|  |  |
| --- | --- |
| **Medium** | **2017** |

**Capabilities covered:**

* Understand the web standards, patterns & protocols
* Define client & server architecture
* Implement Web UI that meets the functional specifications
* Add actions, interactions, business logic using JavaScript (ES6 & TS)
* Develop responsive pages using Material Design, Bootstrap & Media Query
* Follow coding standard and maintain high quality by analysing various reports generated through code quality tools

# Music Store Application

Design and implement a system that manages and organizes your favourite music genres and song collections. It enables users to store into and search for his/her favourite song from the store. Playing the music file/link is outside the scope of the system.

# System Features

|  |  |
| --- | --- |
| **Use Case #1** | **Add a song** |
| Trigger | User clicks on “Add a song” link on the home page |
| Pre-Conditions | - |
| Post-Conditions | The complete information about the song is stored in the application database |
| UI Screen Details | The input form contains the following fields   |  |  |  | | --- | --- | --- | | Song Title / Name | Text value |  | | Artist (s) | Text Value |  | | Location | Text value | http url to the song | | Album | Text Value |  | | Duration | Time | In seconds | | Tags | Text values |  | |
| Main Flow | The user provides the required details in the form and Clicks “Save” button. On successfully saving the song details, user is forwarded to the home page along with the message “Song details have been saved successfully” |
| Data Validations | * Input data has to be validated for numeric and date data fields. * Following fields are mandatory – Name and Location |
| Business Rules | None |
| Exception Flows | * In case of any failure in Data Validation or Business Rules, user should be brought back to the screen along with the error message “The value provided for the field …. Is incorrect or incomplete” |
| Alternate Flows | * When the user selects “Cancel” button, user should be taken back to the home page |

|  |  |
| --- | --- |
| **Use Case #2** | **Create and Search Music Playlists** |
| Trigger | User clicks on “Create and Search Music Playlists ” link on the home page |
| Pre-Conditions | Song details are in the database |
| Post-Conditions | Details of song playlists are stored in database |
| UI Screen Details | * ‘**Song list**’ table needs to be displayed. This displays the list of **all** the songs available in the database. Each song display will have a checkbox beside it. * ‘**Create Playlist**’ form needs to be displayed on the bottom. This form will have a text field for entering the Playlist name and the songs would be selected from the ‘Song list’ table.   + The form will have Save and Cancel button. Save button, on click, will validate user entry and store details of administered vaccination in database   + The cancel button, on click, will reset this form |
| Main Flow | When the page is loaded the ‘Song List’ form and the ‘Create Playlist’ table are displayed one below the other.  The ‘Song Playlist’ table is initially loaded with details in Database. Once the data is entered through the ‘Create Playlist’ form and stored successfully, report with playlists also could be displayed |
| Data Validations | * Input data has to be validated for numeric and date data fields. * Following fields are mandatory – Song Title and url to the song |
| Business Rules | * Name of the play list should be unique in the system |
| Exception Flows | In case of any failure in Data Validation or Business Rules, user should be brought back to the same screen along with a relevant error message |
| Alternate Flows | * When the user selects “Cancel” button, the form “Create Playlist” should be reset * When the user selects “Close” button, user should be taken back to the home page |