Web Database Applications: Report

Name

Institution of Affiliation
# Table of Contents

Assumptions.................................................................................................................. 3
Validation Constraints ........................................................................................................... 3
Test plan.................................................................................................................................. 6
  Customer registration .......................................................................................................... 6
  Customer authentication ...................................................................................................... 8
  Customer booking ................................................................................................................ 9
Entity relationship diagrams ............................................................................................... 10
  Conceptual Model ............................................................................................................. 10
  Logical Model .................................................................................................................. 11
Evaluation of the Application ............................................................................................... 12
Individual evaluation .......................................................................................................... 13
Web Database Applications: Report

Assumptions
The list of assumptions that governed the development of this application include:-

There were no immediate plans to increase the cinema venues so no form is provided for feeding additional venues into the system.

Customers would remember their mail addresses more than they would do with usernames hence the use of mail addresses for usernames.

Cinema booking records will be entered on the very date the user loads the cinema booking form hence the reason the booking_date field in the cinema_bookings table is set to default to the current date.

Finally, each of the cinema venues will be demarcated into lounges, and lounges will have seats belonging to given categories.

Validation Constraints
The validation constraints related to the cinema booking application include the following: -

Mail addresses that customers will register with will need to be unique in order to ensure that each customer receives information about the arrival of a new film. The system will use a back and a front end scheme to accomplish this constraint. On the back end, the email field will be declared as a unique column in the customers table as illustrated in the script below. On the front end, an AJAX call will be made to the server (after the customer has lost focus on the email form field) to instantly validate if the email that a customer supplies is available for use.
Customers won’t have their record saved until they provide a first name, last name, email and password. On the database, this will be validated by defining the aforementioned attributes with the **NOT NULL** keyword as illustrated in the script above. Equally, the front end will have javascript codes to validate the said fields against emptiness.

Customers will need to confirm their passwords whilst registering on the site as hinted in the customer registration form shown below.

The first password will be entered in the password field while the second password will fed in the field labeled confirm password. The system will save the first password, but only after verifying that it matches the second password. Verification of the password match is done using a javascript code on the front end.
To successfully login to the system, customers will need to supply an email and a password. The mail and password supplied will need to match the details they provided during registration.

Provision of incorrect login credentials will lead to the output shown here below.

![Customer Login Form](image)

Entries for the names of cinema venues will need to fall under South Greenwich, East Greenwich or West Greenwich. The project defines this at the database level by defining the venue column as `venue_name nvarchar(30) not null (check venue_name IN ('South Greenwich', 'East Greenwich', 'West Greenwich'))`;

A person charged to manage a cinema will need to be in the list of employees. In the back end, the application tracks staff in charge of cinemas via the employment number obtained from the list of employees. The cinema table definition to actualize this validation constraint is as shown below:-

```sql
CREATE TABLE [dbo].[cinemas]  
    ([cinema_no] [int] IDENTITY(1,1) NOT NULL,  
    [manager_id] [int] NOT NULL,  
    [venue_id] [int] NOT NULL,  
    PRIMARY KEY CLUSTERED  
    (cinema_no)  
    WITH (ADVERTISE_NAME = OFF, STATISTICSことでFREQUENCY = OFF, DASH_DELAYED KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY]
```

Seats in cinemas will need to be either available or unavailable. Thus, the database will validate entries for the status of seats using the constraint defined below:-
Prior to assigning a seat to a customer, the system will need to validate whether the seat in question (in relation to the assigned longue) bears an available status in the seats table. In the SQL server database used, this constraint is implemented using a before insert trigger to verify that the seat in picture is available for assignment.

**Test plan**

*Customer registration*

*Test for emptiness*

The form fields under the new customer registration form will be tested for emptiness. The test will involve two input types-correct and incorrect input. The incorrect input will be leaving all the form fields blank and hitting the register button. The pass criteria for this input type will be the system alerting the customer of the need to have all the form fields filled. On the other hand, the correct input will be hitting the register button after populating all the form fields. The pass criteria for this input type will be the system proceeding to evaluate the content of the individual form fields.

*Testing for alphabetic character set*

This test relates to the first name and last name fields. The incorrect input will be an attempt to register with names bearing non-alphabetic characters. The pass criteria for this input type will be the system alerting the user that the two aforementioned form fields only need alphabetic character sets as inputs. On the other hand, the correct input will be an attempt to register with
names bearing characters in the alphabets. The pass criteria for this input type will be the system proceeding to assess the mail field.

Test for mail correctness
This test will verify if the customer-supplied mail address is in the valid format. The incorrect input will be a mail address in an invalid format e.g. yahoo.com123 and fluez@yahoo. The pass criteria for this input type would be the system alerting the customer that the email supplied is not in a valid format. On the other hand, the correct input will be feeding a mail of a valid format like alejandro212@yahoo.com. The pass criteria for this input type will be the system proceeding to test the mail field for uniqueness with regards to the customer records already in the database.

Test for mail uniqueness
The form will be tested for the uniqueness of the supplied mail. The incorrect input will be an attempt to register with an already existing mail. The pass criteria for this input type would be the system, alerting the customer that the supplied mail is not available for use, with the suggestion that the customer chooses another mail address. On the other hand, the correct input will be the system proceeding to assess the password field.

Test for password length
The password field will be tested for password length, in this case set to have between 8 and 12 characters. The correct input will be an attempt to supply passwords falling out of the aforementioned range like rey, flu and melizedekuskuskuskus. The pass criteria for this input type will be the system alerting the user that passwords would need to have 8-12 characters. On the other hand, the correct input will be an attempt to issue passwords falling with the range mentioned above like testte12, IT@looper2016. The pass criteria for this input type will be the system proceeding to test for password match with the confirm password field.
**Test for password match**

The confirm password field will be matched with the password field. The incorrect input will be an attempt to feed non-matching passwords in the two fields. The pass criteria will be the system alerting the user that the two passwords would need to match. On the contrary, the correct input would be an attempt to feed matching passwords in the two fields. The pass criteria for this input type would be the system saving the customer records in the database alongside a notification of a successful registration.

**Customer authentication**

**Test for emptiness**

The email and the password field will be tested for emptiness. The incorrect input set will be leaving either or both of the form fields empty. The pass criteria for this input type will be the system alerting the user of the need to fill both the email and the password field. On the contrary, the correct input will be an attempt to fill both the mail and the password field prior to hitting the login button. The pass criteria for this input type will be system proceeding to test for a matching combination of mail and password.

**Test for matching login credentials**

The customer login form will be tested for a correct combination of email and password. The incorrect input will be an attempt to feed non-matching combination of email and password. For instance if a password wafteWAFTE@@ is associated with the user email trevor12@yahoo.com, then trying to login with the mail trevor12@yahoo.com and a different password or the password wafteWAFTE@@ and a different mail will lead to an error. The pass criteria for this input type will be the system alerting the customer that the supplied email and password do not match. On the contrary, the correct input will be a matching email and password.
combination. For instance, if a customer registers with the mail trevor12@yahoo.com and the password wafteWAFTE@%, then only that combination is issued during login. The pass criteria for this input type will be the system alerting the customer of a successful login followed by a redirection to the master page.

**Customer booking**

*Test for availability of assigned seat*

The system should verify that the seat assigned to a customer during booking is actually available as per the seats table. The incorrect input type will be an attempt to assign an unavailable seat to a customer. The pass criteria for this input would be a system alert that the seat in question is already assigned to another customer. On the contrary, the correct input would be an attempt to assign a customer a seat that is actually available. The pass criteria for this input would be the system’s action to save the book details into the database alongside a notification of records having been successfully saved.

*Test for seat count limit in cinemas*

Test that a seat assigned to a cinema does not result to a final count surpassing 200. The incorrect input type will be an attempt to add a seat to a cinema already with a seat count of 200. The pass criteria for this input will be a system notification that the cinema in question has already hit the maximum number of seats allowed. On the contrary, the correct input would be an attempt to assign a seat affiliated with a cinema that has not attained the maximum number of 200. The pass criteria will be the system alerting the user of the seat assignment having been successfully saved into the database.
Entity relationship diagrams

Conceptual Model

Customers
- Member_no
- first_name
- last_name
- email
- password

Employees
- staff_no
- first_name
- last_name
- position
- salary

Customer_Booking
- booking_id
- name
- customer_no
- staff_no
- lounge_no
- seat_no
- booking_date

Seat
- seat_no
- lounge_no
- category
- state
- price

Lounge
- lounge_no
- venue_id

Cinema_venue
- venue_id
- name
Evaluation of the Application
The back end database used was MSSQL 2014 enterprise edition. The reason for using a relational database in the back end was based on the need to support the ever changing content of the application e.g. tracking new cinema bookings, adding new employees and registering new customers.

The application uses a common connection string defined in the web.config file. This was a better alternative to defining database connection functionalities in the back end codes associated with the individual web pages. The approach would help ensure that anyone deploying the application only changes the database connection parameters defined in the web.config file. What’s more, this method of defining database connection parameters helps in saving the computational resources that would be needed in establishing database connections.

The system’s front end was created using ASP.NET web forms feature in Visual studio 2015. Use of web forms was informed by the idea that the organization would need its customers to access the application from whichever location, and using web forms allows the application to be deployed on web servers like internet information service then be available to access from any location.

Form fields in the system were validated using javascript. The reason for validation was to ensure that the database ends up with correct data. It is notable that some validation aspects have been handled at the database level too. Examples under this category include names for cinema venues, the status entry for seats and the total count of seats currently associated with a longue.

One way that could help in making the application better is the use of entity framework in the development process. Apart from reducing the time needed to construct the application, use of a
database first, code first or model first approach to framework would make the underlying code maintainable alongside tackling issues of restricting the implantation of the cinema booking application to SQL server.

**Individual evaluation**

The application is not configured to use caching hence could be prone to slow performance especially with regards to operations involving database querying. Use of HTML 5 caching technologies like memcached would be an awesome idea especially if Greenwich cinema booking company is considering scalability.

Use of the edit plus software in the design of the web pages was never a very awesome idea especially in the face of alternatives like Visual studio 2015 web form utility. Going the Visual studio way for the UI design would simplify the process of writing the corresponding C# codes.

The present application can only save records to a central database. Nothing is mentioned regarding data backups or fail-over clusters, yet this is crucial for enterprises running critical applications like Greenwich cinema booking company. A better design would involve proactive approaches to handle possible database server failures.