COMP1752 Object-Oriented Programming: Jukebox Simulation Report

Title Page

- Course Title: COMP1752 Object-Oriented Programming
- Assignment: Jukebox Simulation
- Duong Thanh Dat ID
- Submission Date

Table of Contents

1. Introduction

This report include analysis of coursework functionality and development stages of this project. The jukebox simulation provide view tracks functionality which can find and print track information on screen. This project will improve jukebox simulation with few more features such as create tracks playlist and update tracks information. This report follow the chronological order of the development from design, coding, testing, conclusion and innovation.

2. Design and Development

The track player and view track have the exact same layout as original program but wheat colour background, black text colour, dark orange button background and white foreground as well as the other windows.
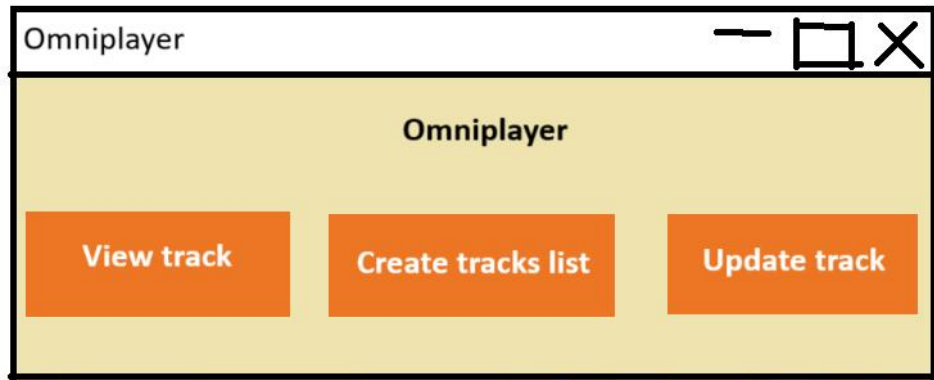
Image 1 Sketch of track player GUI

The create tracks list layout, which have an input entry to get the track number, one button on the same row as the entry to add the track(if exist) to playlist, another button below to reset list or delete all track from list which have red background, a big scroll text to display all track's basic information such as name, artist and play count, at the same row, there is a play button to play all track, in this case it only increase play count of track in playlist.
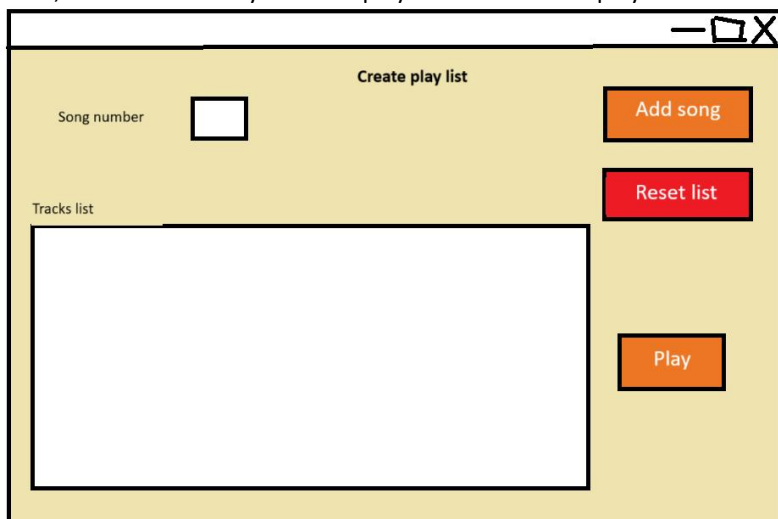

Image 2

Update tracks window, this window have two entries of track number and new rating, update button on the same row to update track rating, change window size display new track information on text box.
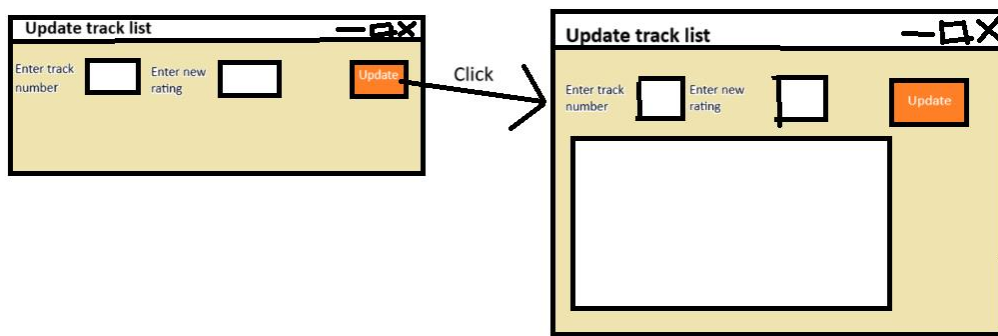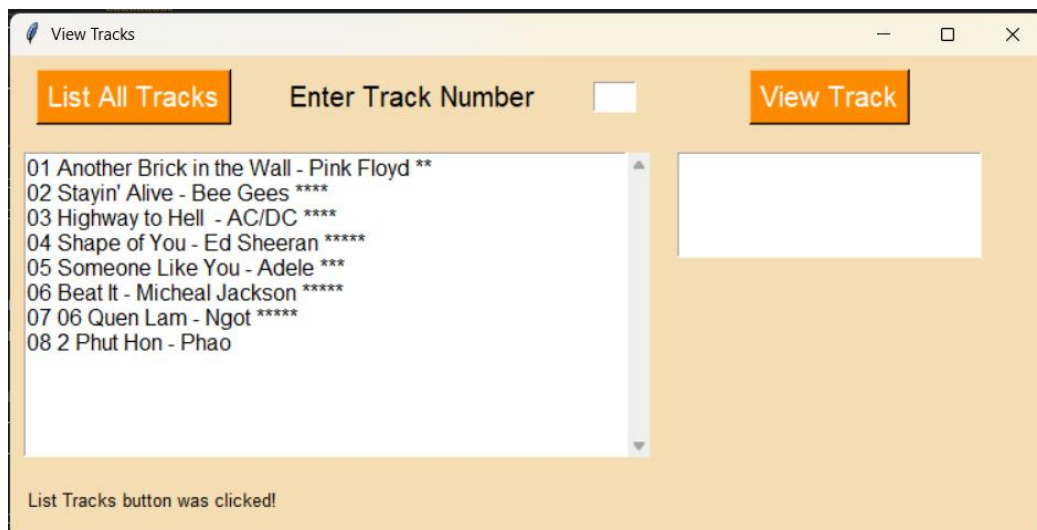
Image 3

Pic 3. Update tracks window which can change after update button clicked

- Screenshots of program in operation
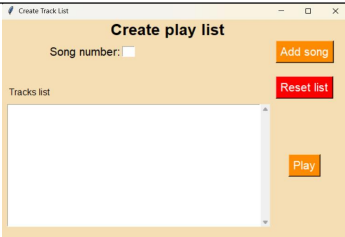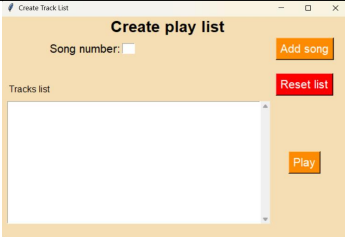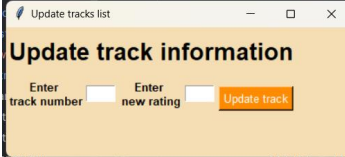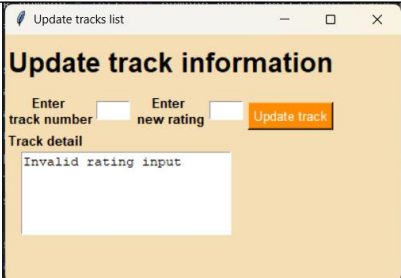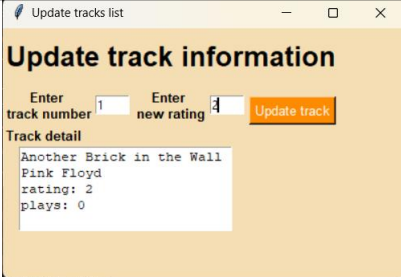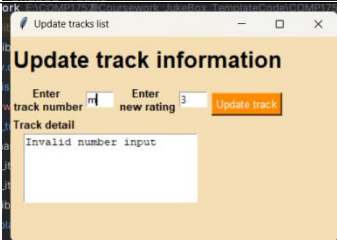
3. Testing and Faults

- Testing methodology
- Summary of test results

Testing subject	Input	Expected output	Actual output	Test result
Track player GUI	None	GUI looks like Image 1	Image 1. Trackplayer window	Pass
Create tracks list button on track player GUI	Create tracks list button	Create tracks list window open		Pass
Create tracks list GUI	None			Pass
Update track GUI before update button	None			Pass
Update track GUI after update button	Any number or None, None number or none or number lesser than 0 or greater than 5,	With invalid rating input on text box		Pass
Update track GUI after update button	1,2	With track number 1 name, artist, rating: 2, play: playcount on text box		Pass
Update track GUI after update button	Non number or None, number from 1 to 5	With invalid number input on text box		Pass

- Discussion of encountered issues

    - Resolved faults and corrections made
    - Unresolved issues and limitations
    - Validation implementation

- Unit testing approach and results

4. Conclusions, Further Development, and Reflection

- Summary of the program achievements
- Evaluation against original requirements
- Further development possibilities (what you would do with additional time)
- Reflection (choose one):

    - Option A: Achievements, difficulties faced and why, straightforward aspects and why
    - Option B: Personal development, knowledge and skills gained, long-term value

5. Innovations

- Innovation 1: [Name of innovation]

    - Description and rationale
    - Implementation details
    - Benefits and limitations

- Innovation 2: [Name of innovation]

    - Description and rationale
    - Implementation details
    - Benefits and limitations

Appendices

Appendix A: Commented Code (Stage 1)

- Complete commented version of view_tracks.py

Appendix B: Test Table and Results

- Input values
- Actions performed
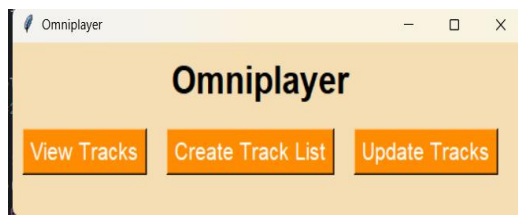
- Expected outputs
- Actual outputs

Image 4. Trackplayer window

- Pass/fail status

Appendix C: Full Source Code

- All Python files with brief descriptions