|  |
| --- |
|  |
| Object Oriented Analysis and Design  Tsirogianni Vasiliki  Sohaib Shahid Saleem  Syed Faisal Imam  Federico Vivaldo |
| **Group 12** |

|  |
| --- |
|  |

**OOAD - 19/20 - Group 12 Coursework Checklist**

|  |  |  |
| --- | --- | --- |
| **Group Number :** | - Documentation **MUST** be in order  - Checklist **MUST** be submitted with Group Coursework  - **MUST** tick each box to verify each section have been submitted  - Every diagram must have a title | |
| ***Use Case (25)*** | | |
| Use Case Diagram | | Checkmark |
| Use Case Scenarios for THREE use cases | | Checkmark |
| Use Case Forms for THREE use cases | | Checkmark |
| ***Activity (10)*** | | |
| THREE Activity Diagrams | | Checkmark |
| ***Class (20)*** | | |
| Key abstraction form | | Checkmark |
| CRC cards | | Checkmark |
| Class Diagram | | Checkmark |
| ***Interaction (20)*** | | |
| 1st Sequence Diagram and Screen | | Checkmark |
| 2nd Sequence Diagram and Screen | | Checkmark |
| 3rd Sequence Diagram and Screen | | Checkmark |
| ***State (5)*** | | |
| State Diagram for a suitable class. (Track class) | | Checkmark |
| ***Report (10)*** | | |
| Team report **(10 marks) ………**   * Role Summary * Successes /Weaknesses * Team Meetings | | Checkmark |
|  | | |
| ***Not required in Group Report :***   * ***Individual Reports - (10 Marks ) - to be submitted by each group member*** | | |

**USE CASE DIAGRAM**

**A close up of a map

Description automatically generated**

**USE CASES FORMS**

|  |  |
| --- | --- |
| **Use Case Name** | **Process chart position** |
| **Description** | The Clerk updates tracks’ necessary information to create the “Chart Position” data. |
| **Actors** | Clerk |
| **Pre-conditions** | The Clerk needs to receive the Top 40 list from the Official Charts Organization, open the system and select from the “Track Details” section the “Update Chart” option from the drop – down menu. |
| **Trigger** | The Clerk must update the “Track Details” section. |
| **Main Flow of**  **Events** | 1. The Clerk enters the unique track No.**[**A1**]** 2. The system processes the information and displays track’s details. 3. The Clerk selects the “Chart type” from the options. (**new-entry** or **re-entry**) 4. The system updates the “Status” of the track,   the “Chart Entry Date” and the “Position” number.   * The “Status” becomes/remains “chart”. * The “Position” and the “Chart entry date”   are filled. (first available position number and present date).   1. The system updates all the tracks’ details that are not in the “Chart Position” page.  * The “Status” becomes “normal”. * The “Chart Entry Date” and the “Position” become blank.  1. The Clerk saves the data by clicking on the “save” button. **[**A2**]** 2. The system checks the “Status” of all tracks. 3. The “Chart Position” list is updated with the tracks that have “chart” status, (removes the “non-chart”). 4. The Clerk repeats the process for all the tracks on the list. |
| **Alternative flow**  **Of Events** | **A1)**The system processes the details without any result. Go back to step 1.  **A2)**The Clerk doesn’t save the document and decides to cancel the process by clicking the “cancel” button. End of Use Case. |
| **Post-condition** | All the tracks’ details are updated. |
| **Issues** | Can be labouring for Clerk to file data for every single track.  Can be a time-consuming process. |

**Process chart position – Scenario 1**

*This is the “normal” Main Flow of events*

Use case begins when the Clerk receives the Top 40 list from the Official Charts Organization. The Clerk opens the system and selects the “Track Details” section. He chooses the “Update Chart” option from the drop-down menu and then enters the unique track No. in the box. The system processes the given information and displays the track’s details. The Clerk chooses the “Chart type” for the track (new-entry or re-entry) by checking the proper box. The system updates the “Status" of the track (from “normal” to “chart” for a new-entry and “chart” again for a re-entry), fills the “Chart entry date” (present date) and the track “Position” number (first available number starting from 1). The system updates the tracks’ details that are not in the “Chart Position” page (changing the status to “normal”, leave the “Chart Entry Date” and “Position” blank). The Clerk saves the data by clicking on the “save” button. The system checks the “Status” of the tracks to find only the “chart” songs. The system then, adds to the “Chart Position” list the tracks with “chart” status, removes the “non-chart”, updating the “Chart Position” page. At last, The Clerk repeats the process for all the tracks on the list.

**Process chart position – Scenario 2**

*This is the “alternative” Flow of events*

Use case begins when the Clerk receives the Top 40 list from the Official Charts Organization. The Clerk opens the system and selects the “Track Details” section. He chooses the “Update Chart” option from the drop-down menu and then enters the unique track No. in the box. **The system processes the given information without any result.** **An error message appears, informing the Clerk that there is no track No. with this number. The Clerk inserts the track No. again.** The system processes the given information and displays the track’s details. The Clerk chooses the “Chart type” for the track (new-entry or re-entry) by checking the proper box. The system updates the “Status" of the track (from “normal” to “chart” for a new-entry and “chart” again for a re-entry), fills the “Chart entry date” (present date) and the track “Position” number (first available number starting from 1). The system updates the tracks’ details that are not in the “Chart Position” page (changing the status to “normal” and leave the “Chart Entry Date” and “Position” blank). The Clerk saves the data by clicking on the “save” button. The system checks the “Status” of the tracks to find only the “chart” songs. The system then, adds to the “Chart Position” list the tracks with “chart” status, removes the “non-chart” and updates the “Chart Position” page. At last, The Clerk repeats the process for all the tracks on the list.

**Process chart position – Scenario 3**

*This is the “alternative” Flow of events*

Use case begins when the Clerk receives the Top 40 list from the Official Charts Organization. The Clerk opens the system and selects the “Track Details” section. He chooses the “Update Chart” option from the drop-down menu and then enters the unique track No. in the box. The system processes the given information and displays the track’s details. The Clerk chooses the “Chart type” for the track (new-entry or re-entry) by checking the proper box. The system updates the “Status" of the track (from “normal” to “chart” for a new-entry and “chart” again for a re-entry), fills the “Chart entry date” (present date) and the track “Position” number (first available number starting from 1). The system updates the tracks’ details that are not in the “Chart Position” page (changing the status to “normal”, leave the “Chart Entry Date” and “Position” blank). **The Clerk decides to cancel the process by clicking the “cancel” button and exits the page.**

**A close up of a map

Description automatically generated**

|  |  |
| --- | --- |
| **Use Case Name** | **Prepare Current Chart Programme Schedule** |
| **Description** | The Station Controller creates 8 daily programme schedules, 3 hours long each, for the 24-hour radio station. |
| **Actors** | Station Controller |
| **Pre-conditions** | Identify Presenters available for next day. Station Controller needs to be able to access Track Details, Favorite Tracks, and Chart Positions. The Controller must select the “Programme schedule” page. |
| **Trigger** | Controller must prepare the daily programs. |
| **Main Flow of**  **Events** | 1. The Controller enters the: “Show date”, “Start time”, “Show type” and “End time”. **[A1]** 2. The system displays the “Programme schedule” page. 3. The system creates the “Programme schedule” list. 4. The system adds the tracks to the current “Chart position” programme list. 5. The “Presenter name” and “Presenter ID” are filled by the system. 6. The Controller saves the programme by clicking on the “save” button. **[A2]** 7. The relevant “Broadcast history” details of “Show date”, “Start time” and “Presenter” are filled in each track, in the “Track Details” page from the system. 8. The Controller repeats the process for all the chart current chart programs. |
| **Alternative flow**  **Of Events** | **A1)** The Controller inputs invalid details. Go back to step 1.  **A2)** There are not enough songs inputted in the schedule (corrupted songs). End of Use Case |
| **Post-condition** | Programme Schedules are completed and ready to use. |
| **Issues** | Can be laboring for Controller to prepare 8 different schedules every day.  Can be a time-consuming process.  Presenter can call in sick the next day meaning the Station Controller has to update the schedule. |

**Possible scenarios for the Prepare Current Chart Programme Schedule Use Case**

**Prepare Current Chart Programme Schedule – Scenario 1**

*This is the “normal” Main Flow of events*

The use case begins when the Controller selects the “Programme schedule” page to create a programme for next day.The Controller enters the: “Show date”, “Start time”. “Show type” and “End time” at the proper boxes. The system displays the “Programme schedule” page and creates a new empty “Programme schedule” list. The system adds the current “Chart position” tracks to the programme. The system fills the “Presenter name” and “Presenter ID” to the schedule and the Controller saves the data by clicking on the “save” button. The relevant “Broadcast history” details of “Show date”, “Start time” and “Presenter” are filled in each track, in the “Track Details” page from the system. The Controller repeats the process for all the current chart programs.

**Prepare Current Chart Programme Schedule – Scenario 2**

*This is the “Alternative” Flow of events*

The use case begins when the Controller selects the “Programme schedule” page to create a programme for next day.The Controller enters the: “Show date”, “Start time”. “Show type” and “End time” at the proper boxes. **The system processes the information without any result. An error message appears informing the** **Controller that the details are invalid. The Controller inserts the details.** The system displays the “Programme schedule” page and creates a new empty “Programme schedule” list. The system adds the current “Chart position” tracks to the programme. The system fills the “Presenter name” and “Presenter ID” to the schedule and the Controller saves the data by clicking on the “save” button. The relevant “Broadcast history” details of “Show date”, “Start time” and “Presenter” are filled in each track, in the “Track Details” page from the system. The Controller repeats the process for all the current chart programs.

*This is the “Alternative” Flow of events*

The use case begins when the Controller selects the “Programme schedule” page to create a programme for next day.The Controller enters the: “Show date”, “Start time”. “Show type” and “End time” at the proper boxes. The system displays the “Programme schedule” page and creates a new empty “Programme schedule” list. The system adds the current “Chart position” tracks to the programme. The system fills the “Presenter name” and “Presenter ID” to the schedule and the Controller saves it by clicking on the “save” button. **An error message appears, informing the Controller that there are not enough songs inputted in the schedule because of corrupted tracks. The Controller exits the page. The “Chart Position” page must be updated again.**

**A picture containing screenshot, text

Description automatically generated**

|  |  |
| --- | --- |
| **Use Case Name** | **Produce programme report** |
| **Description** | The Manager selects a report category from the menu. |
| **Actors** | Station Manager |
| **Pre-conditions** | The Manager opens the system and selects the “Reports” page. The Manager selects a category from the drop-down menu. All the necessary information must be completed form the employees. |
| **Trigger** | The Manager requests a programme report from the system. |
| **Main Flow of**  **Events** | 1. The Manager enters a specific date at the search box to get the report for that period. **[A1]** 2. The Manager clicks on the “next” button. 3. The system displays the details of that programme (Show date, Show Time, Start and End Time, Presenter, Played Tracks etc.) 4. The Manager checks the report and decides if he wants to exit from the page or to download/print the data. |
| **Alternative flow**  **Of Events** | **A1)** The Manager clicks on the “next” button, but the system cannot proceed to the next step. Go to step 3. |
| **Post-condition** | The report is produced. |
| **Issues** |  |

**Produce reports – Scenario 1**

*This is the “normal” Main Flow of events*

Use case begins when the manager requests a report (track, **programme** or statistics) from the system. The manager opens the “Reports” page and selects the “Programme Report” category from the drop- down menu. The manager enters a specific date at the search box to get the report for that day. The manager clicks on the “next” button and the system displays the report. The manager checks the report and decides if he wants to exit from the page or to download/print the data.

**Produce reports – Scenario 2**

*This is an “alternative” Flow of events*

Use case begins when the manager requests a report (track, programme or statistics) from the system. The manager opens the “Reports” page and selects the “Programme Report” category from the drop-down menu. The manager enters a specific date at the search box to get the report for that day. **The manager clicks on the “next” button, but the system cannot proceed to the next step because the date is wrong. A pop-up window displays a message prompting the manager to re-enter a proper a date, following the hints. The manager enters the new date and clicks on the “next” button**. The system displays the report, the manager checks the data and decides if he wants to exit from the page or to download/print the data.

A close up of a map

Description automatically generated

**Key Abstraction Form**

**Name of System : \_\_\_\_Unify Radio\_Group 12 \_\_\_\_\_\_**

|  |  |  |
| --- | --- | --- |
| **Candidate Key Abstraction** | **Reason For Elimination** | **Selected Component Name** |
| Record Company | Not relevant |  |
| MP3 tracks |  | Track class |
| Station Controller | Actor |  |
| Track Data Form | Not relevant |  |
| Unique Track Number | Attribute of Track |  |
| Title | Attribute of Track |  |
| Artist | Attribute of Track |  |
| Track length | Attribute of Track |  |
| Date arrived | Attribute of Track |  |
| Unify radio | The entire system |  |
| Track Status…(Normal/Fav/chart/unusable) | Attribute Track |  |
| Station Fav Tracks | List connected to Track |  |
| List of 30 tracks | List connected to Track |  |
| Official Chart Org | Not relevant |  |
| Top 40 | Collection of tracks |  |
| Clerk | Actor |  |
| Chart Position Sheet |  | Chart class |
| Week Start Date | Attribute of Chart |  |
| Position | Attribute of Track |  |
| New Entry | Value of an attribute of Track |  |
| Chart entry date | Attribute of Chart |  |
| Re-entry | Value of an attribute of Track |  |
| Non-chart tracks | Not relevent |  |
| Day | Not relevant |  |
| Presenter |  | Presenter class |
| 24 Hours | Not relevant to system |  |
| 3 hours | Not relevant to system |  |
| Type of programme | Attribute of Program Schedule |  |
| Programme Schedule |  | Program Schedule |
| Show date | Attribute of Program Schedule |  |
| Start time | Attribute of Program Schedule |  |
| Type of music | Same as show type |  |
| 3 categories(Show Type) | Same as show type |  |
| Current chart | Value of attribute show type in Programme Schedule class |  |
| Chart position sheet | Not relevant |  |
| Current week | Not relevant |  |
| Played | Attribute of Track |  |
| Time played | Attribute of Track |  |
| Blank | Not relevant |  |
| Oldie chart | Value of attribute show type in Programme Schedule class |  |
| Past 10 years | It’s not a class |  |
| Hit mix | Value of an attribute show type in Programme Schedule class |  |
| Tracks library | Not relevant |  |
| Chart show | Not relevant |  |
| Broadcast history | Broadcast History contains data of different classes- so probably not a class |  |
| Show | Same as programme |  |
| Number of times played | Static Attribute of Track |  |
| Station Manager | Actor |  |
| Genre | Attribute of Track |  |
| End Time | Attribute of ProgrammeSchedue |  |
| Presenter ID | Attribute of Presenter Class |  |
| Presenter Name | Attribute of Presenter Class |  |

CRC CARDS:

|  |  |
| --- | --- |
| Track | |
| Display Broadcast History  Update Track Details  Display Track Details | Chart  Presenter |

|  |  |
| --- | --- |
| Presenter | |
| Get Presenter Details  Update Broadcast History  Select Played Tracks | Programme Schedule  Track |

|  |  |
| --- | --- |
| ProgrammeSchedule | |
| Create new schedule  Input Show details  Display Programme Schedule | Chart  Presenter |

|  |  |
| --- | --- |
| Chart | |
| Week start date  Create Chart List | Chart |

CLASS DIAGRAM:

A close up of a map

Description automatically generated

A screenshot of a cell phone

Description automatically generatedA screenshot of a map

Description automatically generatedSEQUENCE DIAGRAMS:

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A close up of a map

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A close up of a map

Description automatically generatedSTATE DIAGRAM TRACK CLASS:

GROUP REPORT:

Group 12 for the Object Orientated Analysis and Design module consisted of: Vasiliki Tsirogrianni, Sohaib Shahid Saleem, Federico Vivaldo and Syed Faisal Imam. The requirements for this project consisted of: developing a Use Case Model including a Use Case diagram associated with scenarios and Use Case forms; Activity Diagrams following the steps of the new system; identifying Classes and Collaborations by creating a Class Diagram linked with a Key Abstraction, as well as highlighting the interactions between classes via CRC Cards; and finally, a State Diagram along with three Sequence Diagrams.

Vasiliki Tsirogrianni was the group leader of this project and her responsibilities were, to oversee the tasks being completed while completing some of the tasks on her own as well. She was key in completing the use case forms as she identified mistakes from the initial attempts and made the necessary changes when required. She was also in charge of designing all the final screen layouts.

Sohaib Shahid Saleem was the quality coordinator for the group project. Along with completing assigned tasks, he was responsible for ensuring all the work produced had no grammar errors as well as ensuring it fulfilled the coursework specification. He was also key in pointing out minor details that other group members did not pick up on which reduced errors for the work produced. He has been also involved in the initial draft of the key abstraction form, CRC cards and use case forms scenarios.

Federico Vivaldo was responsible for putting all the work onto the Star UML file as well as having an input in said tasks specified in the group meeting notes and great ideas during the entire process of analysis. He put in a lot of effort into studying sequence diagrams to ensure the competition of them as accurately to the specification as possible. Also been involved in the complete draft of the activity diagrams, use case and class diagrams.

Syed Faisal Imam was involved by completing the assigned tasks such as the state diagrams. Along with this, a complete overview of one of the significant use case including the form and scenarios linked with it.

Overall, the successes of the project were, every group member was turning up to the group meetings and putting ideas forward. Along with this, we managed to fulfil the schedule and complete the tasks assigned to each group member. In addition, each group member put in a lot of time and effort (often spending up to 4 hours per group meeting) into this project and they all shared the same goal of wanting a high pass in this project.

Despite the positives, the major weakness of this group was that they were overthinking the project and overcomplicating what was required making it appear to be more difficult than it actually was. This sometimes resulted in group meetings where there would be a bit of arguing as whenever an idea was put across, it would often be followed up with a “What if…” question resulting in more confusion with the specification and the tasks. There were times were certain diagrams were constantly being chopped and changed resulting in other pieces of work having to be changed meaning more checks had to be carried out to ensure there were no contradictions in the coursework. Despite all of this, we were all willing to put in the hours required to complete this coursework and get the best mark possible.