Group Project

GroupX

12.11.2020

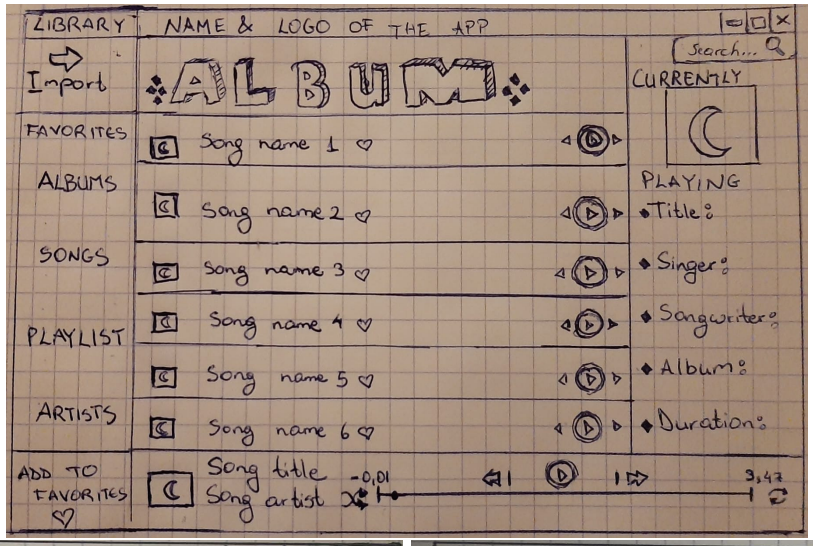
Group Composition

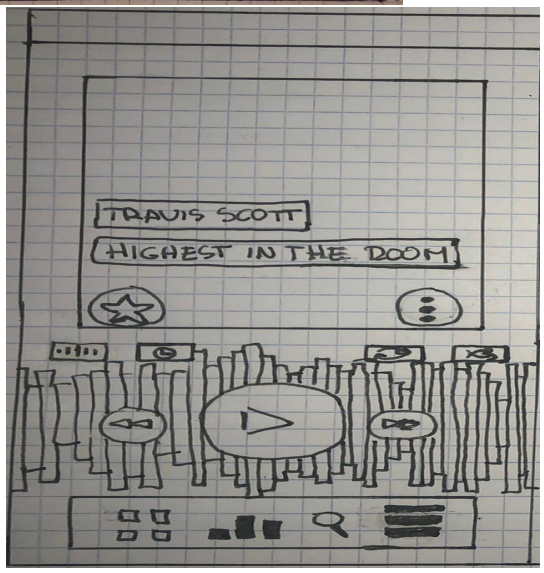
Leader: Gerald Prendi
Members:

- Paola Guma
- Ramazan Tafa
- Lidia Morariu
- Gerald Prendi

IDEA: Music(Media) Player

Design:





Abstract

The program is a music playback program written in Advanced Student Language. It can play *most audio files*. Allows you to import audio files and make playlists containing them.
It displays information regarding the song that is being played, like:

- singer
- author
- duration
- album name
- album cover etc..

Possible to search not only songs, but also for playlists, artists or albums using the search bar.Sorting is another option, where you can choose from A to Z or reverse.
You can add a song to "Favorites". Seek playback by clicking on the progress bar.

Functionality

- import audio files into the program
- playback the audio files
- playback seek by clicking the progress bar
- create and edit playlists
- add songs to "favorites"
- change the theme of the program
- shuffle the songs in the current playlist
- loop through the playlist, or a single song
- search for songs, artists, albums or playlis
- skip the current song or go back to the previous song in the playlist
- sort the songs and/or playlists

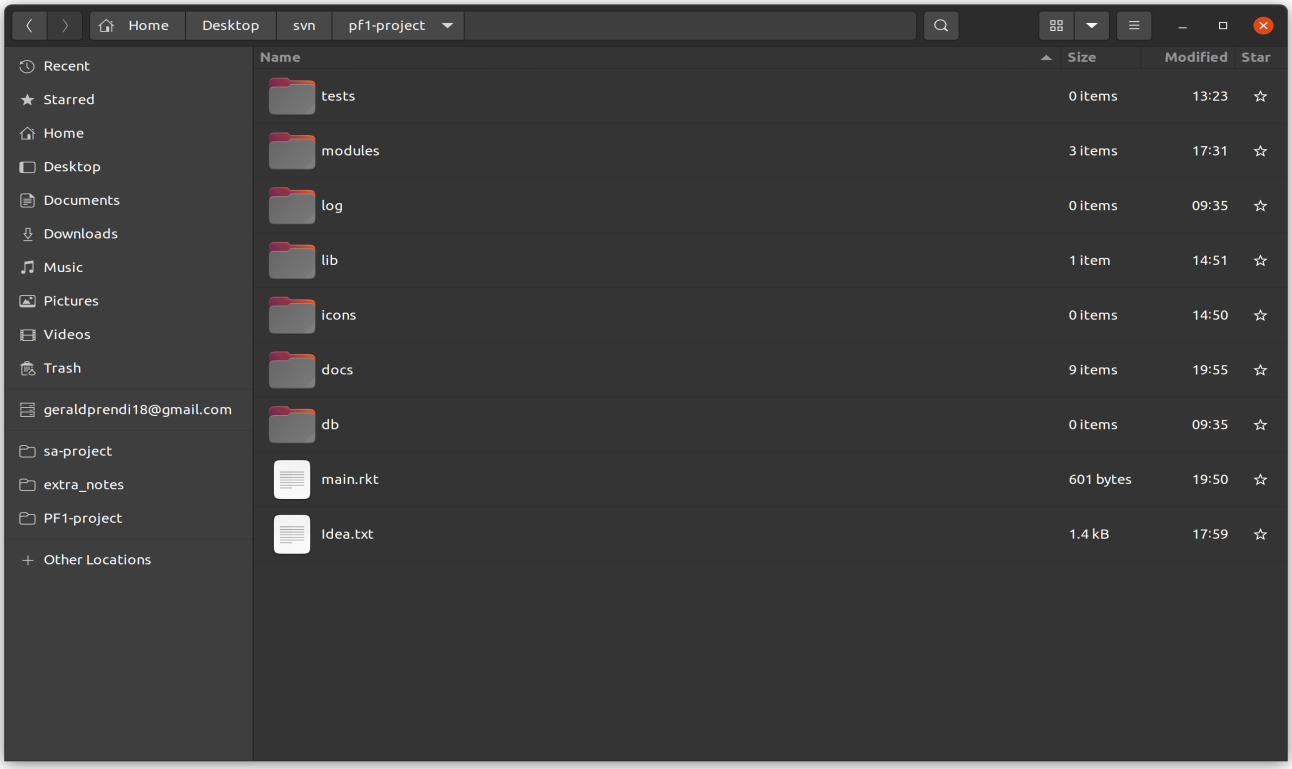
Infrastructure

The work will be adjusted between:
- Microsoft Teams,
- Whatsapp,
- Notion(for note-taking),
- Versioning System: SubVersioN(svn)

Repository

We thought it was better, that the work Repository to be a multi file/folder project, in order for everybody to work in parallel.

Explanation: **'main.rkt'** is the main executable file, everything will come down to main(It will one have the implementation of big-bang)
'lib' folder is the folder where libraries will be installed, we intend to download and put libraries we usually require or install using 'raco' in there.
'modules' folder will have files: 'modules' we created ourselves and will link them to main
'docs' will have the documentation
'db' will host the info files, in form of database(we will not implement a database, just plain text files)
'icons' is a folder for icons needed
'logs' will have error messages or known bugs, that will be fixed
'test' folder will have test files, for each main modules



Delivery:

The project will be delivered in two ways:
- an executable file
- source code

Plan of Work

Steps:

1. First we concept and create a basic idea, with functionalities
2. Structure ideas and start visualising the program and work flow
3. Think of Data Structures, that will be used
4. Work on the backend and think of ideas to handle under the hood work, meanwhile two Members work on design
5. The design will be ready, and the backend will be ready too
6. Group gathers and start building the GUI(part by part as LEGO bricks)
7. Connect GUI with backend
8. Test and Debug
9. Deliver the project, continue maintaining

Implementation:

Requirements :

- Modules:

  racket/system

  htdp/batch-io

  2htdp/image

  2htpd/universe

- Language:

  Advanced Student Language (ASL)

- *Note during the work, we might come up different modules, will consult TA's if we can use them or not

Data Structures:

- List
- Struct
- Atomic Data Structures
- hashes : mostly on the 'Database'

Features:

- String Comparison(and Sort)
- Implementation of Quick Sort
- Input/Output in/from files
- Modules(created and called by us)
- Printing Contents of a directory
- Creating registries with hashes(playlists and songs)

**Note :

The whole group helped in the making of this document:
- the design was done by Paola and Lidia as well as spell check and correction.
- abstract and featuring was done by Ramazan.
We as a group, came together(while respecting the social distancing and other rules) and went into detail about the implementation.
If anything that is listed in the previous parts of the document is not allowed or against the rules, please inform us.
Is it possible to use racket/gui , it would help us speed up the process of designing the GUI of the program?
If you have any questions, feel free to contact us.
Contact:
Gerald Prendi
Email: gerald.prendi@usi.ch