

HSBC internal desk booking system



Team 5: Inconceivable Golden Bear

Colleen Vuong

Ellie Liu

Gaohan Chen

Jennifer Ling

Takeaki Yamasaki

Yozu Liang

Yun Meng

Roles

- Team Manager (on rotation): Jen, Yozu
- System Setup: Jen
- UI: Take, Yun
- Backend: Gaohan, Jen, Yozu
- Database: Ellie, Colleen

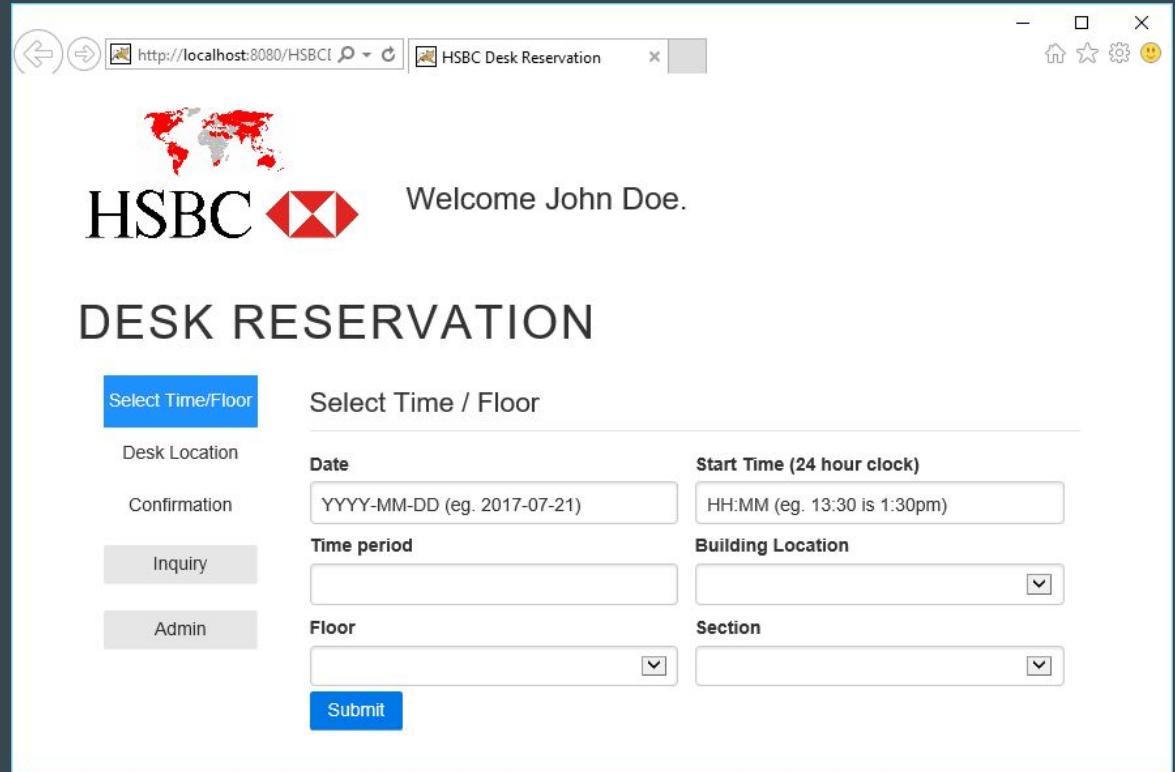
* Colleen, Jen, Gaohan are on UI team during initial development phase

User Interface

Reservation Page

Main Reservation Page

1. User inputs desk query information: start date, start time, building, period (hours), floor (optional), section (optional).
2. Clicking 'Submit' will lead to the Resource Selection Page.



The screenshot shows a web browser window with the URL `http://localhost:8080/HSBCI` and the page title "HSBC Desk Reservation". The page features the HSBC logo and a welcome message "Welcome John Doe.". The main heading is "DESK RESERVATION". On the left, there is a sidebar with a blue "Select Time/Floor" button, a "Desk Location" label, a "Confirmation" label, and two buttons: "Inquiry" and "Admin". The main content area has a heading "Select Time / Floor" and several input fields: "Date" (YYYY-MM-DD), "Start Time (24 hour clock)" (HH:MM), "Time period", "Building Location" (dropdown), "Floor" (dropdown), and "Section" (dropdown). A blue "Submit" button is at the bottom.

HSBC

Welcome John Doe.

DESK RESERVATION

Select Time/Floor

Desk Location

Confirmation

Inquiry

Admin

Select Time / Floor

Date

YYYY-MM-DD (eg. 2017-07-21)

Start Time (24 hour clock)

HH:MM (eg. 13:30 is 1:30pm)

Time period

Building Location

Floor

Section

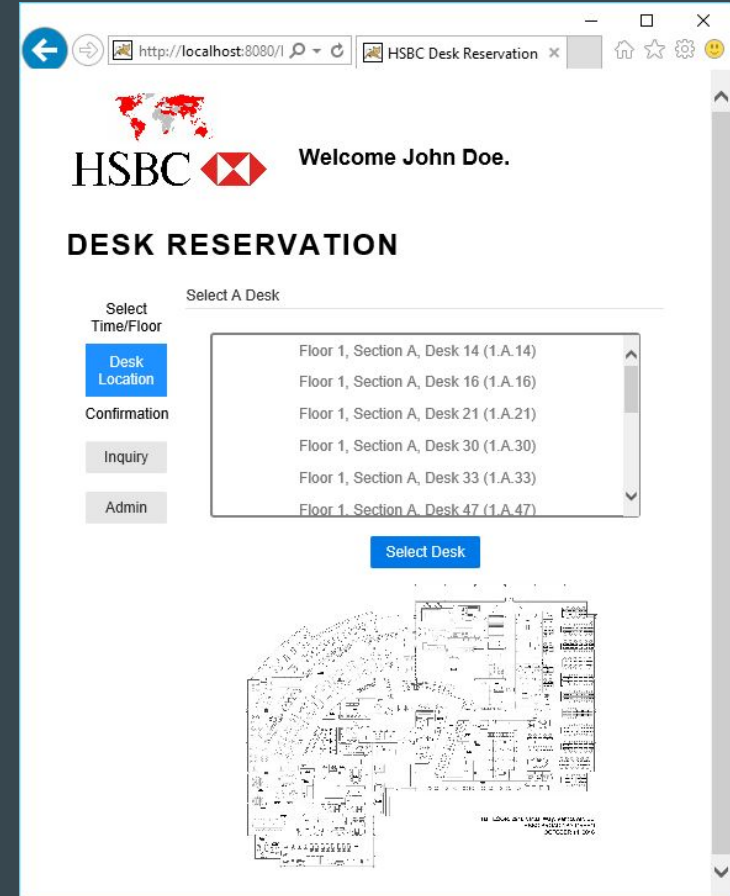
Submit

* There is currently no actual query to retrieve the available desks from the backend.

Reservation Page

Resource Selection Page

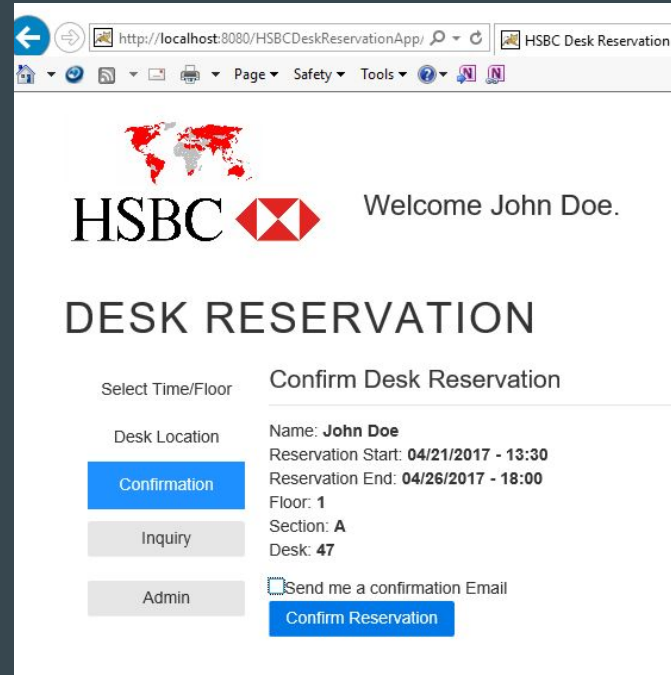
1. Select an available resource from the list.
2. A map at the bottom shows the floor plan for the listed resources.
3. After selecting a resource, the 'Select Desk' button is clicked on to go to the confirmation page.



Reservation Page

Confirmation Page

1. User looks over reservation information to ensure it is correct.
2. The user may check the box for sending confirm information to their staff email
3. Clicking the 'Confirm' button will officially book the reservation and the user will be lead back to the initial Reservation Page.



The screenshot shows a web browser window with the URL `http://localhost:8080/HSBCDeskReservationApp/`. The page features the HSBC logo and a welcome message "Welcome John Doe." Below this is the heading "DESK RESERVATION". The page is divided into two main sections: "Select Time/Floor" and "Confirm Desk Reservation".

Select Time/Floor

Desk Location

Confirmation

Inquiry

Admin

Confirm Desk Reservation

Name: **John Doe**
Reservation Start: **04/21/2017 - 13:30**
Reservation End: **04/26/2017 - 18:00**
Floor: **1**
Section: **A**
Desk: **47**

☐ Send me a confirmation Email

Confirm Reservation

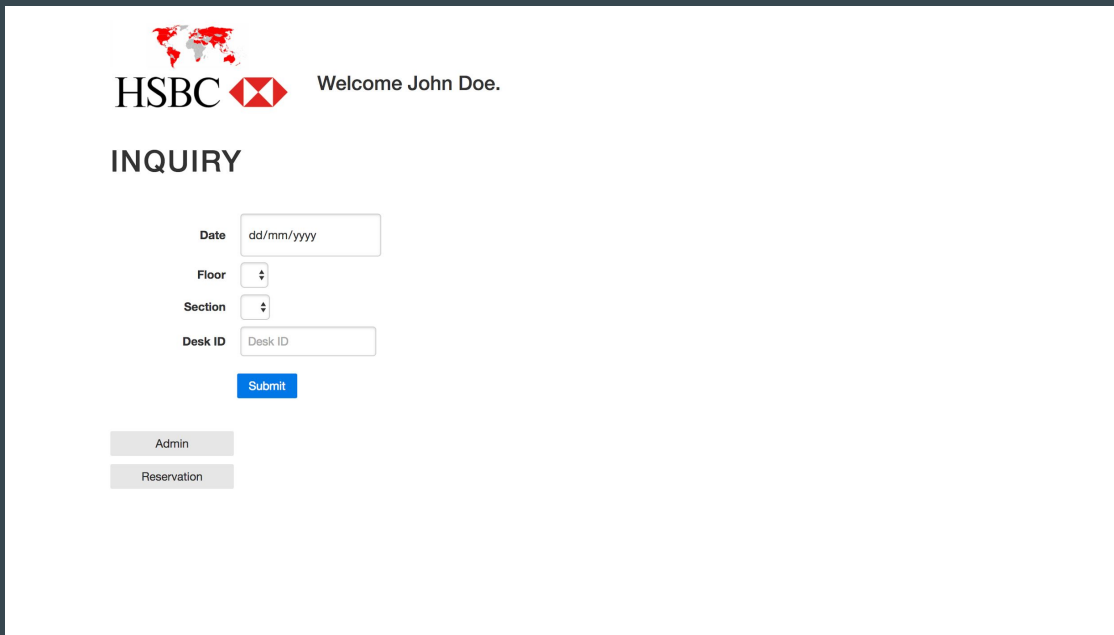
*The reservation details were hardcoded.

*There is no actual reservation processed and stored in the backend at this point.

Inquiry Page

Inquiry Page

1. User inputs desk query information: Date, floor, section, desk ID.
2. Clicking 'Submit' will lead to the result page.



The screenshot shows the HSBC Inquiry Page. At the top left is the HSBC logo, which includes a red and white globe icon and the text "HSBC". To the right of the logo is a welcome message: "Welcome John Doe." Below this is the heading "INQUIRY". The form contains four input fields: "Date" with a placeholder "dd/mm/yyyy", "Floor" with a dropdown arrow, "Section" with a dropdown arrow, and "Desk ID" with a placeholder "Desk ID". A blue "Submit" button is located below the "Desk ID" field. At the bottom of the form are two grey buttons: "Admin" and "Reservation".

* There is currently no actual query to retrieve the available desks from the backend.

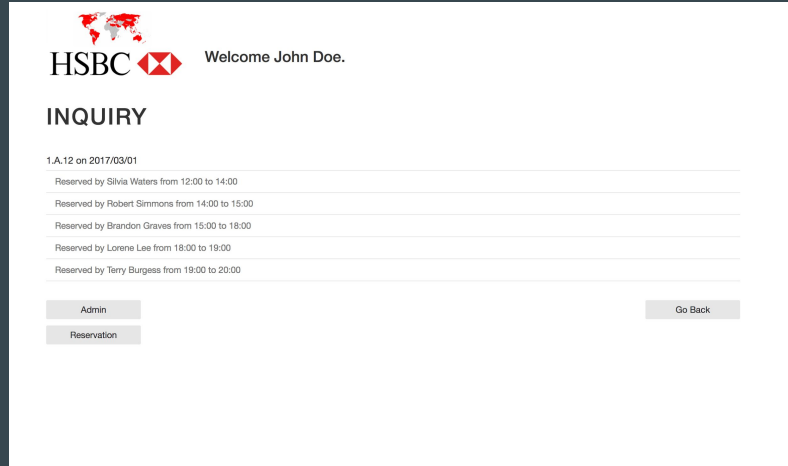
Inquiry Page


Inquiry Result Page

1. The list of users who reserved a desk is shown. Each entry can be expanded and users can see their details.

*The reservation details were hardcoded.

*There is no actual reservation processed and stored in the backend at this point.



HSBC  Welcome John Doe.

INQUIRY

1.A.12 on 2017/03/01

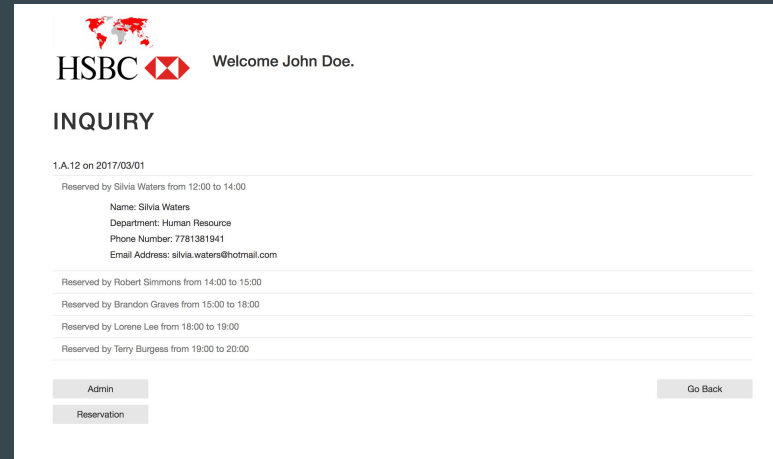
Reserved by Silvia Waters from 12:00 to 14:00
Reserved by Robert Simmons from 14:00 to 15:00
Reserved by Brandon Graves from 15:00 to 18:00
Reserved by Lorene Lee from 18:00 to 19:00
Reserved by Terry Burgess from 19:00 to 20:00


Admin

Reservation

Go Back

Expanded



HSBC  Welcome John Doe.

INQUIRY

1.A.12 on 2017/03/01

Reserved by Silvia Waters from 12:00 to 14:00
Name: Silvia Waters Department: Human Resource Phone Number: 7781381941 Email Address: silvia.waters@hotmail.com
Reserved by Robert Simmons from 14:00 to 15:00
Reserved by Brandon Graves from 15:00 to 18:00
Reserved by Lorene Lee from 18:00 to 19:00
Reserved by Terry Burgess from 19:00 to 20:00

Admin

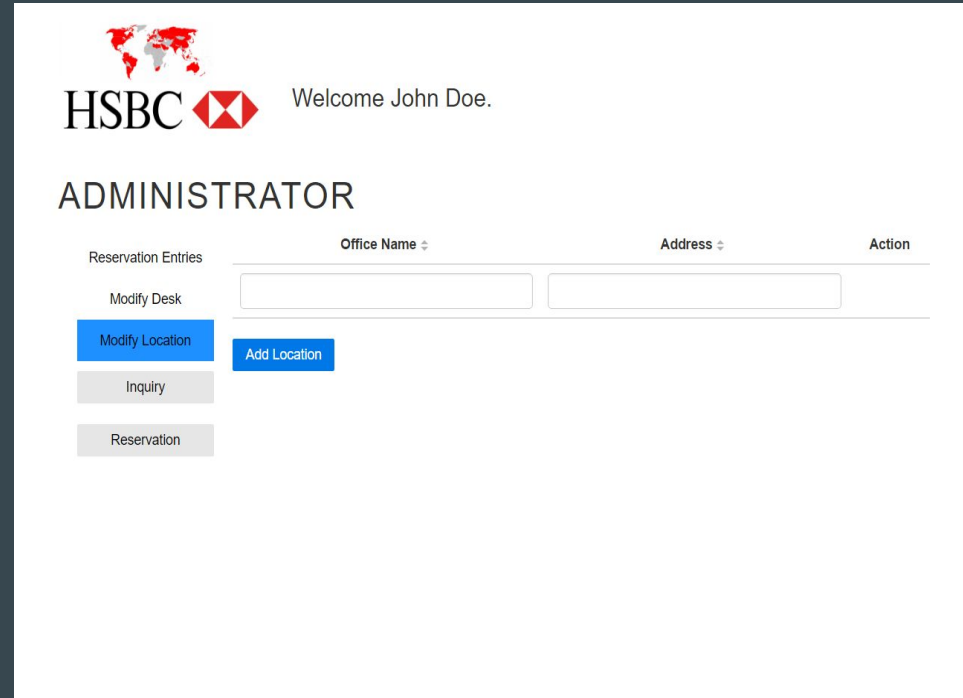
Reservation


Go Back

Admin Page

Modify Desk

1. Admin can edit the desk information
2. Admin can also add a desk. When “Add Desk” is clicked, it will lead to Add Desk page.
3. Admin can also delete desk.



HSBC  Welcome John Doe.

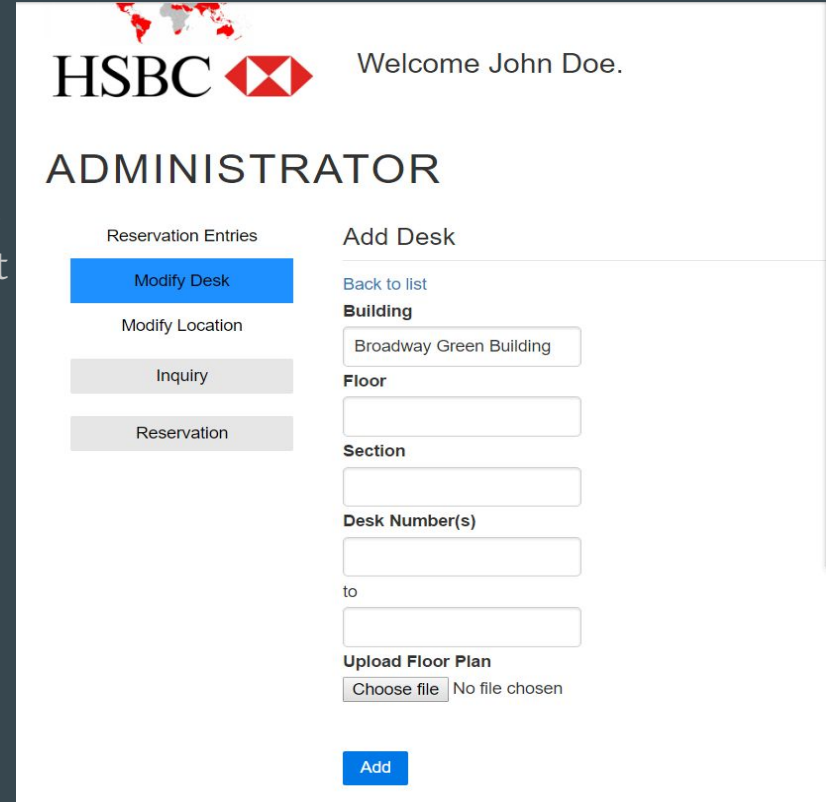
ADMINISTRATOR

Reservation Entries	Office Name	Address	Action
Modify Desk	<input type="text"/>	<input type="text"/>	
Modify Location		Add Location	
Inquiry			
Reservation			

Admin Page

Add Desk

1. Admin inputs the desk information to be added.
2. Building information is also required, default set to be “Broadway Green Building”
3. A range of desk numbers can be inputted.
(e.g. 101-110)
4. A floor plan may also be added (optional).
5. Clicking “Add” will pop-up message to confirm the desk has been added or display the appropriate error message.





The screenshot displays the HSBC Administrator interface. At the top left is the HSBC logo, and at the top right is a welcome message: "Welcome John Doe." Below this is the title "ADMINISTRATOR". On the left side, there is a sidebar with the heading "Reservation Entries" and four buttons: "Modify Desk" (highlighted in blue), "Modify Location", "Inquiry", and "Reservation". The main content area is titled "Add Desk" and contains the following form elements: a "Back to list" link, a "Building" label followed by a text input field containing "Broadway Green Building", a "Floor" label followed by an empty text input field, a "Section" label followed by an empty text input field, a "Desk Number(s)" label followed by an empty text input field, a "to" label followed by another empty text input field, an "Upload Floor Plan" label, a "Choose file" button, and the text "No file chosen". At the bottom right of the form is a blue "Add" button.

Admin Page

Reservation Entries

1. Admin can edit / delete each reservation entries made.
2. Clicking “Edit” icon next to each entry will lead to the Modify Entry Page.

**HSBC** 

Welcome John Doe.

ADMINISTRATOR











Reservation Entries

Modify Desk

Modify Location

Inquiry

Reservation

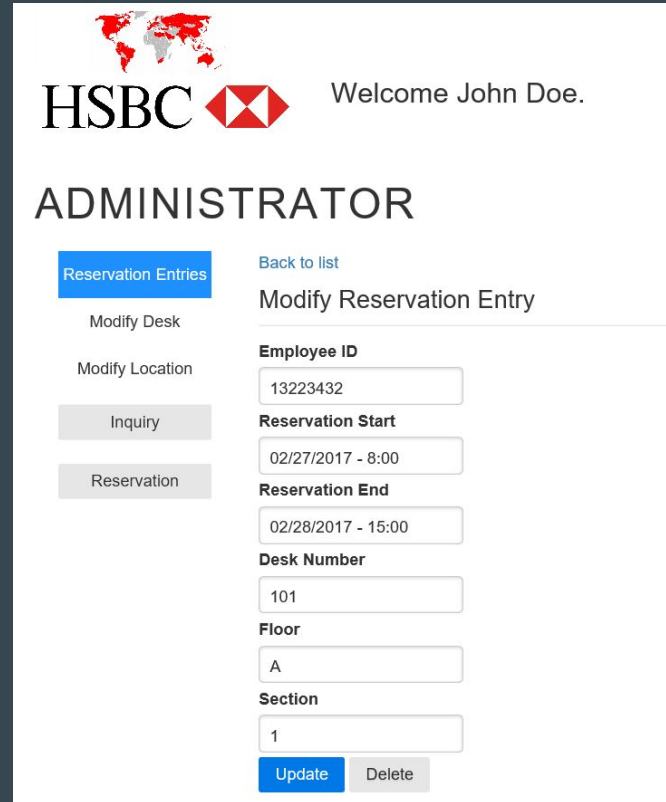
Id		Desk Location		Reservation Start	Reservation End	Action
<input type="text"/>		<input type="text"/>		<input type="text"/>	<input type="text"/>	
13223432		1.A.101		02/27/2017 - 8:00	02/28/2017 - 15:00	 
91234321		1.B.121		02/24/2017 - 15:00	02/26/2017 - 15:00	 
3719230		2.A.109		03/17/2017 - 11:00	03/18/2017 - 12:00	 
34234321		3.A.141		03/02/2017 - 9:00	03/04/2017 - 23:00	 
73892123		1.B.235		02/25/2017 - 17:00	02/26/2017 - 8:00	 

Admin Page


Modify Entry Page

1. Admin may update the info displayed or delete the reservation entry.
2. When “Update” is clicked, a pop-up message will be displayed to confirm the entry has been updated or display the appropriate error message.
3. When “Delete” is clicked, a pop-up warning message will be displayed asking for confirmation. When entry is deleted, user will be led back to list of reservation entries.

* May add other Employee info (read-only)



The screenshot shows the HSBC Administrator interface. At the top, there is a red world map icon, the HSBC logo, and a welcome message "Welcome John Doe." Below this is the title "ADMINISTRATOR". On the left side, there is a sidebar with a blue "Reservation Entries" button and three grey buttons: "Modify Desk", "Modify Location", and "Inquiry". Below these is a grey "Reservation" button. The main content area is titled "Modify Reservation Entry" and contains several form fields: "Employee ID" (13223432), "Reservation Start" (02/27/2017 - 8:00), "Reservation End" (02/28/2017 - 15:00), "Desk Number" (101), "Floor" (A), and "Section" (1). At the bottom right of the form are two buttons: a blue "Update" button and a grey "Delete" button.

HSBC  Welcome John Doe.

ADMINISTRATOR

[Reservation Entries](#)

[Back to list](#)

Modify Reservation Entry

Modify Desk

Modify Location

Inquiry

Reservation

Employee ID

13223432

Reservation Start

02/27/2017 - 8:00

Reservation End

02/28/2017 - 15:00

Desk Number

101

Floor

A

Section

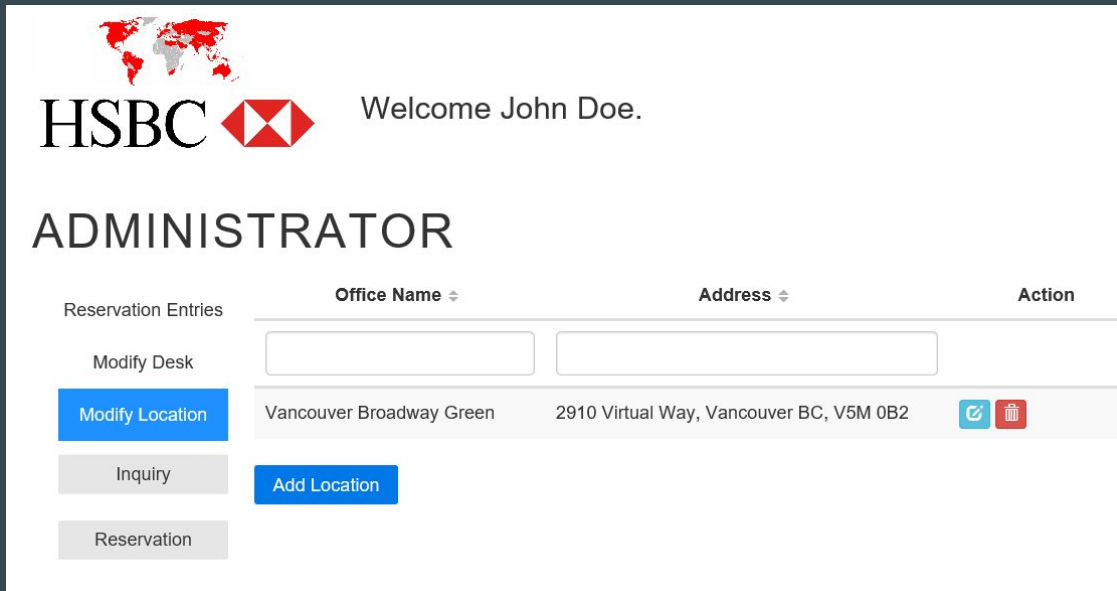
1

[Update](#) [Delete](#)

Admin Page

Modify Location page

1. Admin is able add a new location to the list.
2. Admin also able to edit / delete locations listed in the table.
3. The table allows users to filter / sort each column.



The screenshot displays the HSBC Administrator interface. At the top, the HSBC logo is accompanied by a world map icon and a welcome message: "Welcome John Doe." Below this, the title "ADMINISTRATOR" is prominently displayed. The main content area features a table with columns for "Reservation Entries", "Office Name", "Address", and "Action". The "Office Name" and "Address" columns contain input fields for filtering or sorting. The "Action" column includes icons for editing and deleting records. A sidebar on the left contains navigation buttons: "Modify Desk", "Modify Location" (highlighted in blue), "Inquiry", and "Reservation". A blue "Add Location" button is positioned below the table.

Reservation Entries	Office Name	Address	Action
Modify Desk	<input type="text"/>	<input type="text"/>	
Modify Location	Vancouver Broadway Green	2910 Virtual Way, Vancouver BC, V5M 0B2	Edit Delete
Inquiry	Add Location		
Reservation			

Backend

Communication

1. Uses Spring MVC4 http RESTful to receive request and json data from client

```
@RestController
@RequestMapping("/inquiry")
public class InquiryController {

    // returns JSON
    @RequestMapping(value = "/desk", method = RequestMethod.POST, produces = "application/json")
    public Inquire getDesk(@RequestBody Inquire inq) throws IOException {
        //TODO
        return inq;
    }
}
```

Challenges

1. Setting up the communication between angular (front-end) and Spring MVC4 (backend) is difficult due to deprecated methods examples from internet
2. Initially, was not able to convert incoming json data into user-defined java objects using Spring's `@RequestBody`

Solution

1. Using Jackson to parse json into java objects manually
2. Realized Jackson does not work if the object class is not in the same file
3. Illogical to have object classes (e.g. Desk.Class) in the same file as controller class
4. Worked when getters and setters are created for the class variables
5. But Jackson does not work if the class contains a constructor
6. Illogical to have a class without constructor, object cannot be instantiated other than by Jackson; e.g. Desk(int floor, String section, int deskNum)
7. Realized Jackson needs a default constructor to create object, e.g. Desk()
8. Realized Spring uses Jackson internally, so switched back to using Spring with the default constructor in place → Do not need an explicit Jackson parser for Json now!

Before

```
// returns JSON
@RequestMapping(value = "/desk", method = RequestMethod.POST, produces = "application/json")
public JSONObject getDesk(@RequestBody String deskStr) throws IOException {
    ObjectMapper mapper = new ObjectMapper();
    Desk desk = mapper.readValue(deskStr, Desk.class);

    int employee_id = 012211;
    String name = "Yozu Bear";
    String dept = "IT dept";
    String phoneNum = "1-604-235-8753";
    String email = "yozuBear@hotmail.com";
    Employee employee = new Employee(employee_id, name, dept, phoneNum, email);
    String startTime = "2:05 pm";
    String startDate = "Jan 30, 2017";
    int period = 16;
    ReservationBase base = new ReservationBase(employee, startDate, startTime, period);
    DeskSelection deskSelection = new DeskSelection(base, desk);

    boolean w = deskSelection.holdDesk();
    // Mock Response
    JSONObject deskJSON = new JSONObject();
    deskJSON.put ("response", deskSelection);
    return deskJSON;
}
```

After (What Jackson really needs)

```
package rest.domain;

/**
 * Created by yozubear on 2017-02-07.
 */
public class Desk {
    private int floor;
    private String section;
    private int deskNum;

    public Desk(){}

    public Desk(int floor, String section, int deskNum) {
        this.floor = floor;
        this.section = section;
        this.deskNum = deskNum;
    }

    public int getFloor() { return floor; }

    public String getSection() { return section; }

    public int getDeskNum() { return deskNum; }

    public void setFloor(int floor) { this.floor = floor; }

    public void setSection(String section) { this.section = section; }

    public void setDeskNum(int deskNum) { this.deskNum = deskNum; }
```

After

```
// returns JSON
@RequestMapping(value = "/desk", method = RequestMethod.POST, produces = "application/json")
public DeskSelection getDesk(@RequestBody Desk desk) throws IOException {
    int employee_id = 012211;
    String name = "Yozu Bear";
    String dept = "IT dept";
    String phoneNum = "1-604-235-8753";
    String email = "yozuBear@hotmail.com";
    Employee employee = new Employee(employee_id, name, dept, phoneNum, email);
    String startTime = "2:05 pm";
    String startDate = "Jan 30, 2017";
    int period = 16;
    ReservationBase base = new ReservationBase(employee, startDate, startTime, period);
    DeskSelection deskSelection = new DeskSelection(base, desk);

    boolean w = deskSelection.holdDesk();

    return deskSelection;
}
```

Database

```

CREATE DATABASE IF NOT EXISTS hsbcbanking;
USE hsbcbanking;

/*
Table location:
Stores list of HSBC office location
name: Vancouver Broadway
address: ....
*/
CREATE TABLE location (
  officename varchar(64) NOT NULL,
  address varchar(128) NOT NULL,
  PRIMARY KEY (officename)
);

INSERT location VALUES(
  'Vancouver Broadway Green',
  '2910 Virtual Way, Vancouver BC, V5M 0B2');

/*
Table admin:
Stores employee id that has been assigned as admin
adminid: some empid
name: name of the administrator, default is main admin
*/
CREATE TABLE admin (
  adminid varchar(64) NOT NULL,
  adminname varchar(128) NOT NULL,
  PRIMARY KEY (adminid)
);

```

1. Created tables for desk reservation related information.
2. Add existed location and test user/administrator.
3. Challenge: Still working on a way to parse 1600 desk location from excel file to sql table using only a few queries

Revision since TOR, Project Plan, Requirements and Design

1. Julin clarified at design meeting that 10 minutes countdown (holding time) starts after user submits desk preference rather than at the beginning
2. Database: deleted the duplicate table located_in, and added more reservation information in archive table for documentation.
3. Associate desks to building location
4. Use Bootstrap for front-end

Testing and delivery

1. Unit testing: JUnit 4.12, Karma 1.0, Jasmine 1.3 (unit testing)
2. End-to-end testing: Protractor
3. Load-testing HTTP servers: Apache JMeter
4. Final testing and deployment:
 - a. Windows laptop with Internet Explorer 11
 - b. Apache Tomcat server
 - c. MySQL database.

Demo

Next Steps

- Make style consistent amongst all UI pages
- Make all data input types consistent amongst all UI pages
- Integrate frontend, backend, and database (passing and retrieving data)
- Continue working on features for the minimum viable product
- Updating live documents as required

Questions from stakeholders