Pageantry:

A Pageant Voting System –

Documentation

# Applications Development and Emerging Technologies &

# Event Driven Programming

# CSci 23 & CSci 151

# By: Araneta, Vaughn Cedric L.

# BSCS-2

# Instructor:

# Jomari Joseph A. Barrera

Table of Contents

# About

# Technologies Used

# User and System Requirements

# Pre-requisites

# Use Case and Entity Relationship Diagrams

# UI Events

# Form Events

# Database Context

# Further References

About

# Base Definition

“Pageantry” is a desktop productivity application that is built and aimed to streamline the managing and voting aspects of a Pageant. Pageant events will be created and supervised by a manager. On the other hand, contestant judging is handled by a registered judge of that event.

# Beneficiaries

It has been acknowledged that there are at least 2 main beneficiaries who are considered to be users of the Pageantry project which include pageant managers and pageant judges. The main benefits provided to these aforementioned beneficiaries are as follows:

## Pageant Managers

* To create events and register judges and contestants based on customized specifications.
* To save time, and money regarding event contestant result calculations.
* To privately view pageant contestant results.

## Pageant Judges

* To vote once, per contestant, to ensure result information integrity.

# Goals

The Pageantry project aims to fulfill several goals requested by the aforementioned beneficiaries. The project goals are as follows:

1. To provide a platform for pageant management and voting.
2. To provide automatic event contestant result calculations.
3. To store and view pageant-related information.

# Scope

Having recognized its potential use in the future, if further development has been granted. Unfortunately, it has been decided that the version state of the Pageantry project will remain as a prototype.

Nonetheless, laying out the development scope has been recognized and noted since the first days of its development. The following project scopes include:

1. Must be developed in 3 months or less (Since February 15, 2024).
2. Must possess a local database connection for simplicity.
3. Must be developed, tested and deployed as a desktop application.

# Development Methodology

The Pageantry project is developed via the Rapid Application Development (R.A.D.) methodology. The following reasons as to why such a software development life cycle methodology is necessary are as follows:

1. Developing further features makes it stable and via the Individual development of modules and packages before overall integration.
2. A fixed, 3-month long timeframe makes it long enough to make room for a time extension due to unforeseen circumstances. Not only that, it is also a relatively short timeframe to avoid further extensions to a relatively "short" project.

Technologies Used

The Pageantry project has been decided to be implemented as a desktop software application. Therefore, the correct tools, methods, and environments must be used during development. The listed technologies as well as the rationale for using them are as follows:

# Programming Languages

## C#

* One of the main reasons as to why the C# programming language will be used for this project is due to one of the requirements of the Event Driven Programming subject – to develop a C# project. Such a programming language is also used to implement the Pageantry project via the .NET framework and WinForms GUI framework.

## SQL

* The SQL programming language is used to develop relational and MySQL databases.

# Databases

## MySQL

* The MySQL database is used to develop databases using the SQL programming language. Not only that, it is also stated in one of the requirements that the database used should be under a local connection, which the MySQL database can provide.

# Frameworks

## .NET

* As standard in all types of C# applications, the .NET framework includes built-in code and an environment to develop C# applications.

## WinForms

* The WinForms GUI framework will be used to implement the UI aspects of the Pageantry project. It is well noted, however, that the WinForms framework has been considered to be “difficult to use” compared to more modern alternatives such as the “Guna” framework. Having said that, it has been decided that the WinForms framework should be used to minimize the cost of learning a new framework due to relative time constraints.

# Editors

## Visual Studio 2022

* The Visual Studio 2022 IDE will be used to develop the Pageantry project. Not only that, the aforementioned IDE is well suited to develop C# applications.

## MySQL Workbench

* As aforementioned within the “Databases” section, the MySQL Workbench is used in order to develop MySQL databases.

User and System Requirements

# The user and system requirements are organized based on the main stakeholders of the Pageantry project. It will then be sub-organized based on their respective user requirements. Likewise, it will be sub-organized by its system requirements. The Pageantry project’s user and system requirements are as follows:

# Users (Managers and Judges)

1. Authentication and Authorization.
   1. Log in or Sign up a single user in under 2 seconds.
   2. The database must possess at least +1,000 users.
   3. Deletion of user information must not be allowed to maintain information integrity for past pageant events.
2. Editable Profiles.
   1. Modified information updates whenever the user chooses to exit customization, either from buttons or shortcut events.
   2. User roles can either represent a Manager or a Judge.

# Managers

1. Create and modify pageant event layout and profile.
   1. Strictly adheres to this layout sequence (Event -> Segment -> Round -> Criterium)
   2. A single event must contain at least one segment and ten segments at most.
   3. A single segment must contain at least one round and ten rounds at most.
   4. A single round must contain at least one criterion and ten criteria at most.
   5. The percentage weight of every criterium must sum up to exactly one hundred percent.
2. Register, modify, and remove pageant events, judges, and contestants.
   1. Pageant events must contain at least one judge and one hundred judges at most.
   2. Pageant events must contain at least two contestants and two hundred contestants at most.
3. Administering pageant events.
   1. Rounds will be administered one at a time, in order
   2. Contestants will be judged one at a time
   3. Administration will be handled by a single manager
   4. During a voting session, managers can wait until every judge has submitted their votes.
   5. In rare circumstances, a voting session can be forced to have every judge submit their votes, regardless of status.
   6. After every segment, contestants will be manually selected to either move on to the next segment or not.
4. View pageant event results.
   1. Tally of each contestant to every criterium to a particular event layout (Segments, Rounds, or Criteria).
5. View pageant events, judge, and contestant profiles.
   1. Must retrieve every necessary profile information in under one second.

# Judges

1. Judge contestants must exist during a pageant event
   1. Exactly one pageant contestant can be judged at any given time.
   2. Before judgment, every judge must first select an event to judge.
2. View pageant event, judge and contestant profiles
   1. Must retrieve every necessary profile information in under one second.

Pre-requisites

These are the following procedures before running the application in a hopefully, successful manner. Each must be executed in order.

1. Add a ‘User Variable’ within your environment variables in your desktop, name it ‘StringBuffer’ and set its value to ‘hfrRnNqYLb4K7Jx2u95mStDGsA3Ukzyv’. This will serve as the master key used for encrypting and decrypting data.
2. Install MySQL Community Workbench 8.0.
3. After installation, connect to a local connection, click on the ‘Server’ label on the top left menu. Then click on ‘Users and Privileges’.
4. Click on ‘Add Account’, set ‘Login Name’ on the right panel to ‘Admin’ and set its ‘Password’ to ‘hfrRnNqYLb4K7Jx2u95mStDGsA3Ukzyv’. Remember to confirm your password.
5. After creating the ‘Admin’ User, click on the House icon on the top left.
6. Next, setup a new MySQL Connection with these parameters:
   1. Connection Name: PageantVotingSystem
   2. Host Name: 127.0.0.1
   3. Port Number: 3306
   4. User name: Admin
7. When complete, test and create the MySQL Connection.
8. Next, open the ‘setup\_development.sql’ within the ‘PageantVotingSystem/Databases/’ folder, using the MySQL Workbench. Execute the file.
9. The database has now been setup. Now, open Visual Studio 2022, open the ‘PageantVotingSystem’ solution and run the program via the Green, right-pointing arrow on the top menu bar.

Use Case and Entity Relationship Diagrams

For better readability, the aforementioned diagrams will be located within the same folder as the Pageant Voting System Documentation. They are named as “Use Case Diagrams.drawio” and “Entity Relationship Diagrams.drawio”, respectively. These files will also include several versions as to how each design evolves over time.

Form Events

The following will include tables of Form Events, the Control Elements in which they listen to those Events, as well as a brief description regarding what happens when Control Elements invoke those said Events.

# About Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Hides the current Form and displays the previously displayed Form. |
| Form\_KeyDown | aboutForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |

# Confirmation Dialog Box Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | confirmationButton | Click | Hides the current Form. |
| Button\_Click | goBackButton | Click | The shortcut option for the ‘confirmationButton’ control element. |

# Message Dialog Box Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Hides the current Form. |

# Starting Menu Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | logInButton | Click | Displays to the ‘Log-in’ Form. |
| Button\_Click | signUpButton | Click | Displays to the ‘Sign-up’ Form. |

# Log-in Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the ‘Starting Menu’ Form. |
| Button\_Click | enterButton | Click | Displaying the next Form will be based on the User Role (See ‘Database Context’ for more information). The ‘Manager Dashboard’ Form will be displayed for Managers while the ‘Judge Dashboard’ Form will be displayed for Judges. |
| Form\_KeyDown | logInForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |
| Form\_KeyDown | form | KeyDown (Enter) | A shortcut version for ‘enterButton’. |

# Sign-up Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the ‘Starting Menu’ Form. |
| Button\_Click | enterButton | Click | Displaying the next Form will be based on the User Role (See ‘Database Context’ for more information). The ‘Manager Dashboard’ Form will be displayed for Managers while the ‘Judge Dashboard’ Form will be displayed for Judges. |
| Form\_KeyDown | signUpform | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |
| Form\_KeyDown | signUpForm | KeyDown (Enter) | A shortcut version of ‘enterButton’. |

# Information Not Found Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |

# Manager Dashboard Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | editEventButton | Click | Displays the ‘Edit Event’ Form. |
| Button\_Click | administerEventButton | Click | Displays the ‘Administer Event’ Form. |
| Button\_Click | eventResultsButton | Click | Displays the ‘Event Results’ Form. |

# Judge Dashboard Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | searchButton | Click | Queries and displays a list of Ongoing Pageant Events. |
| Button\_Click | judgeButton | Click | Displays the ‘Judge Contestant Dashboard’ Form. |
| Button\_Click | resetButton | Click | Clears the content of ‘eventQueryInput’, and ‘contestantsLayoutControl’. |
| EventQueryItem\_SingleClick | eventQueryLayout | SingleClick | Selects the Event, Queries the Contestants based on the selected Event who require judgement and displays the list of Contestants on the ‘selectedContestantLayout’. |
| SelectedContestantItem\_SingleClick | selectedContestantLayout | SingleClick | Selects the Contestant we want to Judge |

# Edit User Profile Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Button\_Click | signOutButton | Click | Signs out the current User and Displays the ‘Starting Menu’ Form. |
| Button\_Click | userProfileImageInput | Click | Displays a file dialog for selecting an image within the ‘PageantVotingSystem/Profiles’ folder. |
| Form\_KeyDown | editUserProfileForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |
| OpenFileDialog\_FileOk | userImageProfileFileDialog | FileOk | Confirms the image selection of the User and replaces the image of ‘userProfileImageInput’ with the one the User selected. |

# Event Contestant Profile Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Form\_KeyDown | eventContestantProfileForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |

# User Profile Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Form\_KeyDown | userProfileForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |

# Event Criterium Profile Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Form\_KeyDown | eventCriteriumProfileForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |

# Event Round Profile Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Form\_KeyDown | eventRoundProfileForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |

# Event Segment Profile Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Form\_KeyDown | eventSegmentProfileForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |

# Event Profile Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Form\_KeyDown | eventProfileForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |

# Event Contestants Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Button\_Click | informationButton | Click | Displays the ‘Event Contestant Profile’ Form based on the data from the selected Contestant. |
| Form\_KeyDown | eventContestantsForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |
| Item\_Click | sender | Click | Toggles the visibility of ‘optionsControl’. |

# Event Judges Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Button\_Click | informationButton | Click | Displays the ‘User Profile’ Form based on the data from the selected Judge. |
| Form\_KeyDown | eventJudgesForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |
| Item\_Click | sender | Click | Toggles the visibility of ‘optionsControl’. |

# Event Layout Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Button\_Click | informationButton | Click | Displays the Profile of the selected item of the ‘eventStructureItemLayout’ (Either Event, Segment, Round, or Criterium). |
| Button\_Click | resultsButton | Click | Displays the Contestant Results who participated on either a Segment, Round, or Criterium. |
| Form\_KeyDown | eventLayoutForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |
| Item\_Click | sender | Click | Toggles the visibility of ‘optionsControl’. |

# Event Criterium Result Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Form\_KeyDown | eventCriteriumResultForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |

# Event Round Result Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Form\_KeyDown | eventRoundResultForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |

# Event Segment Result Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Form\_KeyDown | eventSegmentResultForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |

# Event Results Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Button\_Click | searchButton | Click | Queries all Events created by the current Logged-in Manager. |
| Button\_Click | resetButton | Click | Resets the Form to its initial state. |
| Form\_KeyDown | eventResultsForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |
| DataOptionsLayoutItem\_Click | sender | Click | Toggles the visibility of the ‘dataOptionsLayoutControl’ control element. |
| ResultLayoutItem\_Click | sender | Click | If either ‘Layout’, ‘Judges’ or ‘Contestants’ were clicked, the ‘Event Layout’ , ‘Event Judges’ or ‘Event Contestants’ Form will be displayed respectively. |

# Judge Contestant Dashboard Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | confirmButton | Click | Displays the previously displayed Form. |
| Button\_Click | resetButton | Click | Resets the value of every ‘judgeCriteriumResultItemLayout’ control element to 0. |
| Button\_Click | fullInfoButton | Click | Displays the ‘Event Contestant Profile’ Form based on the selected Contestant. |
| Item\_SingleClick | sender | SingleClick | Displays the ‘Event Criterium Profile’ Form based on the selected Criterium. |

# Administer Event Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | contestantsButton | Click | Displays the ‘Administer Event Contestants’ Form. |
| Button\_Click | startVotingButton | Click | First, the program creates empty Result entities of the selected Contestant to the database. And second, the ‘Administer Voting Session’ Form will be displayed. |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Form\_KeyDown | administerEventForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |

# Administer Event Contestant Review Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | profileButton | Click | Displays the ‘Event Contestant Profile’ Form with the selected Contestant’s information. |
| Button\_Click | toggleButton | Click | Toggles the Contestant if he or she is set to either ‘Qualified’ or ‘Disqualified’. |
| Button\_Click | resultsButton | Click | Displays the ‘Event Segment Result’ Form based on the ongoing Segment’s information. |
| Button\_Click | confirmButton | Click | Confirms the status of each Contestant and displays the ‘Administer Event’ Form. |
| Item\_SingleClick | sender | SingleClick | Toggles the visibility of the ‘optionsControl’ control element. |

# Administer Event Contestants Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | profileButton | Click | Displays the ‘Event Contestant Profile’ Form based on the selected Contestant. |
| Button\_Click | selectButton | Click | Displays the ‘Administer Event’ Form. |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Item\_SingleClick | sender | SingleClick | Toggles the visibility of the ‘optionsControl’ control element. |
| Form\_KeyDown | administerEventContestantsForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |

# Administer Event Query Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | searchButton | Click | Queries Events considered ‘Pending’ that belong to the Logged-in Manager. |
| Button\_Click | profileButton | Click | Displays the ‘Event Profile’ Form based on the selected Event. |
| Button\_Click | administerButton | Click | If a Contestant is under judgement, the program displays the ‘Administer Voting Session’ Form. Otherwise, the program will directly display the ‘Administer Event’ Form. |
| Button\_Click | goBackButton | Click | Displays the ‘Manager Dashboard’ Form. |
| Button\_Click | clearButton | Click | Resets the Form to its initial state. |
| PendingEventsItem\_SingleClick | sender | SingleClick | Toggles the visibility of the ‘optionsControl’ control element. |
| Form\_KeyDown | administerEventQueryForm | KeyDown (Escape) | Displays the ‘Manager Dashboard’ Form. |

# Administer Voting Session Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | confirmButton | Click | Confirms the results of each Contestant. |
| Button\_Click | refreshButton | Click | Refreshes the judgement statuses of every registered Judge. |
| Item\_SingleClick | sender | SingleClick | Displays the ‘User Profile’ Form based on the selected Judge. |

# Edit Event Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | informationButton | Click | Displays the ‘Edit Event Profile’ Form. |
| Button\_Click | structureButton | Click | Displays the ‘Edit Event Segment Structures’ Form. |
| Button\_Click | judgesButton | Click | Displays the ‘Edit Event Judges’ Form. |
| Button\_Click | contestantsButton | Click | Displays the ‘Edit Event Contestants’ Form. |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Button\_Click | confirmButton | Click | First, it creates a new Event, its layout, registered judges and contestants. And second, it displays the ‘Manager Dashboard’ Form. |
| Button\_Click | resetButton | Click | Clears all information related to the new Event. |
| Form\_KeyDown | editEventForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |

# Edit Event Profile Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Button\_Click | resetButton | Click | Clears all information related to the new Event profile. |
| Form\_KeyDown | editEventProfileForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |

# Edit Event Segment Structure Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Button\_Click | createSegmentButton | Click | Creates a new Segment. |
| Button\_Click | resetButton | Click | Removes all Segments. |
| Form\_KeyPress | segmentsLayout.SelectedItem | KeyPress (Number Pad 8) | Moves the selected Segment upwards. |
| Form\_KeyPress | segmentsLayout.SelectedItem | KeyPress (Number Pad 2) | Moves the selected Segment downwards. |
| Form\_KeyPress | segmentsLayout.SelectedItem | KeyPress (Delete) | Deletes the selected Segment. |
| Form\_KeyPress | editEventSegmentStructureForm | KeyPress (Escape) | The shortcut option for the ‘goBackButton’ control element. |
| SegmentsLayoutItem\_SingleClick | sender | SingleClick | Selects a Segment. |
| SegmentsLayoutItem\_DoubleClick | sender | DoubleClick | Displays the ‘Edit Event Round Structure’ Form. |

# Edit Event Round Structure Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Button\_Click | createRoundButton | Click | Creates a new Round. |
| Button\_Click | roundResetButton | Click | Removes all Rounds. |
| Form\_KeyPress | roundsLayout.SelectedItem | KeyPress (Number Pad 8) | Moves the selected Round upwards. |
| Form\_KeyPress | roundsLayout.SelectedItem | KeyPress (Number Pad 2) | Moves the selected Round downwards. |
| Form\_KeyPress | roundsLayout.SelectedItem | KeyPress (Delete) | Deletes the selected Round. |
| Form\_KeyPress | editEventRoundsStructureForm | KeyPress (Escape) | The shortcut option for the ‘goBackButton’ control element. |
| RoundsLayoutItem\_SingleClick | sender | SingleClick | Selects a Round. |
| RoundsLayoutItem\_DoubleClick | sender | DoubleClick | Displays the ‘Edit Event Criterium Structure’ Form. |

# Edit Event Criterium Structure Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Button\_Click | createCriteriumButton | Click | Creates a new Criterium. |
| Button\_Click | criteriumResetButton | Click | Removes all Criteria. |
| Form\_KeyPress | criteriaLayout.SelectedItem | KeyPress (Number Pad 8) | Moves the selected Criterium upwards. |
| Form\_KeyPress | criteriaLayout.SelectedItem | KeyPress (Number Pad 2) | Moves the selected Criterium downwards. |
| Form\_KeyPress | criteriaLayout.SelectedItem | KeyPress (Delete) | Deletes the selected Criterium. |
| Form\_KeyPress | editEventCriteriumStructureForm | KeyPress (Escape) | The shortcut option for the ‘goBackButton’ control element. |
| CriteriumLayoutItem\_SingleClick | sender | SingleClick | Selects a Criterium. |

# Edit Event Contestants Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Button\_Click | createContestantButton | Click | Creates a Contestant. |
| Button\_Click | resetButton | Click | Removes all Contestants. |
| Button\_Click | contestantImageInput | Click | Displays the ‘OpenFileDialogue’ control element. |
| Form\_KeyPress | contestantLayout.SelectedItem | KeyPress (Number Pad 8) | Moves the selected Contestant Upwards. |
| Form\_KeyPress | contestantLayout.SelectedItem | KeyPress (Number Pad 2) | Moves the selected Contestant Downwards. |
| Form\_KeyPress | contestantLayout.SelectedItem | KeyPress (Delete) | Deletes the selected Contestant. |
| Form\_KeyPress | editEventContestantsForm | KeyPress (Escape) | The shortcut option for the ‘goBackButton’ control element. |
| ContestantsLayoutItem\_SingleClick | sender | SingleClick | Selects the Contestant. |
| OpenFileDialogue\_FileOk | contestantImageFileInput | FileOk | Confirms the file selection of the User. |

# Edit Event Judges Form

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *Control Element* | *Control Event* | *Description* |
| Button\_Click | goBackButton | Click | Displays the previously displayed Form. |
| Button\_Click | resetButton | Click | Removes all Judges. |
| Form\_KeyPress | selectedJudgesLayout.SelectedItem | KeyPress (Number Pad 8) | Moves the selected Judge upwards. |
| Form\_KeyPress | selectedJudgesLayout.SelectedItem | KeyPress (Number Pad 2) | Moves the selected Judge downwards. |
| Form\_KeyPress | selectedJudgesLayout.SelectedItem | KeyPress (Delete) | Deletes the selected Judge. |
| Form\_KeyPress | editEventContestantsForm | KeyPress (Escape) | The shortcut option for the ‘goBackButton’ control element. |
| JudgeQueryLayoutItem\_SingleClick | sender | SingleClick | Adds the Judge towards a list of selected Event Judges. |
| Control\_KeyDown | editEventJudgesForm | KeyDown (Escape) | The shortcut option for the ‘goBackButton’ control element. |

Database Context

Several pieces of database information might be left undisclosed at least given without context. On the following pages include every Look-up tables within the Pageantry database, their lookup values, as well as their respective descriptions.

# Judge Status Table

|  |  |
| --- | --- |
| *Attribute Value* | *Description* |
| Present | A Judge who is present in a Pageant Event. |
| Absent | A Judge who is not present in a specific Pageant Event. |

# Scoring System Table

|  |  |
| --- | --- |
| *Attribute Value* | *Description* |
| Percentage | Contestant ranking will be based on a Contestant’s Net Percentage. |

# User Role Table

|  |  |
| --- | --- |
| *Attribute Value* | *Description* |
| Manager | The main organizer of a Pageant Event. |
| Judge | Responsible for filling in results towards Contestants. |

# Round Status Table

|  |  |
| --- | --- |
| *Attribute Value* | *Description* |
| Complete | A round where every Judge has Voted on every Contestant on a specific Round. |
| Incomplete | A round where not every Judge has Voted on every Contestant on a specific Round. |

# Marital Status Table

|  |  |
| --- | --- |
| *Attribute Value* | *Description* |
| Single | In a single relationship. |
| Married | In a married relationship. |
| Divorced | In a divorced relationship. |
| With Significant Other | An unspecified romantic relationship. |
| Rather Not Say | Deliberately unfilled data. |

# Gender Table

|  |  |
| --- | --- |
| *Attribute Value* | *Description* |
| Male | Representing the Male gender. |
| Female | Representing the Female gender. |
| Non-binary | Representing Gender non-conforming. |
| Rather Not Say | Deliberately unfilled data. |

# Contestant Status Table

|  |  |
| --- | --- |
| *Attribute Value* | *Description* |
| Qualified | A Contestant who is qualified to participate on a certain Event Segment. |
| Disqualified | A Contestant who is not qualified to participate on a certain Event Segment. |

# Event Layout Status Table

|  |  |
| --- | --- |
| *Attribute Value* | *Description* |
| Complete | A complete Event Round. |
| Incomplete | An incomplete Event Round. |
| Pending | An ongoing Event Round. |

# Round Contestant Status Table

|  |  |
| --- | --- |
| *Attribute Value* | *Description* |
| Complete | A Contestant who is yet to be judged in a specific Round. |
| Incomplete | A Contestant who is yet to be judged in a specific Round. |
| Pending | A Contestant who is under judgement in a specific Round. |

Further References

# GitHub Repository:

* https://github.com/CedricDeVon/PageantVotingSystem.git