | - |
| **Description** | It is used to load data from local storage after all components are rendered. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | getProfile () |
| **Parameters** | - |
| **Description** | It is used to get user profile |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | edit () |
| **Parameters** | e |
| **Description** | It is used to edit account information |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 4 |
| **Name** | dashboard () |
| **Parameters** | e: event |
| **Description** | It is used to display dashboard |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 5 |
| **Name** | logout () |
| **Parameters** | e: event |
| **Description** | It is used to log out from system |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 6 |
| **Name** | render () |
| **Parameters** | - |
| **Description** | It is used to render component. |
| **Returns** | <Profile> |

## CD-10 EditProfile

图片包含 文字

描述已自动生成

Figure 13: Edit Profile class

**Field Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | LocalStorageToken | token | User identify in the local storage |
| 2 | Default | jwt\_decode | decode | It is used to decode |
| 3 | Default | string | username | The username of account |
| 4 | Default | string | status | The status time of accounts |
| 5 | Default | string | trial\_time | The trial time of account |
| 6 | Default | boolean | passwordChangeOpen | It is used to judge if password change |
| 7 | Default | boolean | usernameChangeOpen | It is used to judge if username change |
| 8 | Default | string | password | The password of account |
| 9 | Default | string | newUsername | The new username of account |
| 10 | Default | string | newPassword | The new password of account |
| 11 | Default | array | errors | The array of errors |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | render () |
| **Parameters** | - |
| **Description** | It is used to render component |
| **Returns** | <Edit> |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | usernameHandleClickOpen () |
| **Parameters** | - |
| **Description** | It is used to listen if username handle clicks open |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | usernameHandleCancel () |
| **Parameters** | - |
| **Description** | It is used to listen if username handle clicks cancel |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 4 |
| **Name** | passwordHandleClickOpen () |
| **Parameters** | - |
| **Description** | It is used to listen if password handle clicks open |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 5 |
| **Name** | usernameHandleConfirm () |
| **Parameters** | - |
| **Description** | It is used to listen if username handle confirms |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 6 |
| **Name** | passwordHandleCancel () |
| **Parameters** | - |
| **Description** | It is used to listen if password handle clicks cancel |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 7 |
| **Name** | passwordHandleConfirm () |
| **Parameters** | - |
| **Description** | It is used to listen if password handle confirms |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 8 |
| **Name** | logout () |
| **Parameters** | - |
| **Description** | It is used to logout from system |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 9 |
| **Name** | onUsernameChanged () |
| **Parameters** | e |
| **Description** | It is used to listen username change. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 10 |
| **Name** | onPasswordChanged () |
| **Parameters** | e |
| **Description** | It is used to listen password change. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 11 |
| **Name** | onNewPasswordChanged () |
| **Parameters** | e |
| **Description** | It is used to listen new password change. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 12 |
| **Name** | showValidationErr () |
| **Parameters** | elm, msg |
| **Description** | It is used to show validation errors |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 13 |
| **Name** | clearValidationErr () |
| **Parameters** | elm |
| **Description** | It is used to remove validation errors |
| **Returns** | - |

## CD-11 Summary

**图片包含 屏幕截图

描述已自动生成**

Figure 14: Summary class

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | render () |
| **Parameters** | - |
| **Description** | It is used to render sparkline, line chart, bar chart and stacked chart. |
| **Returns** | SparkLine, LineChart, Barchart, StackedChart |

## CD-12 SparkLine

**图片包含 屏幕截图

描述已自动生成**

Figure 15: SparkLine

**Field Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | DataServices | dataServices | It is an instance of DataService |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | render () |
| **Parameters** | - |
| **Description** | It is used to render component |
| **Returns** | SparkLine |

## CD-13 LineChart

**图片包含 屏幕截图

描述已自动生成**

Figure 16: LineChart

**Field Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | DataServices | dataServices | It is an instance of DataService |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | render () |
| **Parameters** | - |
| **Description** | It is used to render component. |
| **Returns** | LineChart |

## CD-14 BarChart

**图片包含 屏幕截图

描述已自动生成**



Figure 17: BarChart

**Field Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | DataServices | dataServices | It is an instance of DataService |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | render () |
| **Parameters** | - |
| **Description** | It is used to render component. |
| **Returns** | BarChart |

## CD-15 StackedChart

**图片包含 屏幕截图

描述已自动生成**



Figure 18: StackedChart

**Field Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | DataServices | dataServices | It is an instance of DataService |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | render () |
| **Parameters** | - |
| **Description** | It is used to render component |
| **Returns** | StackedChart |

## Class diagram for web services

## CD-16 DataService

**图片包含 屏幕截图

描述已自动生成**



Figure 19: DataService

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | getAccounts () |
| **Parameters** | - |
| **Description** | It is used to get accounts |
| **Returns** | array |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | getUser () |
| **Parameters** | - |
| **Description** | It is used to get users |
| **Returns** | array |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | getPending () |
| **Parameters** | - |
| **Description** | It is used to get pending |
| **Returns** | array |

## CD-17 UserService

**图片包含 屏幕截图

描述已自动生成**



Figure 20: UserService

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | register () |
| **Parameters** | user: userObject |
| **Description** | It is used to register account. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | login () |
| **Parameters** | user: userObject |
| **Description** | It is used to login to the system |
| **Returns** | token |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | adminLogin () |
| **Parameters** | user: adminObject |
| **Description** | It is used to admin log in to the system. |
| **Returns** | token |

|  |  |
| --- | --- |
| **ID** | 4 |
| **Name** | addUser () |
| **Parameters** | user: userObject |
| **Description** | It is used to add user. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 5 |
| **Name** | deleteUser () |
| **Parameters** | user: userObject |
| **Description** | It is used to delete user. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 6 |
| **Name** | editUser () |
| **Parameters** | oldData: userObject, newData: userObject |
| **Description** | It is used to edit user information. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 7 |
| **Name** | newData () |
| **Parameters** | : userObject |
| **Description** | It is used to get new data. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 8 |
| **Name** | approveAll () |
| **Parameters** | - |
| **Description** | It is used to approve all admin register requests |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 9 |
| **Name** | rejectAll () |
| **Parameters** | - |
| **Description** | It is used to reject all admin register requests |
| **Returns** | array |

|  |  |
| --- | --- |
| **ID** | 10 |
| **Name** | approve () |
| **Parameters** | info: userObject |
| **Description** | It is used to approval admin register requests |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 11 |
| **Name** | reject () |
| **Parameters** | info: userObject |
| **Description** | It is used to reject admin register requests |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 12 |
| **Name** | edit Password () |
| **Parameters** | info: userObject |
| **Description** | It is used to edit password |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 13 |
| **Name** | editUsername () |
| **Parameters** | info: userObject |
| **Description** | It is used to edit username information |
| **Returns** | - |

## Class diagram of Backend

## CD-18 Mongo

**图片包含 屏幕截图

描述已自动生成**

Figure 21: Mongo class

**Field Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | Flask(\_name\_) | app | Create Flask application |
| 2 | Default | string | app.config[‘MONGO\_DBNAME’] | Config the MongoDB name |
| 3 | Default | string | app. config[‘MONGO\_URL’] string | Config the MongoDB URL name |
| 4 | Default | string | app.config[‘JWT\_SECRET\_KEY’] | Config JWT\_SECRET\_KEY |
| 5 | Default | PyMongo(app) | mongo | Create PyMongo application by using app |
| 6 | Default | Bcrypt(app) | bcrypt | Create Bcrypt object by using app |
| 7 | Default | JWTManager(app) | jwt | Create JWTManager object by using app |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | register () |
| **Parameters** | - |
| **Description** | It is used to register account. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | login () |
| **Parameters** | - |
| **Description** | It is used to login to system. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | admin\_login () |
| **Parameters** | - |
| **Description** | It is used for admin to login to system. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 4 |
| **Name** | delete\_account () |
| **Parameters** | username: string |
| **Description** | It is used to delete account. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 5 |
| **Name** | get\_account\_by\_username () |
| **Parameters** | username: string |
| **Description** | It is used to get account by username |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 6 |
| **Name** | get\_all\_accounts () |
| **Parameters** | - |
| **Description** | It is used to get all users |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 7 |
| **Name** | edit\_username () |
| **Parameters** | - |
| **Description** | It is used to edit username |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 8 |
| **Name** | edit\_password () |
| **Parameters** | - |
| **Description** | It is used to edit password |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 9 |
| **Name** | approve\_user () |
| **Parameters** | - |
| **Description** | It is used to approve user account register request |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 10 |
| **Name** | reject\_user () |
| **Parameters** | - |
| **Description** | It is used to reject user account register request |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 11 |
| **Name** | get\_pending () |
| **Parameters** | - |
| **Description** | It is used to get pending |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 12 |
| **Name** | add\_user () |
| **Parameters** | - |
| **Description** | It is used to add new user |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 13 |
| **Name** | trial\_renew () |
| **Parameters** | - |
| **Description** | It is used to renew user trial time. |
| **Returns** | jsonify |

## 4.3 ER Diagram

图片包含 屏幕截图

描述已自动生成

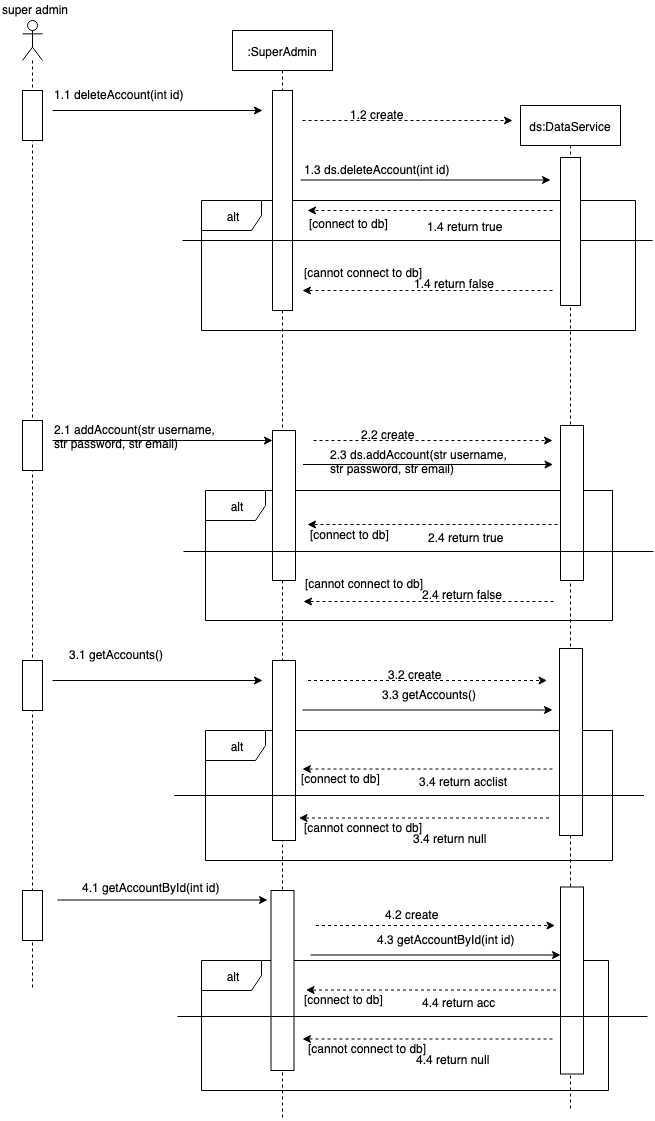
Figure 22: ER diagram

## 4.4 Sequence Diagram

(On the next page)

Figure 23: Sequence Diagram

Sequence diagram including URS-02: Super admin can view account management page, URS-03: Super admin can edit account information, URS-05: Super admin can add admin account directly.



(On the next page)

Figure 24: Sequence Diagram

This sequence diagram including URS-08: Admin can register account, URS-09: Admin can login to dashboard. URS-12: Admin can edit account information.URS-13: Admin can logout, URS-14: Admin can view all the summary sparkline chart, URS-15: Admin can view positive comments rate line chart, URS-16: Admin can view word frequency bar chart, URS-17: Admin can view number of comments bar chart, URS-18: Admin can view types of comments stacked column chart.

图片包含 文字, 地图

描述已自动生成chart, URS-18: Admin can view types of comments stacked column chart.

# 5. User Interface Design

## 5.1 Web Application UI

## 5.1.1 W-UI01-Homepage

图片包含 屏幕截图, 室内, 监视器, 电视

描述已自动生成

Figure 25: Homepage UI

This page is using for all users to view system, it consists of:

1. “Login” button for admin to login to dashboard.
2. “Register” button for register new account.
3. “Admin Login” button for super admin login to system.

## 5.1.2 W-UI02-Super admin login

图片包含 电子产品

描述已自动生成

Figure 26: Super admin login UI

This page is using for Super admin to login to system, it consists of:

1. Text flied used for input username and password.
2. “Login” button for login to dashboard.
3. “Cancel” button for going to previous page.

## 5.1.3 W-UI03-Register

图片包含 电子产品

描述已自动生成

Figure 27: Register UI

This page is using for admin to register account, they will need to input

1. Username for creating a new account.
2. Password for login to new account.
3. “Cancel” button for going to previous page.

## 5.1.4 W-UI04-Admin Login

图片包含 电子产品

描述已自动生成

Figure 28: Admin Login UI

This page is using for admin to login to system, it consists of:

1. Text flied used for input username and password.
2. “Login” button for login to dashboard.
3. “Cancel” button for going to previous page.

## 5.1.5 W-UI05-Add accounts

图片包含 屏幕截图

描述已自动生成

Figure 29: Add accounts UI

This page is using for super admin to add accounts for admin, it consists of URS-02, URS-03, URS-04, URS-05, URS-06, URS-07, the page consists of:

1. Text flied used for input username and trial time.
2. A search button for searching admin account.
3. “Status” button for accepting or rejecting admin register request.
4. “✓” button for saving change.
5. “X” button for canceling change.

## 5.1.5 W-UI06-Dashboard

图片包含 监视器, 室内, 电子产品, 就坐

描述已自动生成

Figure 30: Dashboard UI

1. “Summary” button for showing all data visualization summaries which have:

1.1 All the summary sparkline chart

1.2 Positive comments rate line chart

1.3 Word frequency bar chart

1.4 Number of comments bar chart

* 1. Types of comments stacked column chart

1. “Word-Cloud” button for showing the word cloud.
2. “Heatmap” button for showing the heatmap.
3. “Logout” button for decision maker to logout from dashboard.

## 5.1.7 W-UI07-User profile page

图片包含 监视器, 天空, 屏幕截图

描述已自动生成

Figure 31: User profile page UI

This page displays the profile page after admin login, in consists of:

1. “Refresh” button for admin to refresh page.
2. “Edit” button for admin to edit their account information.
3. “Dashboard” button for admin to view the dashboard.
4. “Logout” button for admin to logout from dashboard.

## 5.1.8 W-UI08-Edit username page

图片包含 监视器, 室内, 屏幕截图, 屏幕

描述已自动生成

Figure 32: Edit username page UI

This page displays the edit page after admin login, in consists of:

1. “Cancel” button for admin to cancel.
2. “Confirm” button for admin to confirm changes.
3. “Username” to show the original username.
4. “New username” input box to change the username to what admin want.

## 5.1.9 W-UI09-Edit password page

图片包含 屏幕截图, 监视器, 屏幕, 室内

描述已自动生成

Figure 33: Edit password page UI

This page displays the edit page after admin login, in consists of:

1. “Cancel” button for admin to cancel.
2. “Confirm” button for admin to confirm changes.
3. “Old password” input box to input the old password.
4. “New password” input box to change the password to what admin want.

# 6. List of Figures

[Figure 1: System Architecture 8](#_Toc18271446)

[Figure 2: Component Diagram 11](file:////Users/almostloverv/Library/CloudStorage/iCloud%20Drive/Desktop/RIDT/ProgressI/Progress1SDD.docx#_Toc18271447)

[Figure 3: Class Diagram Overview of whole system 12](#_Toc18271448)

[Figure 4: App class 13](#_Toc18271449)

[Figure 5: Home class 14](#_Toc18271450)

[Figure 6: Register class 16](#_Toc18271451)

[Figure 7: Login class 18](#_Toc18271452)

[Figure 8: AdminLogin class 21](#_Toc18271453)

[Figure 9: Admin class 22](#_Toc18271454)

[Figure 10: Accounts class 25](#_Toc18271455)

[Figure 11: Dashboard class 27](#_Toc18271456)

[Figure 12: User Profile class 29](#_Toc18271457)

[Figure 13: EditProfileclass 32](#_Toc18271458)

[Figure 14: Summary class 36](#_Toc18271459)

[Figure 15: SparkLine 37](#_Toc18271460)

[Figure 16: LineChart 38](#_Toc18271461)

[Figure 17: BarChart 39](#_Toc18271462)

[Figure 18: StackedChart 40](#_Toc18271463)

[Figure 19: DataService 42](#_Toc18271464)

[Figure 20: UserService 44](#_Toc18271465)

[Figure 21: Mongo class 48](#_Toc18271466)

[Figure 22: ER diagram 52](#_Toc18271467)

[Figure 23: Sequence Diagram 53](#_Toc18271468)

[Figure 24: Sequence Diagram 55](#_Toc18271469)

[Figure 25: Homepage UI 57](#_Toc18271470)

[Figure 26: Super admin login UI 58](#_Toc18271471)

[Figure 27: Register UI 59](#_Toc18271472)

[Figure 28: Admin Login UI 60](#_Toc18271473)

[Figure 29: Add accounts UI 61](#_Toc18271474)

[Figure 30: Dashboard UI 62](#_Toc18271475)

[Figure 31: User profile page UI 63](#_Toc18271476)

[Figure 32: Edit username page UI 64](#_Toc18271477)

[Figure 33: Edit password page UI 65](#_Toc18271478)