Приложение “Кинотеатр”

**Объекты**

В кинотеатре представлены следующие основные объекты : **пользователь, фильм и билет.**

* **Пользователь** (id, логин, пароль, уровень пользователя)  
  Может быть три уровня пользователя (у каждого свой уровень доступа).  
  1) **Простой пользователь** может:  
   - просматривать мероприятия (фильмы)   
   - покупать и возвращать билеты  
   - просматривать купленные билеты  
  2) **Менеджер** может:  
   - редактировать мероприятия (фильмы)  
   - покупать и возвращать билеты определенного пользователя  
  3) **Админ** может:  
   - удалять и изменять(создавать) пользователей   
   - удалять и изменять(создавать) мероприятия  
  *Пользователь должен регистрироваться сам.   
  Админ и менеджер добавляется изначально и не создается через форму регистрации.*
* **Билет** (id, пользователь, фильм, номер места, цена, флаг на куплен или нет)   
  *Если место не куплено, тогда значение пользователя должно быть пустым.*
* **Фильм** (id, название, число + время, список билетов)

*Другие объекты создаем по надобности.*

**Хранение**

Есть 2 варианта хранения информации:

* **База Данных** (MySQL, MS SQL, PostgreSQL, SQLite...)
* **Набор файлов**

1) В первом случае необходимо установить и настроить БД. Настроить подключение к Java приложению. Создать необходимые таблицы в БД и настроить их зависимости (диаграммы).   
После данной настройки можно приступать к написанию JDBC коннектора в нашем приложении, с помощью которого впоследствии будут заполняться таблицы БД.

2) Во втором случае создаем отдельный файл для каждого вида хранимой информации.   
После создания необходимого кол-ва файлов описываем сервисный класс который будет записывать информацию в файл определенным образом с такой структурой чтобы впоследствии ее можно было с легкостью парсить с помощью RegEx. Соответственно создаем и сервисный класс для чтения данных файлов.

*Рекомендую сначала настроить БД/Структуру файлов, а только потом приступать к написанию проекта.  
Плюс будет проще сначала создать часть данных таблиц/файлов, научиться работать с ними и после этого расширяться.*

**Дополнительные функции**

**Лог действий пользователя**  
Каждый шаг пользователя должен регистрироваться. Записывать дату и время входа.  
Т.е. запись всех действий конкретного пользователя сразу после регистрации.  
Запись построчно в файл с именем логина пользователя (pupkin.txt).  
Пример лога:

Пользователь Pupkin зарегистрировался.  
 Дата и время входа: 12.12.19 / 18:30  
 Открыто главное меню.  
 Открыто меню выбора фильма.  
 Выбран фильм “Елки 284”.  
 Выбрано место 1.  
 Пользователь совершил покупку билета на фильм “Елки 284”.  
 Пользователь отменил покупку билета на фильм “Елки 284”.  
 Открыто меню выбора фильма.  
 Выбран фильм “Звездные Воины”.  
 Выбрано место 25.  
 Пользователь совершил покупку билета на фильм “Звездные Воины”.  
 Открыто меню просмотра купленных билетов.  
 Пользователь завершил работу с приложением.  
 Пользователь Pupkin вошел в приложение.  
 Дата и время входа: 13.12.19 / 12:46  
 Открыто главное меню.  
 Открыто меню просмотра купленных билетов.  
 Пользователь завершил работу с приложением.  
  
*Писать лог и для простых пользователей и для менеджеров/админов.*

**Дополнительные условия**

* В приложении сделать три слоя ( 1 - для меню, 2 - сервисный + работа с базой/файлами, 3 - бины)   
   - Слой бинов должен только хранить бизнес-объекты (User, Cinema, Ticket ...).  
   - Слой сервисов должен содержать сервисные классы для работы с данными хранимыми в бинах или же в базе/файлах. (Например UserService будет иметь методы createUser(...), updateUser(...), deleteUser(...), writeUserToDB(User user) и т.д.)  
  Данный слой **не хранит** информацию, только **обрабатывает**.  
   - Слой консольного меню должен выводить информацию на экран, уметь получать информацию с экрана.  
  При этом консольный слой не должен производить обработку информации, для этого должны вызываться специальные методы из сервисного слоя.
* Консольное меню на русском или английском языке на выбор.
* Пароль в приложении хешируем (используем кодировку). Чтобы в базе не было конкретных паролей пользователей. (Ссылка на пример с описанием <https://www.codeflow.site/ru/article/java-password-hashing>)
* Приложение можно дополнять своим функционалом, но логично!!!
* ООП и полиморфизм используем по полной.

**Уровни реализации приложения**

Т.к. приложение достаточно большое и обладает широким функционалом логично разделить реализацию приложения на несколько уровней.

1. **Уровень первый**. В данной реализации требуется иметь:
   1. БД/Файлы хранения
   2. Консольное меню
   3. Простого пользователя
   4. Хотя бы один фильм на который можно купить/вернуть билеты и их просмотреть.
2. **Уровень второй**. В данной реализации требуется иметь:
   1. БД/Файлы хранения
   2. Консольное меню
   3. Простых пользователей + менеджера
   4. Фильмы на которые можно купить/вернуть билеты и их просмотреть.
3. **Уровень третий**. В данной реализации требуется иметь:
   1. БД/Файлы хранения
   2. Консольное меню
   3. Простых пользователей + менеджера + админа
   4. Фильмы на которые можно купить/вернуть билеты и их просмотреть.
   5. Лог действий
   6. Хэширование паролей.

Разработку приложения рекомендую производить по этим же уровням чтобы в любой момент выполнения у вас имелось готовое приложение с тем или иным функционалом (хотя бы минимальным).