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 8](#_Toc21515732)

[2. Introduction 9](#_Toc21515733)

[2.1 Purpose 9](#_Toc21515734)

[2.2 Intended Audience and Reading Suggestions 9](#_Toc21515735)

[2.3 Project Scope 10](#_Toc21515736)

[2.4 Acronyms 10](#_Toc21515737)

[3. System Architecture 11](#_Toc21515738)

[4. Detail Design (UML Diagram) 13](#_Toc21515739)

[4.1 Component Diagram 13](#_Toc21515740)

[4.2 Class Diagram 15](#_Toc21515742)

[Class diagram overview of whole system 15](#_Toc21515743)

[CD-01 App 17](#_Toc21515745)

[CD-02 Home 18](#_Toc21515746)

[CD-03 Register 20](#_Toc21515747)

[CD-04 Login 22](#_Toc21515748)

[CD-05 AdminLogin 25](#_Toc21515749)

[CD-06 Admin 26](#_Toc21515750)

[CD-07 Accounts 29](#_Toc21515751)

[CD-08 Dashboard 31](#_Toc21515752)

[CD-09 User Profile 34](#_Toc21515753)

[CD-10 EditProfile 36](#_Toc21515754)

[CD-11 Summary 41](#_Toc21515755)

[CD-12 SparkLine 42](#_Toc21515756)

[CD-13 LineChart 45](#_Toc21515757)

[CD-14 BarChart 46](#_Toc21515758)

[CD-15 StackedChart 47](#_Toc21515759)

[CD-16 DataService 49](#_Toc21515760)

[CD-17 UserService 52](#_Toc21515761)

[[Progress II] 55](#_Toc21515762)

[CD-18 Mongo 55](#_Toc21515763)

[CD-19 DataProvider 61](#_Toc21515764)

[CD-20 Log 65](#_Toc21515765)

[CD-21 Hotel 67](#_Toc21515766)

[CD-22 View 70](#_Toc21515767)

[CD-23 HeatMap 72](#_Toc21515768)

[CD-24 Word-Cloud 73](#_Toc21515769)

[4.3 ER Diagram 74](#_Toc21515771)

[4.4 Sequence Diagram 75](#_Toc21515772)

[5. User Interface Design 82](#_Toc21515773)

[5.1 Web Application UI 82](#_Toc21515774)

[5.1.1 W-UI01-Homepage 82](#_Toc21515775)

[5.1.2 W-UI02-Super admin login 83](#_Toc21515776)

[5.1.3 W-UI03-Register 84](#_Toc21515777)

[5.1.4 W-UI04-Admin Login 85](#_Toc21515778)

[5.1.5 W-UI05-Add accounts 86](#_Toc21515779)

[5.1.5 W-UI06-Dashboard 87](#_Toc21515780)

[5.1.7 W-UI07-User profile page 88](#_Toc21515781)

[5.1.8 W-UI08-Edit username page 89](#_Toc21515782)

[5.1.9 W-UI09-Edit password page 90](#_Toc21515783)

[5.1.10 W-UI10- All users view comments 91](#_Toc21515784)

[5.1.11 W-UI11- Admin writes comments 92](#_Toc21515786)

[5.1.12 W-UI12- Super admin views system log 93](#_Toc21515788)

[6. List of Figures 94](#_Toc21515789)

# 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 webserver 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 2: Component Diagram

## 4.2 Class Diagram

## Class diagram overview of whole system

(On the next page)

Figure 3: Class diagram

## 图片包含 屏幕截图 描述已自动生成

## 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. |
| 5 | Default | boolean | isWordCloudOpen | The boolean to judge if word cloud is opening |
| 6 | Default | boolean | isHeatMapOpen | The boolean to judge if heat map 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

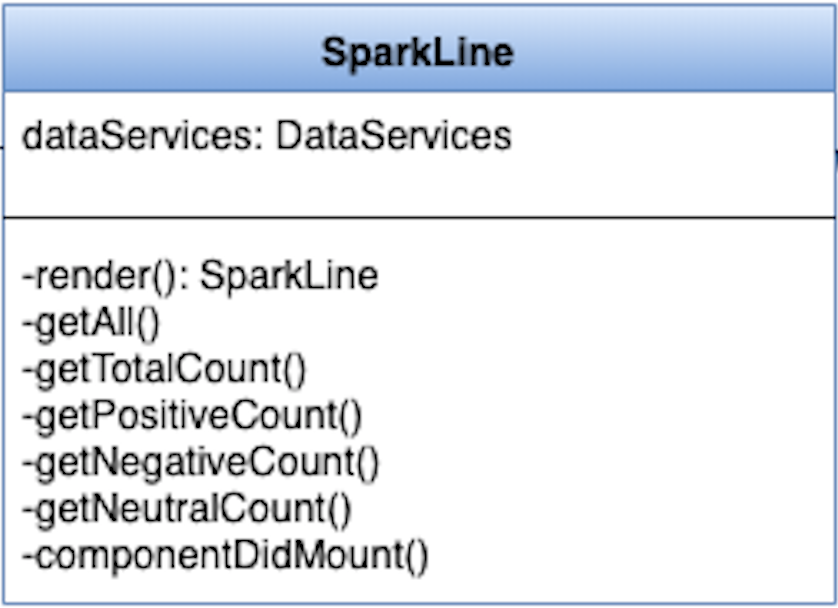
****

Figure 15: SparkLine class

**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> |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | getAll () |
| **Parameters** | - |
| **Description** | This method is used to get all data. |
| **Returns** | <Word-Cloud> |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | getTotalCount () |
| **Parameters** | - |
| **Description** | It is used to get total count |
| **Returns** | <Word-Cloud> |

|  |  |
| --- | --- |
| **ID** | 4 |
| **Name** | getPositiveCount () |
| **Parameters** | - |
| **Description** | It is used to get positive count |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 5 |
| **Name** | getNegativeCount () |
| **Parameters** | - |
| **Description** | It is used to get negative count |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 6 |
| **Name** | getNeutralCount () |
| **Parameters** | - |
| **Description** | It is used to get neutral count |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 7 |
| **Name** | componentDidMount () |
| **Parameters** | - |
| **Description** | It is used to load data from local storage after all components are rendered. |
| **Returns** | - |

## CD-13 LineChart

**图片包含 屏幕截图

描述已自动生成**

Figure 16: LineChart class

**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 class

**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** | - |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | componentDidMount () |
| **Parameters** | - |
| **Description** | It is used to render component. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | getAll () |
| **Parameters** | - |
| **Description** | It is used to render component. |
| **Returns** | - |

## CD-15 StackedChart

**图片包含 屏幕截图

描述已自动生成**



Figure 18: StackedChart class

**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 |

## CD-16 DataService

图片包含 屏幕截图

描述已自动生成

Figure 19: DataService class

**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 |

|  |  |
| --- | --- |
| **ID** | 4 |
| **Name** | getNumber () |
| **Parameters** | - |
| **Description** | It is used to get number |
| **Returns** | array |

|  |  |
| --- | --- |
| **ID** | 5 |
| **Name** | getTotalCount () |
| **Parameters** | - |
| **Description** | It is used to get total count |
| **Returns** | array |

|  |  |
| --- | --- |
| **ID** | 6 |
| **Name** | getPositiveCount () |
| **Parameters** | - |
| **Description** | It is used to get positive count |
| **Returns** | array |

|  |  |
| --- | --- |
| **ID** | 7 |
| **Name** | getNegativeCount () |
| **Parameters** | - |
| **Description** | It is used to get negative count |
| **Returns** | array |

|  |  |
| --- | --- |
| **ID** | 8 |
| **Name** | getNeutralCount () |
| **Parameters** | - |
| **Description** | It is used to get neutral count |
| **Returns** | array |

|  |  |
| --- | --- |
| **ID** | 9 |
| **Name** | getHotelList () |
| **Parameters** | - |
| **Description** | It is used to get hotel list |
| **Returns** | array |

|  |  |
| --- | --- |
| **ID** | 10 |
| **Name** | getHotelByName () |
| **Parameters** | string: hotel |
| **Description** | It is used to get hotel by name |
| **Returns** | array |

|  |  |
| --- | --- |
| **ID** | 11 |
| **Name** | addComment () |
| **Parameters** | comment: Comment |
| **Description** | It is used to add comments |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 12 |
| **Name** | getLog () |
| **Parameters** | - |
| **Description** | It is used to get log information |
| **Returns** | array |

## CD-17 UserService

**图片包含 屏幕截图

描述已自动生成**



Figure 20: UserService class

**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** | - |

## 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 |

|  |  |
| --- | --- |
| **ID** | 14 |
| **Name** | edit\_user () |
| **Parameters** | - |
| **Description** | It is used to edit user. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 15 |
| **Name** | approve\_all () |
| **Parameters** | - |
| **Description** | It is used to approve all request in waiting list. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 16 |
| **Name** | reject\_all () |
| **Parameters** | - |
| **Description** | It is used to reject all request in waiting list. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 17 |
| **Name** | get\_number () |
| **Parameters** | - |
| **Description** | It is used to get number. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 18 |
| **Name** | get\_total\_count () |
| **Parameters** | - |
| **Description** | It is used to get total counts |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 19 |
| **Name** | get\_count\_positive () |
| **Parameters** | - |
| **Description** | This method will receive http GET request and get the positive comment count by date from database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 20 |
| **Name** | get\_count\_negative () |
| **Parameters** | - |
| **Description** | This method will receive http GET request and get the negative comment count by date from database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 21 |
| **Name** | get\_count\_neutral () |
| **Parameters** | - |
| **Description** | This method will receive http GET request and get the neutral comment count by date from database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 22 |
| **Name** | get\_hotel () |
| **Parameters** | - |
| **Description** | This method will receive http GET request and get the number of comments of each hotel from database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 23 |
| **Name** | get\_hotel\_by\_name () |
| **Parameters** | string: hotel |
| **Description** | This method will receive http GET request and get the comment of inputting hotel from database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 24 |
| **Name** | add\_comment () |
| **Parameters** | - |
| **Description** | This method will receive http POST request and store the comment to database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 25 |
| **Name** | get\_log () |
| **Parameters** | - |
| **Description** | It is used to get log information |
| **Returns** | jsonify |

## CD-19 DataProvider

**图片包含 屏幕截图

描述已自动生成**

Figure 22: DataProvider class

**Field Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | pymongo | client | It is an instance of pymongo |
| 2 | Default | client | db | It is an instance of pymongo client. |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | get\_total () |
| **Parameters** | - |
| **Description** | This method will get counts of comments from database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | get\_positive () |
| **Parameters** | - |
| **Description** | This method will get positive comment data from database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | get\_negative () |
| **Parameters** | - |
| **Description** | This method will get negative comment data from database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 4 |
| **Name** | get\_neutral () |
| **Parameters** | - |
| **Description** | This method will get neutral comment data from database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 5 |
| **Name** | get\_all () |
| **Parameters** | - |
| **Description** | This method will get comment data from database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 6 |
| **Name** | get\_count\_by\_date () |
| **Parameters** | - |
| **Description** | This method will get the comment count by date from database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 7 |
| **Name** | get\_count\_by\_date\_positive () |
| **Parameters** | - |
| **Description** | This method will get the positive comment count by date from database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 8 |
| **Name** | get\_count\_by\_date\_negative () |
| **Parameters** | - |
| **Description** | This method will get the negative comment count by date from database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 9 |
| **Name** | get\_count\_by\_date\_neutral () |
| **Parameters** | - |
| **Description** | This method will get the neutral comment count by date from database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 10 |
| **Name** | get\_hotel () |
| **Parameters** | - |
| **Description** | This method will get the number of comments of each hotel from database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 11 |
| **Name** | get\_hotel\_by\_name () |
| **Parameters** | string: name |
| **Description** | This method will get the comment of inputting hotel from database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 12 |
| **Name** | add\_comment () |
| **Parameters** | string: content, string: hotel |
| **Description** | This method will store the comment to database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 13 |
| **Name** | record\_log () |
| **Parameters** | string: user, string: event, string: content |
| **Description** | This method will record log information and store to database. |
| **Returns** | jsonify |

|  |  |
| --- | --- |
| **ID** | 14 |
| **Name** | get\_log () |
| **Parameters** | - |
| **Description** | This method will get the log information. |
| **Returns** | jsonify |

## CD-20 Log

**图片包含 屏幕截图

描述已自动生成**

Figure 23: Log class

**Field Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | logs | array | It is an array of log information |
| 2 | Default | columns | array | It is an array of columns. |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | componentDidMount () |
| **Parameters** | - |
| **Description** | It is used to get data from backend before rending |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | getAll () |
| **Parameters** | - |
| **Description** | This method is used to get all hotels. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | render () |
| **Parameters** | - |
| **Description** | It is used to render component |
| **Returns** | <MaterialTable> |

## CD-21 Hotel

**图片包含 屏幕截图

描述已自动生成**

Figure 24: Hotel class

**Field Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | comments | array | It is an array of comments. |
| 2 | Default | isLoggedln | boolean | It is used to judge if logged in |
| 3 | Default | comment | string | It is a string of comment. |
| 4 | Default | length | int | It is an integer of the length of displayed comment. |
| 5 | Default | username | string | It is account username |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | componentDidMount () |
| **Parameters** | - |
| **Description** | It is used to get data from backend before rending |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | getHotel () |
| **Parameters** | - |
| **Description** | This method is used to get all hotels. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | render () |
| **Parameters** | - |
| **Description** | It is used to render component. |
| **Returns** | listItems, <TestField> |

|  |  |
| --- | --- |
| **ID** | 4 |
| **Name** | onCommentChange () |
| **Parameters** | - |
| **Description** | It is used to view when comment is typing. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 5 |
| **Name** | clear () |
| **Parameters** | - |
| **Description** | It is used to clear input. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 6 |
| **Name** | submit () |
| **Parameters** | - |
| **Description** | It is used to submit comment. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 7 |
| **Name** | onBack () |
| **Parameters** | - |
| **Description** | It is used to listen the back event and return to previous page. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 8 |
| **Name** | logout () |
| **Parameters** | - |
| **Description** | It is used to log out. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 9 |
| **Name** | viewMore () |
| **Parameters** | - |
| **Description** | It is used to show more content. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 10 |
| **Name** | viewLess () |
| **Parameters** | - |
| **Description** | It is used to show less content. |
| **Returns** | - |

## CD-22 View

**图片包含 屏幕截图

描述已自动生成**

Figure 25: View class

**Field Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Modifier | Type | Field | Description |
| 1 | Default | hotels | array | It is an array of hotels |

**Method Detail:**

|  |  |
| --- | --- |
| **ID** | 1 |
| **Name** | componentDidMount () |
| **Parameters** | - |
| **Description** | It is used to get data from backend before rending |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | getAllHotel () |
| **Parameters** | - |
| **Description** | It is used to get all hotels data. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | render () |
| **Parameters** | - |
| **Description** | It is used to render component |
| **Returns** | - |

## CD-23 HeatMap

**图片包含 屏幕截图

描述已自动生成**

Figure 26: HeatMap class

**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** | <HeatMap> |

## CD-24 Word-Cloud

## 

Figure 27: Word-Cloud class

**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** | <Word-Cloud> |

|  |  |
| --- | --- |
| **ID** | 2 |
| **Name** | componentDidMount () |
| **Parameters** | - |
| **Description** | It is used to load data from local storage after all components are rendered. |
| **Returns** | - |

|  |  |
| --- | --- |
| **ID** | 3 |
| **Name** | getWords () |
| **Parameters** | - |
| **Description** | It is used to get words frequency. |
| **Returns** | - |

## 4.3 ER Diagram

图片包含 文字, 地图

描述已自动生成

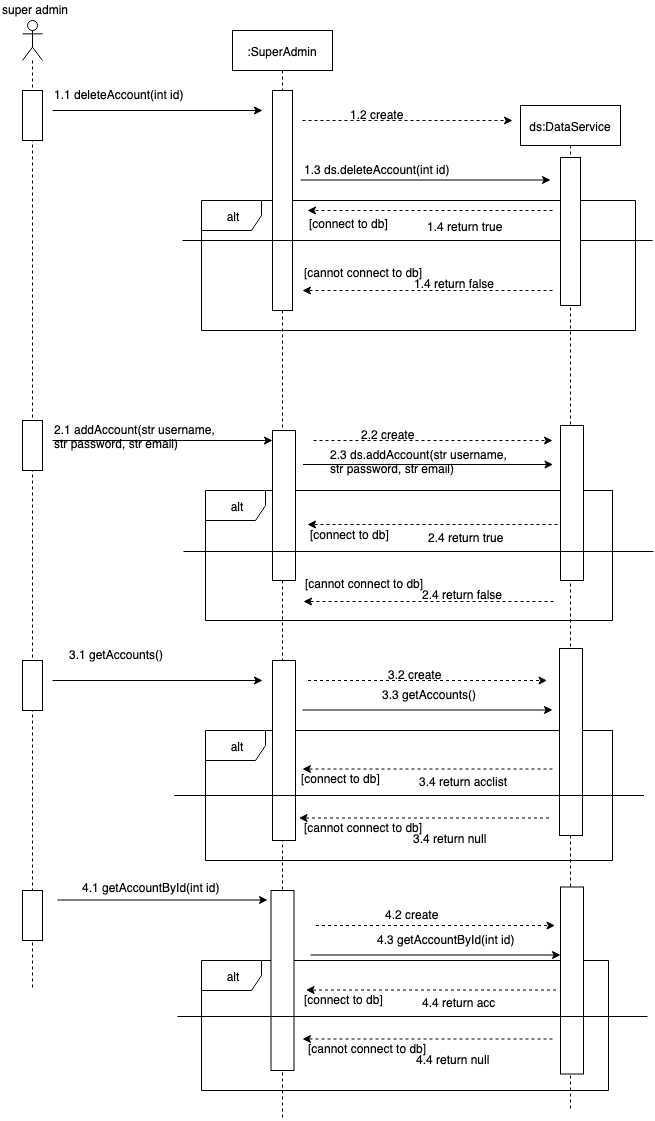
Figure 28: ER Diagram

## 4.4 Sequence Diagram

(On the next page)

Figure 29: 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 30: Sequence Diagram

This sequence diagram including URS-08: Admin can register account, URS-09: Admin can login to dashboard. URS-10: Admin can edit account information.URS-11: Admin can logout, URS-12: Admin can view all the summary by using sparkline chart, URS-13: Admin can view positive comments rate by using line chart, URS-14: Admin can view word frequency by using bar chart. URS-15: Admin can view number of comments by using bar chart, URS-16: Admin can view types of comments stacked by using 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.

(On the next page)

Figure 31: Sequence Diagram

This sequence diagram including URS-17: All the website user can view the real time comments in the first page, URS-18: All the website user can view hotel name, URS-19: All the website user can view the status of rating (1-5 stars). URS-20: Admin can write hotel comments.

**图片包含 地图, 文字

描述已自动生成**

**图片包含 屏幕截图

描述已自动生成**

Figure 32: Sequence Diagram

This sequence diagram including URS-21: Super admin can view the system log (Time, type of log, creator, info), URS-22: Super admin can view online user, URS-23: Super admin can view number of users who are typing comments.

# 5. User Interface Design

## 5.1 Web Application UI

## 5.1.1 W-UI01-Homepage

图片包含 屏幕截图, 监视器, 室内, 电子产品

描述已自动生成

Figure 33: 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 34: 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 35: 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 36: 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 37: 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 38: 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 39: 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 40: 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 41: 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.

## 5.1.10 W-UI10- All users view comments

## 图片包含 屏幕截图, 笔记本电脑, 就坐 描述已自动生成

Figure 42: All users view comments UI

This page displays the comments, in consists of:

1. “Clear” button to clear the text field.
2. “Submit” button to submit comments.
3. List of comments.

## 5.1.11 W-UI11- Admin writes comments

## 图片包含 屏幕截图, 监视器 描述已自动生成

Figure 43: Admin writes comments UI

This page displays the admin writes comments page, in consists of:

1. “Clear” button to clear the text field.
2. “Submit” button to submit comments.
3. “Back to home” button to return previous page
4. List of comments.
5. “Logout” button to logout system.

## 5.1.12 W-UI12- Super admin views system log

图片包含 屏幕截图, 监视器

描述已自动生成

Figure 44: Super admin views system log UI

This page is using for super admin to view system log, the page consists of:

1. System logs Id
2. System logs User
3. System logs Time
4. System logs Type
5. System logs Content

# 6. List of Figures

[Figure 1: System Architecture 9](#_Toc21481790)

[Figure 2: Component Diagram 12](file:////Users/almostloverv/Desktop/RIDT/ProgressII/Progress2SDD.docx#_Toc21481791)

[Figure 3: Class diagram 13](file:////Users/almostloverv/Desktop/RIDT/ProgressII/Progress2SDD.docx#_Toc21481792)

[Figure 4: App class 15](#_Toc21481793)

[Figure 5: Home class 16](#_Toc21481794)

[Figure 6: Register class 18](#_Toc21481795)

[Figure 7: Login class 20](#_Toc21481796)

[Figure 8: AdminLogin class 23](#_Toc21481797)

[Figure 9: Admin class 24](#_Toc21481798)

[Figure 10: Accounts class 27](#_Toc21481799)

[Figure 11: Dashboard class 29](#_Toc21481800)

[Figure 12: User Profile class 32](#_Toc21481801)

[Figure 13: Edit Profile class 34](#_Toc21481802)

[Figure 14: Summary class 39](#_Toc21481803)

[Figure 15: SparkLine class 40](#_Toc21481804)

[Figure 16: LineChart class 43](#_Toc21481805)

[Figure 17: BarChart class 44](#_Toc21481806)

[Figure 18: StackedChart class 45](#_Toc21481807)

[Figure 19: DataService class 47](#_Toc21481808)

[Figure 20: UserService class 50](#_Toc21481809)

[Figure 21: Mongo class 53](#_Toc21481810)

[Figure 22: DataProvider class 59](#_Toc21481811)

[Figure 23: Log class 63](#_Toc21481812)

[Figure 24: Hotel class 65](#_Toc21481813)

[Figure 25: View class 68](#_Toc21481814)

[Figure 26: HeatMap class 70](#_Toc21481815)

[Figure 27: Word-Cloud class 71](#_Toc21481816)

[Figure 28: ER Diagram 72](#_Toc21481817)

[Figure 29: Sequence Diagram 73](#_Toc21481818)

[Figure 30: Sequence Diagram 75](#_Toc21481819)

[Figure 31: Sequence Diagram 77](#_Toc21481820)

[Figure 32: Sequence Diagram 79](file:////Users/almostloverv/Desktop/RIDT/ProgressII/Progress2SDD.docx#_Toc21481821)

[Figure 33: Homepage UI 80](#_Toc21481822)

[Figure 34: Super admin login UI 81](#_Toc21481823)

[Figure 35: Register UI 82](#_Toc21481824)

[Figure 36: Admin Login UI 83](#_Toc21481825)

[Figure 37: Add accounts UI 84](#_Toc21481826)

[Figure 38: Dashboard UI 85](#_Toc21481827)

[Figure 39: User profile page UI 86](#_Toc21481828)

[Figure 40: Edit username page UI 87](#_Toc21481829)

[Figure 41: Edit password page UI 88](#_Toc21481830)

[Figure 42: All users view comments UI 89](#_Toc21481831)

[Figure 43: Admin writes comments UI 90](#_Toc21481832)

[Figure 44: Super admin views system log UI 91](#_Toc21481833)