Проведени тестове

1.Тестови случаи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер | Описание | Предварителни  изисквания | Стъпки за изпълнение | Очакван резултат |
| 1 | Тестване на функциалността влизане на пациент в системата, осъществена от AccountService. | Потребителят да е вече регистриран в системата. | 1.Инстанциране на обект от тип  AccountService  2.Извикване на функцията LoginUser.  3.Проверка дали потребителят е с роля пациент.  4.Проверка дали метода LoginUser връща true. | True |
| 2 | Тестване на функционалността неуспешен вход на пациент, осъществена от Account Service. | Потребителят не е регистриран, въведени са грешно име и парола | 1.Инстанциране на обект от тип AccountService  2.Извикване на функцията LoginUser.  4.Проверка дали метода LoginUser връща false. | False |
| 3 | Тестване на функциалността успешна регистрация на пациент, осъществена от Account Service | Въведените данни за регистрацията да са валидни | 1.Инстанциране на обект от тип AccountService  2.Извикване на функцията RegisterPatient.  4. Проверка за грешки.  5. Проверка дали потребителят е регистриран | Няма грешки,  потребителят е регистриран |
| 4 | Тестване на функциалността неуспешна регистрация на пациент, осъществена от Account Service. | Въведените данни за регистрация са невалидни, потребителят е вече регистриран | 1.Инстанциране на обект от тип AccountService  2.Извикване на функцията RegisterPatient.  4. Проверка за грешки.  5.Проверка дали потребителят е регистриран | Има грешки, потребителят не е регистриран |
| 5 | Тестване на функциалността успешна промяна на данните от профила на пациент, осъществена от Account Service | Въведените данни са валидни | 1.Инстанциране на обект от тип AccountService  2.Извикване на функцията EditPatient.  4. Проверка за грешки.  5.Проверка дали резултатът от операцията успешен | Няма грешки |
| 6 | Тестване на функциалността неуспешна промяна на данните от профила на пациент, осъществена от Account Service | Въведено е име на друг потребител | 1.Инстанциране на обект от тип AccountService  2.Извикване на функцията EditPatient.  3. Проверка за грешки.  4.Проверка дали резултатът от операцията успешен | Има грешки |
| 7 | Тестване на функциалността за успешно генериране на токен за смяна на парола на метода ForgotPassword на Account Service | Потребителят е вече регистриран | 1.Инстанциране на обект от тип AccountService  2.Извикване на функцията ForgotPassword.  3. Проверка за наличие на удостоверяващ токен за забравена парола. | Наличен е удостоверяващ токен за забравена парола |
| 8 | Тестване на функциалността за неуспешно генериране на токен за забравена на парола на метода ForgotPassword на Account Service | Потребителят не съществува | 1.Инстанциране на обект от тип AccountService  2.Извикване на функцията ForgotPassword.  3. Проверка за наличие на удостоверяващ токен за забравена парола. | Удостоверяващ токен за забравена парола е неналичен. |
| 9 | Teстване на функционалността успешно нулиране на паролата, осъществена от Account Service | Предоставени име, нова парола, удостоверяващ токен за смяна на парола. | 1.Инстанциране на обект от тип AccountService  2.Извикване на функцията ForgotPassword.  3.Извикване на функцията ResetPassword.  4. Проверка за грешки  5. Проверка дали паролата е сменена. | Няма грешки, паролата на потребителя е сменена |
| 10 | Тестване на функционалността успешна промяна на парола, осъществена от Account Service | Предоставен идентификационен номер на акаунта | 1.Инстанциране на обект от тип AccountService  2.Извикване на функцията ChangePassword.  3. Проверка за грешки  4. Проверка дали паролата е сменена. | Няма грешки, паролата на потребителя е сменена |
| 11 | Тестване на функционалността неуспешна промяна на парола, осъществена от Account Service | Предоставеният идентификационен номер на акаунта не съществува | 1.Инстанциране на обект от тип AccountService  2.Извикване на функцията ChangePassword.  3. Проверка за грешки  4. Проверка дали паролата е сменена. | Има грешки, паролата не е сменена |
| 12 | Тестване на метод GetUserRole от AccountService | Предоставен идентификационен номер на акаунта | 1.Инстанциране на обект от тип AccountService  2.Извикване на функцията GetUserRole.  3. Проверка за грешки  4. Проверка далил паролата е сменена. | Акаунтът има роля пациент. |
| 13 | Тестване на метод GetUserRole от AccountService | Предоставен идентификационен номер на акаунта | 1.Инстанциране на обект от тип AccountService  2.Извикване на функцията GetUserRole.  3. Проверка за грешки  4. Проверка далил паролата е сменена. | Акаунтът има роля доктор. |
| 14 | Тестване на функциалността успешно влизане на администратор в системата, осъществена от AdminService. | Администраторът е регистриран в системата, предоставени са протребителско име и парола. | 1.Инстанциране на обект от тип AdminService  2.Извикване на функцията LoginAdmin.  3.Проверка дали потребителят е с роля администратор.  4.Проверка дали метода LoginUser връща true. | True |
| 15 | Тестване на функциалността неуспешно влизане на администратор в системата, осъществена от AdminService. | Невалидно име или парола, потребителят е регистрирано, но не е в роля Администратор. | 1.Инстанциране на обект от тип AdminService  2.Извикване на функцията LoginAdmin.  3.Проверка дали потребителят е с роля администратор.  4.Проверка дали метода LoginUser връща false. | False |
| 16 | Тестване на функционалността успешно изтриване на потребител от системата, осъществена от AdminSerice. | Потребителят съществува. | 1.Инстанциране на обект от тип AdminService  2.Извикване на функцията DeleteUser.  3.Проверка за грешки  4.Проверка за броя на потребителите след изтриването. | Няма грешки, броя на потребителите е с един по-малък. |
| 17 | Тестване на функционалността успешно изтриване на потребител от системата, осъществена от AdminSerice. | Потребителят не съществува. | 1.Инстанциране на обект от тип AdminService  2.Извикване на функцията DeleteUser.  3.Проверка за грешки  4.Проверка за броя на потребителите след изтриването. | Има грешки, броя на потребителите в системата е не променен. |
| 18 | Тестване на функциалността успешна регистрация на доктор, осъществена от AdminService. | Въведените данни за регистрацията да са валидни | 1.Инстанциране на обект от тип AdminService  2.Извикване на функцията RegisterDoctor.  4. Проверка за грешки.  5. Проверка дали потребителят е регистриран. | Няма грешки, потребителят е регистриран. |
| 19 | Тестване на функциалността успешна регистрация на доктор, осъществена от AdminService. | Въведените данни за регистрация са невалидни, потребителят е вече регистриран | 1.Инстанциране на обект от тип AdminService  2.Извикване на функцията RegisterDoctor.  4. Проверка за грешки.  5. Проверка дали потребителят е регистриран. | Има грешки, потребителят не е регистриран |

2.Проведени тестове

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номер на тестов случай | Версия на програмата | Успешно преминаване на теста | Получен резултат, различаващ се от очаквания | Номер на задача в системата за управление на задачи | Име на студента извършил теста |
| 1 | [e5d0a6f](https://github.com/Kroxitrock/cancer-registry/commit/e5d0a6fc2fde2464f4931fb937a9f83b733abfb6) | Да | - | 10, 11 | Любомир Филипов |
| 2 | [e5d0a6f](https://github.com/Kroxitrock/cancer-registry/commit/e5d0a6fc2fde2464f4931fb937a9f83b733abfb6) | Да | - | 10, 11 | Любомир Филипов |
| 3 | [a949667](https://github.com/Kroxitrock/cancer-registry/commit/a94966716bdb0ad8fd54f11d13c18540d55d04f5) | Да | Повтарящи се грешки | 10, 11 | Любомир Филипов |
| 4 | [a949667](https://github.com/Kroxitrock/cancer-registry/commit/a94966716bdb0ad8fd54f11d13c18540d55d04f5) | Да | Повтарящи се грешки | 10, 11 | Любомир Филипов |
| 3 | [e5d0a6f](https://github.com/Kroxitrock/cancer-registry/commit/e5d0a6fc2fde2464f4931fb937a9f83b733abfb6) | Да | - |  | Любомир Филипов |
| 4 | [e5d0a6f](https://github.com/Kroxitrock/cancer-registry/commit/e5d0a6fc2fde2464f4931fb937a9f83b733abfb6) | Да | - |  | Любомир Филипов |
| 5 | [e5d0a6f](https://github.com/Kroxitrock/cancer-registry/commit/e5d0a6fc2fde2464f4931fb937a9f83b733abfb6) | Да | - | 20 | Любомир Филипов |
| 6 | [e5d0a6f](https://github.com/Kroxitrock/cancer-registry/commit/e5d0a6fc2fde2464f4931fb937a9f83b733abfb6) | Да | - | 20 | Любомир Филипов |
| 7 | [e5d0a6f](https://github.com/Kroxitrock/cancer-registry/commit/e5d0a6fc2fde2464f4931fb937a9f83b733abfb6) | Да | - | 28 | Любомир Филипов |
| 8 | [e5d0a6f](https://github.com/Kroxitrock/cancer-registry/commit/e5d0a6fc2fde2464f4931fb937a9f83b733abfb6) | Да | - | 28 | Любомир Филипов |
| 9 | [e5d0a6f](https://github.com/Kroxitrock/cancer-registry/commit/e5d0a6fc2fde2464f4931fb937a9f83b733abfb6) | Да | - | 28 | Любомир Филипов |
| 10 | [e5d0a6f](https://github.com/Kroxitrock/cancer-registry/commit/e5d0a6fc2fde2464f4931fb937a9f83b733abfb6) | Да | - | 10 | Любомир Филипов |
| 11 | [e5d0a6f](https://github.com/Kroxitrock/cancer-registry/commit/e5d0a6fc2fde2464f4931fb937a9f83b733abfb6) | Да | - | 11 | Любомир Филипов |
| 12 | [e5d0a6f](https://github.com/Kroxitrock/cancer-registry/commit/e5d0a6fc2fde2464f4931fb937a9f83b733abfb6) | Да | - |  | Любомир Филипов |
| 13 | [e5d0a6f](https://github.com/Kroxitrock/cancer-registry/commit/e5d0a6fc2fde2464f4931fb937a9f83b733abfb6) | Да | - |  | Любомир Филипов |
| 14 | [0831c70](https://github.com/Kroxitrock/cancer-registry/commit/0831c70957a8fdd2a11694c4c88336832aef786e) | Да | - | 12 | Любомир Филипов |
| 15 | [0831c70](https://github.com/Kroxitrock/cancer-registry/commit/0831c70957a8fdd2a11694c4c88336832aef786e) | Да | - | 12 | Любомир Филипов |
| 16 | [0831c70](https://github.com/Kroxitrock/cancer-registry/commit/0831c70957a8fdd2a11694c4c88336832aef786e) | Да | - | 12 | Любомир Филипов |
| 17 | [0831c70](https://github.com/Kroxitrock/cancer-registry/commit/0831c70957a8fdd2a11694c4c88336832aef786e) | Да | - | 12 | Любомир Филипов |
| 18 | [0831c70](https://github.com/Kroxitrock/cancer-registry/commit/0831c70957a8fdd2a11694c4c88336832aef786e) | Да | - | 13 | Любомир Филипов |
| 19 | [0831c70](https://github.com/Kroxitrock/cancer-registry/commit/0831c70957a8fdd2a11694c4c88336832aef786e) | Да | - | 13 | Любомир Филипов |

3. Бъгове.

|  |  |  |
| --- | --- | --- |
| Състояние | Описание | Поправка |
| Fixed | След влизане на потребител, property User(текущият влязал в системата потребител) от UserManager<ApplicationUser> e null. | След влизането на потребителя в системата, контролера пренасочва към друг action метод, а не връща определеното view. |
| Fixed | Account service връща една и съща грешка два пъти при регистрация на пациент. | След като има грешки се прави Distinct на property-то Errors на OperationResult. |
| Fixed | Добавяне на пациент в контекста на доктор не записва пациента в базата на DiagnoseContext. | След добавяне на определения пациент, се извиква SaveChanges. |
| Fixed | Грешките при валидация на даннитe при регистрация и влизане в системата са на английски. | Създаване на custom съобщения за грешки. |
| Fixed | Позволено логване на пациент чрез страницата за логване на доктор и обратното. | Направени са нужните проверки за остановяване на типа на влизащия потребител. |
| Fixed | След изтриване на профил на пациент или доктор, следва срив на системата поради невалидност на данните. | Изтриване на всички данни свързани с изтритият потребител. |
| Fixed | След въвеждане на определеното от доктора лечение, в базата данни постъпват едни и същи данни. | Направено е съответствие между параметрите от view-то и controller-а. |