*Take One* movie Theatre Ticketing System

Requirments document for information development system

Project Phase Two

Johané le Roux 31614744

Christopher Slaghuis 31713858

Jean Marx 32313845

Reinardt Nel 27327884

James Uys 28461789

Contents

[**1.** **Project Plan** 2](#_Toc34314712)

[**2.** **Definitions, acronyms and abbreviations** 3](#_Toc34314713)

[**3.** **Project description and Scope** 4](#_Toc34314714)

[**4.** **Functional requirements i.e.** 5](#_Toc34314715)

[**a.** **business data requirements (inputs, outputs)** 5](#_Toc34314716)

[**b.** **business processes requirements and** 5](#_Toc34314717)

[**c.** **business interface requirements** 5](#_Toc34314718)

[**5.** **Non-functional requirements (categorized according to the PIECES Framework)** 6](#_Toc34314719)

[**6.** **Candidate Systems Matrix** 7](#_Toc34314720)

[**7.** **Feasibility Analysis Matrix** 8](#_Toc34314721)

[**8.** **Use-Case Glossary & Use-Case Model Diagram** 9](#_Toc34314722)

[**9.** **Examples of data, questionnaires, fact-finding techniques used** 10](#_Toc34314723)

[**10.** **Summary, future & further planning (PERT Chart)** 10](#_Toc34314724)

# **Project Plan**

# **Definitions, acronyms and abbreviations**

# **Project description and Scope**

The main objective of this project is to provide a computerized system for the BEAUT ART gallery to automate daily business processes and provide extensive business reporting to support business decisions. In addition the system should provide extensive Help functionality, backup functionality and data must be secured from unwanted access. Before the new system can be used in a production environment, all history data must be loaded from the existing Excel system. The FAST methodology, following several phases, will be used as a formal approach to develop and implement the system. The following functionality will be included in the scope of this project:

The *TAKE ONE movie theatre* is a new up and coming movie theatre in a residential area of South Africa. The movie theatre will show recently released **movies** as well as special movies for special occasions (e.g. Christmas season, Valentine’s day, ladies’ night). Audience members will be able to book **tickets** for the movie of their choice. Each day there will be a specific time schedule according to which the movies will be shown. **Members (loyalty program)** of the movie theatre will receive special deals and newsletters.

There is currently no *automated* system in place at the movie theatre, and the owners requested a system that is not run by a third party. However we will implement an Information System using existing hardware.

The system must include functionality for the following:

1. Maintain members of movie theatre
2. Maintain movies
   1. Regular movies that are newly released
   2. Special occasions (Valentines, ladies & boys nights, etc.)
3. Selling of tickets
   1. Notify client about ticket information when sale is complete.
   2. The notification will include their movie ticket code.
4. Indicate seats that are booked.
5. Export data
   1. Backups
   2. Profit margins
6. Request report:

* List of movies
* Attendance list of movies to determine popular genre per time period (e.g. comedy, drama, action)
* Audience size per time period (time of day, day of weekend)
* Graphical representation of popular times movies is watched.

# **Functional requirements**

Login Form

Maintain

Genres

Movie Schedule

Book Seats

Special occasions

Sell tickets

Movies

Backup Data

Reporting

|  |  |  |
| --- | --- | --- |
| **Maintain** | | |
| **Add new movies** | | |
| Input Data | Processing | Output |
| * Movie description * Movie genre * Ticket Price * Movie length * Cast and directors’ names | * Display a warning message if movie to be added is similar to a currently added movie. * When the save button is clicked and no movies of the same title exist, save the record to the database as well as mark movie times on the schedule. * Make the movie able to be booked by the public | * Display a warning message if a movie with the same title already exists. * Display a new message when the new movie has been added to the database and added to the schedule. * Display the scheduled movie time in the schedule. |

|  |  |  |
| --- | --- | --- |
| **Maintain** | | |
| **Edit Movies** | | |
| Input Data | Processing | Output |
| * Select Movie to edit * Select field to edit * Make edits to field * Click the edit button | * Display a warning message before editing * When the edit button is clicked validate all new data entered * Make the edited movie able to be booked by the public | * Display a warning message before editing * Display a new message when the edited movie has been added to the database and added to the schedule. * Display the scheduled movie time in the schedule. |

|  |  |  |
| --- | --- | --- |
| **Maintain** | | |
| **Schedule movies (ADD, EDIT, DELETE)** | | |
| Input Data | Processing | Output |
| * Select movie to schedule * Select time slot | * When the save button is clicked and no movies clash, save the record to the schedule. * Make the movie able to be booked by the public. * Check that there are no clashes between scheduled movies. * Display a warning message if a scheduled time is booked by a different movie. * When editing and deleting check to make sure now one has book the previously scheduled movie. | * Display a warning message if a scheduled time is booked by a different movie. * Display a message when the movie has been added to the schedule. * Display the scheduled movie time in the schedule. |

|  |  |  |
| --- | --- | --- |
| **Book tickets** | | |
|  | | |
| Input Data | Processing | Output |
| * Select a movie to book * Select a time slot * Select a seat | * Verify that the seats are available to be booked. * Display a warning message if the selected seats are already booked * Save the seat selection the database * Update the available seats for that movie | * Display a warning message if the selected seats are already booked * Display a message when the seats have been booked * Sent tickets to the customer. |

|  |  |  |
| --- | --- | --- |
| **Backup Data** | | |
|  | | |
| Input Data | Processing | Output |
| * Click a Button to backup data | * Check to see if there is enough available space in backup location. * Close the movie database * Copy database to backup location * Open database * Recorded date at time of backup | * Display a warning message if there is not enough space available in backup location * Display message when backup is complete * Display message if an error occurs |

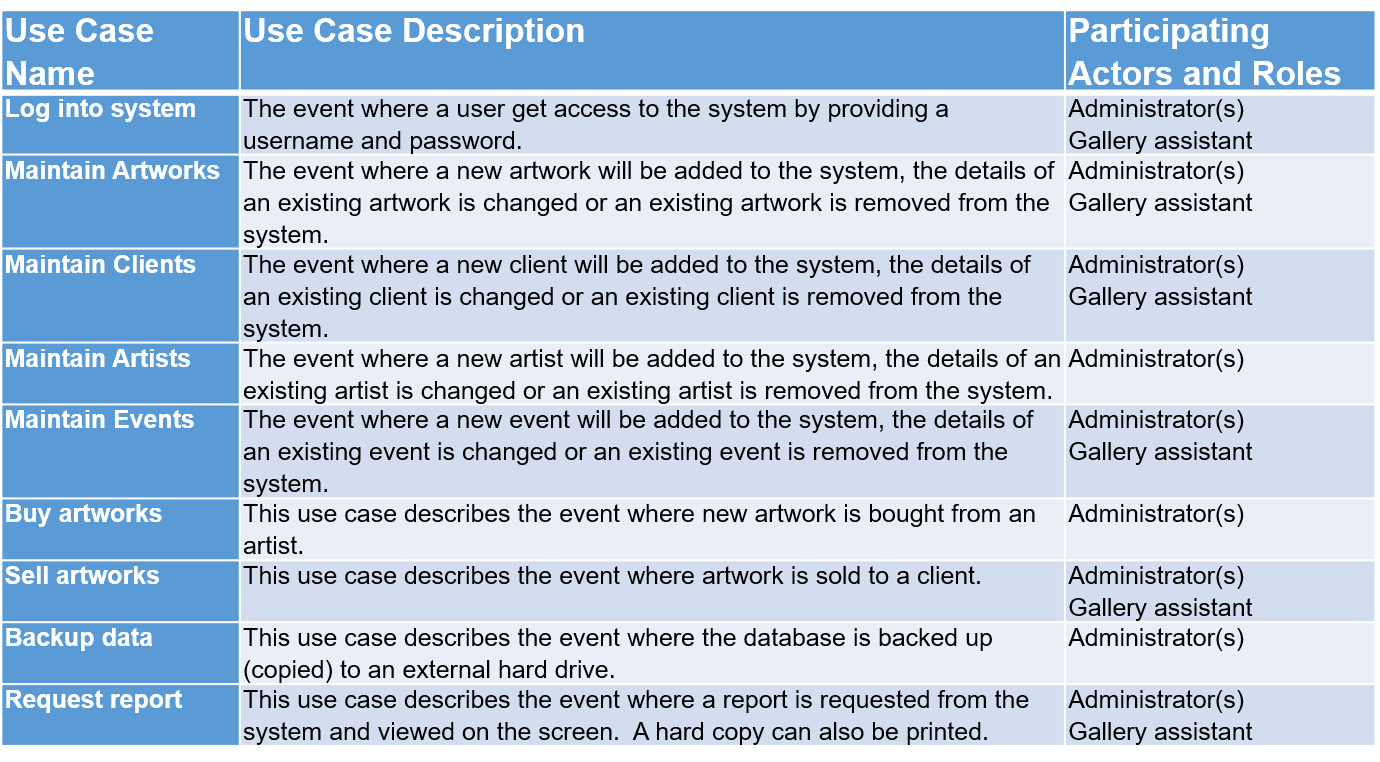
# Non-functional requirements (categorized according to the PIECES Framework)

|  |  |
| --- | --- |
| **Requirements** | **Piece categories** |
| Have unique identifiers for all employees and management | Security |
| Have a through help function | Efficiency of People and Processes |
| The database will hold information of all movies ever showed as well as when they were shown. | Performance  Information and Data |
| All queries made to the database should never take more than 5 Seconds | Performance |
| There will be three different users of the system. Users that will book tickets, employees who will be able to amend bookings and management who will be able to amend bookings, schedule movies and view reports created by the program. | Control or Security |

# **Candidate Systems Matrix**

# **Feasibility Analysis Matrix**

# **Use-Case Glossary & Use-Case Model Diagram**



# **Examples of data, questionnaires, fact-finding techniques used**

# **Summary, future & further planning (PERT Chart)**