Deliverable 2

Team NTP

Date: 2/24/2021

Nicholas Reilly - Team Leader

Paul Fundaro - Recorder

Tim Kodz

https://github.com/TimothyKodz/MediForm

**Actors:**

RegistrationStaff, Nurse, Physician, BillingStaff

**Use Cases and Scenarios**

|  |  |
| --- | --- |
| Use case name | loginUser |
| Participating actors | Initiated by User |
| Entry conditions | 1. The User starts up the MediForm system on their terminal. |
| Flow of events | 1. The system boots up and presents the User with a login screen. 2. The User enters their login username and password into the appropriate fields, and submits the information. 3. MediForm checks the users login information against its collection of user login information. If the login information does not match an entry in the collection, the information fields are cleared and the user is prompted to reenter the information. If the login information matches an entry in the collection the User is logged into the system. |
| Exit conditions | 1. The User is successfully logged in to the system |
| Quality requirements | 1. Mediform should provide feedback on the verification of User's login information within 5 seconds. |

|  |  |
| --- | --- |
| Scenario name | loginToMediformSystem |
| Participating actor instances | Nurse |
| Flow of events | 1. The Nurse power's on the terminal and launches the MediForm program. 2. The Nurse enters their username into the username text field. 3. The Nurse enters their password into the password test field. 4. The Nurse submits their login information by pressing the login button. 5. The Nurse is logged into the system after a brief delay. 6. The Nurse is brought to the first page of their MediForm interface. |

|  |  |
| --- | --- |
| Use case name | createRegistrationForm |
| Participating actors | Initiated by RegistrationStaff |
| Entry conditions | The user is logged into MediForm.  The user is role RegistrationStaff. |
| Flow of events | 1. The user initiates the createRegistrationForm function using the corresponding button. 2. MediForm provides the RegistrationStaff a blank registration form. 3. The RegistrationStaff completes the form with the information gathered from the patient. This information should include: name, address, phone number, date of birth, insurance information, primary physician, emergency contact information, current medication, allergies, relevant medical history, symptoms. 4. MediForm compiles the information entered and assigns the patient an ID number based on the number of previous patient entries. 5. MediForm saves the information, including ID, as a new entry in the collection. 6. MediForm presents the completed form to the RegistrationStaff, including patient ID, for review. 7. The RegistrationStaff finalizes the form and submits it. |
| Exit conditions | 1. The patient's registration form is saved to the collection. |
| Quality requirements | 1. Mediform should provide the user the blank medical form within 5 seconds. 2. Mediform should process and save the completed and finalized form within 5 seconds. |

|  |  |
| --- | --- |
| Scenario name | createRegistrationForm |
| Participating actor instances | RegistrationStaff |
| Flow of events | 1. The RegistrationStaff presses the createRegisrtationForm button on their main menu. 2. The RegistrationStaff enters the patient's: name, address, phone number, date of birth, insurance information, primary physician, emergency contact information, current medication, allergies, relevant medical history, symptoms; which they have received from the patient. 3. The RegistrationStaff submits the information using the submit button at the bottom of the form. 4. The RegistrationStaff reviews the compiled information and patient ID number. 5. The RegistrationStaff finalizes the information and submits it to the database. 6. The RegistrationStaff is returned to their main menu. |

|  |  |
| --- | --- |
| Use case name | ViewMedicalInfo |
| Participating actors | Initiated by Physician |
| Flow of events | 1. The Physician selects the ViewMedicalInfo function from the main menu. 2. MediForm responds by retrieving the medical info sheet from the medical info database. If no medical info sheet exists for the patient the NewMedicalInfo use case is used. 3. MediForm presents the generated form to the Physician. 4. The Physician updates any information, creates diagnosis using CreateDiagnosis use case, orders medical test using OrderMedicalTest use case, orders medication using OrderMedication use case, or views notes using ViewNotes use case. 5. The Physician closes the medical info sheet. 6. Mediform updates the patient’s medical information in the database. 7. Mediform returns the Physician to their main menu. |
| Entry conditions | 1. The user is logged into the MediForm system. 2. The user has the correct permissions. |
| Exit conditions | 1. The medical information sheet is saved for the patient. |
| Quality requirements | 1. MediForm should display the form within 5 seconds of being initiated. |

|  |  |
| --- | --- |
| Use case name | newMedicalInfo |
| Participating actors | Initiated by MediformSystem from viewMedicalInfo use case |
| Entry conditions | The system does not find a medical entry linked to patient's ID number. |
| Flow of events | 1. The MediformSystem searches the database's collections for entries linked to the patient's ID number. 2. From the entries found, the system pulls all relevant medical information, including: symptoms, allergies, medication, vitals, notes. 3. The MediFormSystem compiles the data into the medical information format, and displays the information to the user. |
| Exit conditions | 1. The patient's medical information is presented to the user |
| Quality requirements | 1. MediForm should display the empty form within 5 seconds of being initiated. 2. Mediform should save the medical information within 5 seconds. |

|  |  |
| --- | --- |
| Scenario name | newMedicalInfo |
| Participating actor instances | MediformSystem |
| Flow of events | 1. The MediformSystem searches the database's collections for entries linked to the patient's ID number. 2. From the entries found, the system pulls all relevant medical information, including: symptoms, allergies, medication, vitals, notes. 3. The MediFormSystem compiles the data into the medical information format, and displays the information to the user. |

|  |  |
| --- | --- |
| Use case name | createNotes |
| Participating actors | Initiated by the Physician or the Nurse |
| Flow of events | 1. The actor selects the CreateNotes Function 2. MediForm responds by retrieving a blank text box to add additional information about patient. 3. The actor hits submit at the bottom and create not sheet closes 4. MediForm updates patient notes |
| Entry conditions | 1. The user is logged into the MediForm system. 2. The user has the correct permissions. |
| Exit conditions | 1. The notes are saved to the system |
| Quality requirements | 1. MediForm should display the form within 5 seconds of being initiated. |

|  |  |
| --- | --- |
| Scenario name | notePatientStatus |
| Participating actor instances | Physician or Nurse |
| Flow of events | 1. Patient seem to be acting strangely or if there is something else wrong with patient then what is shown. 2. Actor will need to make a note so that the person knows to check for that certain issue. |

|  |  |
| --- | --- |
| Use case name | viewNotes |
| Participating actors | Initiated by the Physician or the Nurse |
| Flow of events | 1. The actor opens ViewNotes function at the bottom of their form. 2. MediForm responds by retrieving all the patients notes from the data base. 3. Displays the previous notes showing the most recent ones first. 4. The actor closes the view notes and returns to |
| Entry conditions | 1. The user is logged into the MediForm system. 2. The user has the correct permissions. |
| Exit conditions | 1. Returns to previous page. |
| Quality requirements | 1. MediForm should display the form within 5 seconds of being initiated. |

|  |  |
| --- | --- |
| Scenario name | checkNotes |
| Participating actor instances | Nurse and Physician |
| Flow of events | 1. The actor will need to check the notes before meeting with patient so that they know if the patient has anything else wrong with them. |

|  |  |
| --- | --- |
| Use case name | orderMedication |
| Participating actors | Initiated by the Physician |
| Flow of events | 1. Clicks the OrderMedication function in the ViewMedicalInfo sheet. 2. MediForm retrieves patients symptoms and picks appropriate medications that could be helpful for patient. 3. Displays the patients symptoms and a list of appropriate medication with a brief piece of information about medication. 4. The physician will be able to click on the medication he wants to give the patient. 5. The physician hits the order button and the order for the medication will go out and the medication will be saved to the data base. 6. The OrderMedication function will close and return to the ViewMedicalInfo form with the updated medication applied. |
| Entry conditions | 1. The user is logged into the MediForm system. 2. The user has the correct permissions. |
| Exit conditions | 1. Returns to ViewMedicalInfo sheet. |

|  |  |
| --- | --- |
| Scenario name | needMedication |
| Participating actor instances | Physician |
| Flow of events | 1. Physician tells patient that they will need medication to feel better. 2. They log onto MediForm to get the correct medication for the patients illness. |

|  |  |
| --- | --- |
| Use case name | OrderMedicalTest |
| Participating actors | Initiated by Physician |
| Flow of events | 1. The Physician selects the ViewMedicalInfo function from the main menu. 2. MediForm responds by retrieving the medical info sheet from the medical info database. If no medical info sheet exists for the patient the NewMedicalInfo use case is used. 3. MediForm presents the generated form to the Physician. 4. The Physician selects the OrderMedicalTest button at the bottom of the page, and then selects from a number of specific tests based on current patients’ symptoms. If more than one test needs to be done, the Physician will select as many as needed. 5. The Physician then clicks continue at the bottom of the page and is returned to the medical information page. 6. Mediform updates the patient’s medical tests ordered/performed in the database. |
| Entry conditions | 1. The user is logged into the MediForm system. 2. The user has the correct permissions. |
| Exit conditions | 1. The medical information sheet is saved for all actors necessary. |
| Quality requirements | 1. MediForm should allow user to OrderMedicalTest in less than 5 minutes. |

|  |  |
| --- | --- |
| Scenario name | OrderSingleTest |
| Participating actor instances | Physician |
| Flow of events | 1. The Physician logs into the system and enters the patients ID. 2. The Physician clicks the “Order Medical Test” button and enters the next screen. 3. The Physician selects to order an X-ray due to pain in the patient’s arm and swelling symptoms. 4. The Physician then clicks continue at the bottom of the screen and returns to the medical information form. 5. The X-ray test is added to the patient’s medical information form as well as the MediForm database. |

|  |  |
| --- | --- |
| Use case name | CompileBill |
| Participating actors | Billing Staff |
| Flow of events | 1. The Billing Staff user is logged into MediForm. 2. The system asks current user (Billing Staff) for a patient ID via text field and is entered in by the Billing Staff. 3. The Billing Staff then clicks the “Create Bill” button. 4. A printable bill sheet is then displayed for the given patient. |
| Entry conditions | 1. The user is logged into the MediForm system. 2. The user has the correct permissions. |
| Exit conditions | 1. Patient Bill is printed for patient overview and/or mailing |
| Quality requirements | 1. MediForm should display the form within 5 seconds of being initiated. |

|  |  |
| --- | --- |
| Scenario name | PrintBill |
| Participating actor instances | Billing Staff |
| Flow of events | 1. The Billing Staff user logs in to the system and enters the patient’s ID into the text field. 2. The Billing Staff user then clicks the “Create Bill” button in order to pull up the required patient’s bill. 3. The Billing Staff then prints the bill displayed and hands it to the patient. |

|  |  |
| --- | --- |
| Use case name | CreateDiagnosis |
| Participating actors | Physician |
| Flow of events | 1. The user will be viewing the medical information for the patient. 2. The user will look over patient’s symptoms/vitals/test results and will determine the best diagnosis pertained to patient. 3. The user will click the “Diagnose” button at the bottom of the medical information form. 4. The user will now select, from a list of 10 diagnosis’s, the correct diagnosis pertaining to patient by checking the corresponding checkbox (This will allow for more than one diagnosis to be selected). 5. The user will then click the “Add Diagnosis” button at the bottom of the page. 6. The user will now be brought back to the patient’s medical information form. |
| Entry conditions | 1. The user is logged into the MediForm system.  2. The user has the correct permissions. |
| Exit conditions | 1. Patient Diagnosis is updated into the system and medical information |
| Quality requirements | 1. MediForm should allow the user to input a diagnosis within 5 minutes. |

|  |  |
| --- | --- |
| Scenario name | BrokenBone |
| Participating actor instances | Physician |
| Flow of events | 1. The Physician reviews the notes and symptoms of the patient. 2. The Physician then hits the “Diagnose” button on the screen and rules that patient has a broken bone. 3. The Physician will then check the “broken bone” box in the next screen and then click the “Add Diagnosis” button 4. The Physician will now be brought back to the medical information page of that patient, and the broken bone diagnosis will be updated into the patient’s medical information as well as the MediForm database. |