**План за изпълнение на проекта по СД**

1. Функционалности

//“Coding to interfaces, not implementation.”

* + Първоначално
    - „help“ меню
    - вход
    - регистрация
  + След успешен вход
    - промяна на данни
    - добавяне на песен
    - задаване на рейтинг
    - генериране на плейлист
    - записване на плейлист под име
    - зареждане на плейлист по име
    - информация за всички песни //на зареден плейлист

1. Изготвяне на класове

//може да направя sette-ите и getter-ите protected и само friend функциите да имат достъп

//така не се нарушава енкапсулацията

* class **User**
* string username
* string password
* string full\_name
* string birth\_date
* vector<string> fav\_genres
* vector<*Playlist*> playlists//обмисли промяна на типа
* bool is\_logged //проверява дали потребителят е в системата
* class **Song**
  + string name
  + string artist
  + string genre
  + Album album
  + int release\_date
  + double rating
* class **Album**
  + string artist
  + vector<*Song*> songs //обмисли промяна на типа
  + int release\_year
* class **Playlist** //може да сложа дата на създаване
  + string name
  + vector<*Song*> songs //обмисли промяна на типа
  + bool is\_loaded
* class **Library**
  + B-Tree<*User*> users
  + B-Tree <*Song*> songs
  + B-Tree <*Playlist*> playlists
  + B-Tree <*Album*> albums //????

1. Основни структури от данни и модели

// видове, информация и причина за избор

* + **B-Tree** //данни в class Library

//трябва още информация за B+, B\* trees !!!!!!!!!!!!!!

* + - *Самобалансиращо* се *search tree*
    - „дебело“ дърво
    - minimum degree ‘t’ (t depends upon disk block size)
    - Възли
      * ***min*** (t-1)/2 keys
      * ***max*** (t-1) keys
      * children = (keys + 1)
      * ***root:***  min 1 key
      * all keys: *↑ sorted*
      * all ***leaves*** = same level
    - search/insert/delete O (log n)

* + - *Предимства*:
      * Подходящо за големи количества данни

(прекалени за main memory; съхраняват се на disk)

* + - * Бърза работа с данните

(constant O(log n) time complexity)

* ↑ disk access time (compared to main memory access)
* ↑ number of keys & ↓ height
  + data-reading from disk in form of *blocks*
  + reduced disk access
* generally: tree size = disk block size
  + - * Запазване на данните на disk

(в постоянната памет)

Улеснено зареждане при повторно отваряне на програмата

* + Работа с файлове
    - Съхранение в txt/bin файл в хард диска
    - // ?по какъв начин ще осъществявам сериализация и десериализация
      * ((())()())
      * Матрица
      * По-подходящ вариант
    - Дърветата ще са от тип клас
    - По-умна идея: съхранение във файл (подобен на) dot от лекциите
      * (aka подходящ и приспособен за дървета)

1. Класове – *методи*

//

* **User**
* **Song**
* **Album**
* **Playlist**
* **Library**
  + register()
  + login()
    - in>> username
    - in>> password

// moje da dobavq „0“ za vryshtane nazad

// primerno (prazen )enter za krai na komandata

//moje da gi napravq kato spisyk ot komandi i da se vyvejda ot 1-10 (schneller)

((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

🏶 **help**

⇨ List of commands:

1. .........
2. ..........
3. .........
4. .......
5. .......

🏶 **login**

⇨ Enter username:

🏶 .........

⇨ Enter password:

🏶 \*\*\*\*\*\*

👍⇨ success

👎⇨ Wrong data (un/pw)

⇨ 1. (login)

⇨ 2. register

🏶 **register**

⇨ Enter username:

🏶 .........

⇨ Enter password:

🏶 \*\*\*\*\*\*

⇨ Repeat password:

🏶 \*\*\*\*\*\*

👍⇨ success

👎⇨ (register)

🏶 **change data**

⇨ choose between:

1. username
2. password
3. full name
4. birth date
5. fav genres
   1. add
   2. remove

🏶 ....

👍⇨ success👎⇨ (choose between)

🏶 **add song**

⇨ \*enter name

⇨ \*enter artist

⇨ \*enter genre

⇨ \*enter album

⇨ \*enter release date

🏶....

👍⇨ success 👎⇨(add song)

🏶 **rate song**

⇨ enter name

🏶

👍⇨ enter rate

🏶

👍⇨ success! 👎⇨ (enter rate)

👎⇨(enter name)

🏶 **generate playlist** // кой определя макс размера? – аз

⇨ choose criteria:

1. rating >
2. included genres
3. excluded genres
4. only fav genres
5. before year
6. from year
7. after year

🏶 .... //може повече от една; по сила

🏶 **save playlist** //същото като change playlist`s name

⇨ choose name:

🏶 .....

👍⇨ success! 👎⇨ (choose name)

🏶 **load playlist**

⇨ enter name:

🏶 ....

👍⇨ success! 👎⇨ (enter name)

🏶 **change playlist`s name**

⇨ enter name:

🏶 ....

👍⇨ success! 👎⇨ (enter name)

🏶 **show info 4 all songs**

👍⇨ ….👎⇨ (load playlist)