SampleController

1. Съдържа 2 конструктора: експлицитен, подразбиращ се
2. Има 3 функции вътре
3. userLogin

* Проверява дали въведените username и парола са правилни, ако са допуска потребителя до съответния home page.
* Проверява дали някое от полетата е празно, ако някое от тях е празно изкарва съобщение. Ако и двете полета са попълнени, създава UserListModel съдържащ текста в полетата. Извиква се функцията UserLogin от UserService. В UserLogin се взимат всички съществуващи до момента потребители, с помощта на два if statement-а проверяваме дали съществува потребител със същите име и парола. Ако съществува такъв, проверяваме дали статуса му е not pending, ако и това условие е изпълнено функцията връща true.
* Ако съществува потребителя, с помощта на функцията FindUserType разбираме какъв тип е потребителя, като 1 отговаря за user, 2 отговаря за operator, 3 отговаря за админ.
* (USER) Ако потребителя е user се зарежда userHomePage, след което се прави проверка дали потребителя е върнал книгите си на време, при просрочване на срока за връщане, статуса на потребителя става “bad”, докато не върне своите книги.
* (OPERATOR) Ако потребителя е Operator, се прави същата проверка като user, но също се проверява дали има формуляри за одобряване, ако има такива, излиза нотификация към оператора.
* (ADMIN) Ако потребителя е Admin, се правят същите проверки като при user и operator. Прави се проверка дали има стари книги, които не са архивирани, при наличие на такива, показва нотификация.
* При всеки вид потребител извиква функцията displayId от съответният контролер.

Тя се ползва, да укаже кои потребител е логнат във всеки даден момент.

1. goToRegistration

* Закарва ни на страницата за регистриране.

1. cancelButtonOnAction

* Затваря прозореца.

RegistrationController

1. Съдържа 2 конструктора: експлицитен, подразбиращ се

2. Има 2 функции вътре

1. addUser

* Взима информацията попълнена в полетата. Прави проверка дали някое от полетата е празно, ако да (if statement), излиза съобщение.
* При цялостно попълнени полета, се създава UserListModel и се проверява дали съществува потребител със еднакви данни. Ако не съществува такъв потребител, той бива създаден.
* Ако се създаде потребител, създаваме UserInfoListModel с помощта на който запазваме userInfo-то на потребителя ни. Също се създава формуляр съдържащ информацията за потребителя, за бъдещо одобрение.
* Докато не е одобрен формуляра от регистрацията ни, не можем да се логнем

b. goBack

* Отваря логин страницата - sampleController

UserHomePageController

1. Съдържа 2 конструктора: експлицитен, подразбиращ се

2. Има 7 функции вътре

1. displayId

* Променя текста на label в страницата спрямо зададеното id на потребител

1. Test

* Изкарва съобщение със съдържание – параметъра на функцията.

1. LendBook

* Отваря страница за заемане на книги и извиква функцията displayId за съответната страница.

1. ReturnBook

* Отваря страница за връщане на книги и извиква функцията displayId за съответната страница.

1. SearchBook

* Отваря страница за търсене на книги и извиква функцията displayId за съответната страница.

1. UserRatings

* Отваря страница за рейтинга на потребителите и извиква функцията displayId за съответната страница.

1. Logout

* Закарва ни на логин страницата.

LendingBooksController

1. Съдържа 2 конструктора: експлицитен, подразбиращ се

2. Има 8 функции

1. displayId

* Променя текста на label в страницата спрямо зададеното

1. Showinfo

* Показва съобщение, което обяснява как да вземеш книги.

1. CheckBoxes

* Когато натиснем един от check box-овете, другия се deselect-ва.

1. showBooks

* Задейства се спрямо кой check box е избран, като показва достъпните книги, спрямо дали са в читалнята или не

1. getRow

* Взима селектирания ред от таблицата

1. addCopyToLendList

* Взима избраното копие и го премахва от таблицата – достъпни книги и го мести в таблицата – книги за заемане, като прави check box-овете недостъпни. По този начин се осигуряваме че не могат да се изкарват книги извън читалнята.

1. lendCopies

* Проверява кой от двата check box-а е избран, след което взима всички книги в таблицата за заемане и запазва lendInfo за всяка една от книгите и създава lendBook за заемането

1. goToHomePage

* закарва ни на userHomePage

ReturnBookController

1. Съдържа 2 конструктора: експлицитен, подразбиращ се

2. Има 5 функции

1. displayId

* Променя текста на label в страницата спрямо зададеното

1. Showbooks

* Взима потребителя които в момента е логнат с помощта на displayId, след което проверява дали потребителя е заел някакви книги, ако е заел – ги изкарва на екрана, ако не е – изкарва съобщение

1. ReturnBook

* Взима селектираната във момента книга, изтрива за нея lendInfo-то, прави ексемпяра available, след което прави проверка дали има останали книги за връщане, ако има, показва само тях, ако няма, изтрива lendBooks-а за потребителя

1. getRow

* Взима селектирания ред от таблицата

1. goToHomePage

* закарва ни на userHomePage

SearchBooksController

1. Съдържа 2 конструктора: експлицитен, подразбиращ се

2. Има 5 функции

1. displayId

* Променя текста на label в страницата спрямо зададеното

1. initialize

* При стартиране страницата, изкарва всички ексемпляри в таблицата.

1. searchBooks

* Прави проверка дали някой от радио бутоните е селектиран, ако е – извежда всички книги които отговарят по име и състояние. Ако няма ексемпляри които да отговарят на търсенето, извежда съобщение. Ако нито един от радио бутоните не е селектиран, проверява дали има ексемпляри със такова име.

1. getRow

* Взима селектирания ред от таблицата

1. goToHomePage

* закарва ни на userHomePage

UserRatingController

1. Съдържа 2 конструктора: експлицитен, подразбиращ се

2. Има 6 функции

1. displayId

* Променя текста на label в страницата спрямо зададеното

1. initialize

* При стартиране страницата, изкарва всички потребители в таблицата.

1. Loyal

* Изкарва всички потребители със добър рейтинг, ако няма такива – изкарва съобщение.

1. Disloyal

* Изкарва всички потребители със лош рейтинг, ако няма такива – изкарва съобщение.

1. getRow

* Взима селектирания ред от таблицата

1. goToHomePage

* закарва ни на userHomePage

UserHomePageController

1. Съдържа 2 конструктора: експлицитен, подразбиращ се

2. Има 11 функции вътре

1. displayId

* Променя текста на label в страницата спрямо зададеното id на потребител

1. Test

* Изкарва съобщение със съдържание – параметъра на функцията.

1. LendBook

* Отваря страница за заемане на книги и извиква функцията displayId за съответната страница.

1. ReturnBook

* Отваря страница за връщане на книги и извиква функцията displayId за съответната страница.

1. SearchBook

* Отваря страница за търсене на книги и извиква функцията displayId за съответната страница.

1. UserRatings

* Отваря страница за рейтинга на потребителите и извиква функцията displayId за съответната страница.

1. Logout

* Закарва ни на логин страницата.

1. createUserPage

* Отваря страница за одобряване на заявки за регистрация и извиква функцията displayId за съответната страница.

1. deleteUserPage

* Отваря страница за изтриване на профил на потребител и извиква функцията displayId за съответната страница.

1. formsPage

* Отваря страницата за справки за формуляри и извиква функцията displayId за съответната страница.

1. userInfo

* Отваря страницата за справки за потребители и извиква функцията displayId за съответната страница.

UserRatingController

1. Съдържа 2 конструктора: експлицитен, подразбиращ се

2. Има 5 функции

1. displayId

* Променя текста на label в страницата спрямо зададеното

1. initialize

* При стартиране на страницата, запълва combo box-а със имената на потребителите които са подали регистрационни заявки.

1. parseDate

* Превръща подадения стринг във дата.

1. createUser

* Взима селектирания от combo box-а потребител и го одобрява, ако няма селектиран потребител – изкарва съобщение.

1. goToHomePage

* закарва ни на operatorHomePage

DeleteUserController

1. Съдържа 2 конструктора: експлицитен, подразбиращ се

2. Има 4 функции

1. displayId

* Променя текста на label в страницата спрямо зададеното

1. initialize

* При стартиране на страницата, запълва combo box-а със имената на потребителите.

1. DeleteUser

* Взима селектирания от combo box-а потребител и го изтрива, ако няма селектиран потребител – изкарва съобщение.

1. goToHomePage

* закарва ни на operatorHomePage

FormsController

1. Съдържа 2 конструктора: експлицитен, подразбиращ се

2. Има 5 функции

1. displayId

* Променя текста на label в страницата спрямо зададеното

1. Initialize

* Когато страницата стартира, взима всички формуляри които съществуват и ги изкарва в таблицата.

1. ShowPending

* Показва pending формулярите, ако няма такива изкарва съобщение.

1. ShowNotPending

* Показва not pending формулярите, ако няма такива изкарва съобщение.

1. goToHomePage

* закарва ни на operatorHomePage

SearchUserInfoController

1. Съдържа 2 конструктора: експлицитен, подразбиращ се

2. Има 4 функции

1. displayId

* Променя текста на label в страницата спрямо зададеното

1. Initialize

* При стартиране на страницата, запълва колоните със информация за потребителите в таблицата.

1. SearchUser

* Взима текста от text field-а, проверява дали има такъв потребител, ако има – изкарва информацията за потребителя и проверява дали е взел книги, ако няма или не е взел книги, изкарва съобщение .

1. goToHomePage

* закарва ни на operatorHomePage

AdminHomePageController

1. Съдържа 2 конструктора: експлицитен, подразбиращ се

2. Има 15 функции вътре

1. displayId

* Променя текста на label в страницата спрямо зададеното id на потребител

1. Test

* Изкарва съобщение със съдържание – параметъра на функцията.

1. LendBook

* Отваря страница за заемане на книги и извиква функцията displayId за съответната страница.

1. ReturnBook

* Отваря страница за връщане на книги и извиква функцията displayId за съответната страница.

1. SearchBook

* Отваря страница за търсене на книги и извиква функцията displayId за съответната страница.

1. UserRatings

* Отваря страница за рейтинга на потребителите и извиква функцията displayId за съответната страница.

1. Logout

* Закарва ни на логин страницата.

1. createUserPage

* Отваря страница за одобряване на заявки за регистрация и извиква функцията displayId за съответната страница.

1. deleteUserPage

* Отваря страница за изтриване на профил на потребител и извиква функцията displayId за съответната страница.

1. formsPage

* Отваря страницата за справки за формуляри и извиква функцията displayId за съответната страница.

1. userInfo

* Отваря страницата за справки за потребители и извиква функцията displayId за съответната страница.

1. addNewBook

* Отваря страницата за добавяне на книги в библиотеката и извиква функцията displayId за съответната страница.

1. PromoteToOperatorPage

* Отваря страницата за повишаване на потребител към оператор и извиква функцията displayId за съответната страница.

1. archiveBookPage

* Отваря страницата за архивиране стари книги и извиква функцията displayId за съответната страница.

1. ScrapBookPage

* Отваря страница за бракуване на книги и извиква функцията displayId за съответната страница.