OP Music

Group Members: Charley Wu, Dennis Contreras, Amani Mohammed Alkhalifa

Concept

- A java application that allows users to interact with a server and rate songs (1-10) from the database that is behind the server
- Each user gets to keep a profile of songs that they have rated and people can see each other's lists of songs, what score they rate for each song, and what songs they know in common
- On the main page, a ranking system is displayed based on the cumulative scores of the songs

★★★★★ Excellent
★★★★★ Above Average
★★★★★ Average
★★★★★ Below Average
★★★★★ Poor

Outside Courses & Topics

- CS 103
  - Functions and classes
- CS 104
  - Data structures & IO
- CS 109
  - Network, OOP, Encapsulation, and abstraction
- CS 170
  - Ranking algorithm and searching based on CS 170
- CS 356
  - Debugger/debugging experience

Skills and Tools

Software / Purpose

- Codeshare - Simultaneous editing within a coding environment (collaboration, Ideation)
- Eclipse - Compiling / testing
- MySQL - Permanent Data Storage and handling
- Google Drive - Design Documents collaboration, responsibility delegation
- Gmail - Communication, software emergency- troubleshooting
- Google Slides - Presentation
- Discord - Live instant communication, quick share of code and quick revision
- Notepad++ - Editing outside of core project code, new ideas, quick revisions

Design and Development Decisions (Part 1)

Ideation Phase and Scope:

What Worked:
Ideation: The team had great ideas for the concept.
The core vision was communicated effectively.

We wanted to multiple users to be able to rank their favorite music and share their opinion about music with others. (profiles and personal scores)

We wanted users to be able to see what everyone thought about the music. (the overall score)

What Did Not Work:
-Being able to add tags to individual songs that allow the users to search for songs based on genres and features.

- Fully Functional GUI with navigation bar,
 And pretty format

Reasons: Not enough time

Design and Development Decisions (Part 2)

Data Design:

What Worked:

The important data could be stored and to a certain degree "handled" by the database system MySQL

What Did Not Work:
-Being able to add tags to individual songs that allow the users to search for songs based on genres and features.

- Fully Functional GUI with navigation bar,
  And pretty format

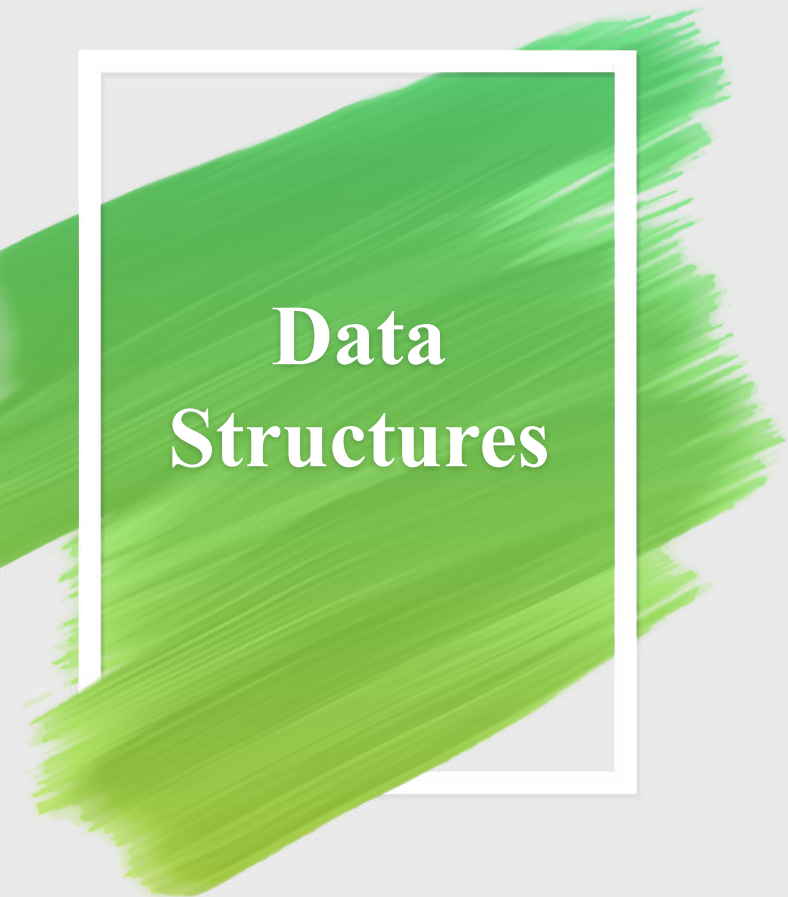Reasons: Not enough time

Teamwork

Charley:
- Concepts
- Code Design & Writing
  - Establishing guest-server connection
  - Login & register functionalities
- Testing

Dennis:
- Concepts
- Code Design & Writing
  - Display functionalities
  - Interacting with database (adding and calculating scores)
- Testing

Amani:
- Code suggestions

Data Structures

- **Arrays** for parsing user input
- **List** for keeping the server threads
- **ResultSet** to catch output from query
- Database
  - **Lists** and **queues** for SQL parsing
  - **Trees** for SQL query execution plan
  - **Database tables** to store data

Login Functionality

- After connecting to the server, the guest will be presented with a variety of options to do, one of which is to login
- To login, the user will type "login username password"
- Username and password will be verified by the records in our database

```
Welcome to MyMusicList
Music Browser:
a        7
f        6
b        5
e        5
d        4
c        3


What would you like to do next:
                Type: login username password, to login in
                Type: register username password, to register
                Type: look_up username, to look up other people's profile
Once you are logged in, you can:
                Type: my_profile, to view your profile
                Type: add_score songname score, to give your score to a song and add it to your profile
                Type: music_Library, to view our entire music catalouge
```

Guests & Users

Guests
- Browsing the ranking list
- Looking up other people's profile

Users
- Add scores to songs
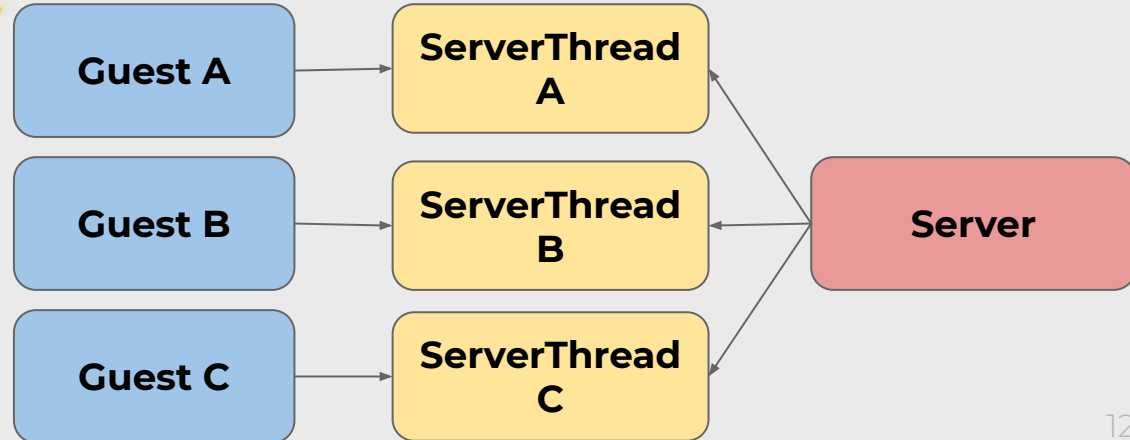- Keeping a profile of all the songs scored

Network Programming

- Guests are connected to a server
- Guests can write requests to the server
- The server accept requests from the guests and process them

| Guest A | Connection → | Server |

Multi Threading

- Guests and Server both extend Threads
- Multiple guests can be connected to the server and sending server requests at the same time

Thank you!