

Техническа спецификация на системата за избор на песен

1. Въведение

Системата за избор на песен е предназначена за съхранение и управление на информация за музикални композиции, като позволява на потребителите ефективно да добавят, редактират и търсят песни според различни критерии. Целта е да се осигури интуитивен и отзивчив потребителски интерфейс за лесно разглеждане на музикални колекции.

2. Функционални изисквания

2.1 Добавяне на информация за песен

Системата трябва да позволява на потребителите да добавят нова песен със следните характеристики:

- Заглавие
- Изпълнител (певец/група)
- Времетраене (дължина на възпроизвеждане)
- Година на издаване
- Жанр
- Допълнителна информация (незадължителна)
- Допълнителни тагове (незадължителни)

2.2 Търсене и филтриране на песни

Потребителите трябва да могат да търсят песни по следните критерии:

- Заглавие
- Изпълнител
- Година на издаване
- Жанр

2.3 Управление на списъка с песни

- Изтриване и редактиране на информация за песни
- Преглед на пълния списък с песни
- Опции за сортиране (по година, времетраене и други параметри)

2.4 Изтриване на информация за песен!

Системата трябва да позволява на потребителите да изтриват информация за песен от системата

3. Нефункционални изисквания

- **Интерфейс:** Трябва да бъде удобен и интуитивен за потребителя.
 - **База данни:** Информацията трябва да се съхранява в релационна база данни (MySQL).
 - **Производителност:** Търсенето трябва да се извършва бързо, дори при голям брой записи.
 - **Мащабируемост:** Функционалността трябва да може да бъде разширявана в бъдеще.
-

4. Архитектура и технологичен стек

- **Backend:** Java Spring Boot
- **Frontend:** React + TypeScript
- **База данни:** MySQL
- **API:** REST API за взаимодействие между frontend и backend

5. Взаимодействие с потребителя (Use Case)

5.1. Добавяне на песен

- **Описание:** Потребителят въвежда информация за нова песен.
 - **Предусловия:** Потребителят трябва да е влязъл в системата.
 - **Основен поток:**
 1. Потребителят натиска бутона "Добави песен".
 2. Появява се форма за въвеждане на данни: заглавие, изпълнител, дължина, година, жанр и други.
 3. Потребителят попълва задължителните полета.
 4. Системата валидира въведените данни.
 5. При успех, песента се запазва в базата от данни.
 6. Показва се съобщение за успешно добавяне.
 - **Алтернативен поток:**
 - Ако липсва задължителна информация, системата показва грешка и не запазва песента.
 - Ако вече съществува дубликат (същото заглавие + изпълнител), системата показва предупреждение.
-

5.2. Търсене на песен

- **Описание:** Потребителят търси песни според избрани критерии.
 - **Основен поток:**
 1. Потребителят отваря секцията за търсене.
 2. Избира един или повече филтри (заглавие, изпълнител, жанр, година).
 3. Натиска бутона "Търси".
 4. Системата показва списък с песни, които отговарят на филтрите.
 - **Алтернативен поток:**
 - Ако не е въведен филтър, се показват всички песни.
 - Ако няма резултати, се показва съобщение „Няма намерени песни“.
-

5.3. Редактиране на песен

- **Описание:** Позволява промяна на информация за вече съществуваща песен.

- **Предусловия:** Потребителят трябва да е оторизиран.
 - **Основен поток:**
 1. Потребителят избира песен от списъка.
 2. Натиска "Редактирай".
 3. Появява се форма с текущата информация.
 4. Потребителят прави промени и натиска "Запази".
 5. Системата валидира и актуализира записа.
 6. Показва се потвърждение.
 - **Алтернативен поток:**
 - Ако се въведат невалидни данни (напр. година в бъдещето), системата показва грешка.
-

5.4. Изтриване на песен

- **Описание:** Премахване на запис от базата данни.
 - **Предусловия:** Потребителят трябва да е оторизиран.
 - **Основен поток:**
 1. Потребителят избира песен от списъка.
 2. Натиска "Изтрий".
 3. Системата изисква потвърждение.
 4. При потвърждение, песента се изтрива от базата.
 5. Появява се съобщение за успешно изтриване.
 - **Алтернативен поток:**
 - Ако потребителят откаже, изтриването се отменя.
-

6. Разработка и тестване

- Кодът трябва да бъде покрит с модулни (unit) тестове.
 - Трябва да се извърши тестване на потребителския интерфейс и API преди разгръщане.
 - Финалната версия не трябва да съдържа грешки, които засягат основната функционалност.
-

7. Разгръщане и поддръжка

- Кодът ще бъде хостван в GitHub.

- Базата данни ще бъде разположена на отдалечен сървър или в облачно хранилище.
- Системата трябва да бъде поддържана с възможност за обновления и корекции на грешки.