**CSCI 4400**

**Spring 2018**

**Report**

**RIS**

**Ben Denton**

**Jed Dockery**

**Josh Knight**

**Wesley Donald**

1.Introduction

a. Purpose:

* An RIS system that can schedule and register patients, maintain billing information, create/format and store diagnostic reports, track film folders, process patient and images, generate reports, monitor status of patients and examinations

b. Scope:

* The scope of the project is to develop a system that orders can be scheduled in

c. Objectives/ Success Criteria:

d. Definitions, Acronyms, and Abbreviations

* Definitions
  + Use Case:
    - specify a sequence of actions, including variants, that the system can perform, and that yield an observable result of value to a particular actor
    - Represents something of value that the system does for its actors
  + Actors:
    - Someone/something outside the system, acting in a ***role*** that interacts with the system
    - An actor represents a role that a human, hardware device, or another system can play in relation to the system
  + Functional Requirement:
    - The intended interactions between users and the system, independent of implementation
  + Non-Functional Requirement:
    - Specifies criteria that can be used to judge the operation of a system, rather than specific behaviors
  + Pseudo Requirement:
    - Imposed by the client or the environment in which the system is to operate
* Abbreviations
  + Admin:
    - Administrator of the System

2. Current system

* The actual system has not begun development; however, prototypes are being developed

3. Proposed System

a. Overview

* Create a system that allows for the management of data obtained by radiological devices. After a radiologist successfully logs in with their assigned credentials, the system should be able to have a dedicated database to hold the list of images pertaining to a specified patient.

b. Functional requirements

* The software doesn’t allow users to have the same username, there are no two usernames alike
* Ability to register a new account on the system
* Database changes(add, delete, or edit of information) should be updated in the tables within the program
* The RIS should be able to provide a login system, with users that have different permissions.
* The RIS should be able to schedule patients for different radiology exams
* The RIS should be able send patient records to the PACS for storage
* The RIS should be compatible with different modalities.
* The RIS should be able to display the acquired images.
* The RIS should be able to create reports.

c. Non-functional requirements

* Error Handling & Extreme conditions
* Quality Issues
* System Modifications
* Security Issues
* The RIS should be able to pull the user's data off of a database.
* The RIS will be able to view occupied time slots and be able to create time slots in empty areas.
* The RIS will be able to upload data to the database (PACS)
* The different modalities will produce similar file types that can be displayed on the reading stations.
* The reading station will be able to read images from the PACS with its original resolution.
* The reading station will be able to support the creation and editing of fillable pdfs

d. Pseudo Requirements

* Implementation language using Java
* Must be able to connect to internet to connect to database

e. System Models

* Use Case Models:

|  |  |
| --- | --- |
| UC Name | ReportSelect |
| UC ID | REPORTSELECT |
| Screen Mock-ups | UserMainPage, SelectionPage, SelectionPopup, SelectionError |
| Version | 1.0.0 |
| Author | Denton, Dockery, Donald, Knight |
| Date | 2/3/18 |
| Summary | This Use Case will show how a **MD**'s report selection will bring the **MD** to an information popup page that will display detailed information on said selection. |
| Basic Path | 1. The system displays a screen with a list of **Report** objects available for selection by the **MD**. 2. **MD** selects from available list of options or searches catalog. 3. System will locate the selected **Report** object. 4. System will display a popup screen with detailed information about the **Report** object. |
| Exception Paths | In step 2: If **MD** catalog, exit ReportSelect use case and enter SearchReport use case.  In step 4: If **MD** is unable to retrieve object information page, display SelectionError popup. |
| Triggers | The **MD** decides to select a report from the provided screen of options after the initial login. |
| Pre-conditions | The **MD** has searched for an item using the search function or the **MD** has decided to select a list of reports from the home screen.(ie. Recently Added, etc.) |
| Post-conditions | **MD** object is available to the other use cases available to the same **MD**. |

|  |  |
| --- | --- |
| UC Name | CreateReport |
| UC ID | CREATEREPORT |
| Screen Mock-ups | SettingsPage, CreateReportPage, |
| Version | 1.0.0 |
| Author | Denton, Dockery, Donald, Knight |
| Date | 2/5/18 |
| Summary | This use case will give the **MD** the ability to create a new report and add it to the list of reports for a given **Patient** |
| Basic Path | 1. The **System** displays a screen with the option to create or manage reports. 2. **User** selects the create reports option and is redirected to a screen with a form about the new report. 3. **Admin** fills out the form with the appropriate information about the report i.e. the title, brief description, etc. 4. **System** will display a confirmation screen and **Admin** will either accept or decline the current information. 5. Once submitted the new report will be added to the database. |
| Exception Paths | 1. The **Admin** decides to decline the accuracy of the current information and is redirected back to the form screen to edit information. |
| Triggers | The **Admin** wishes to add another report to the database. |
| Pre-conditions | The **Admin** has accessed the system settings page. |
| Post-conditions | The system has added a new movie to the database and the **Admin** is returned to the settings page. |

|  |  |
| --- | --- |
| UC Name | LogIn |
| UC ID | LOGIN |
| Screen Mock-ups | Login, AdminMainPage, MDMainPage, ErrorMessage, SuccessMessage |
| Version | 1.0.0 |
| Author | Denton, Dockery, Donald, Knight |
| Date |  |
| Summary | This use case describes the steps necessary to gain access to the application and use all of its functions. |
| Basic Path | 1. The system displays prompts for the **Username** and **Password** and **User Type**; the **User Type** is displayed as a selection list; the **LOGIN button** is displayed as well. 2. The **User** enters the **Username**. 3. The **User** enters the **Password.** 4. The **User** selects a **User Type.** 5. The **User** selects whether or not he/she is an admin by checking the checkbox 6. The **User** presses the **LOGIN** button. 7. The system verifies the **User**’s access rights. 8. The system locates the **User**’s **User** object. 9. The system displays a welcome message and the main page for the user side according to **User Type** (**Admin, User**). |
| Exception Paths | If in step 6. The **User** fails the verification, they will be prompted with a **ErrorMessage** and will be returned to step 1. |
| Triggers | The **User** decides to use the system. |
| Pre-conditions | The **User** hasaccessed the application. |
| Post-conditions | **User** object is available to the other use cases available to the same **User**. |

|  |  |
| --- | --- |
| UC Name | LogOut |
| UC ID | LOGOUT |
| Screen Mock-ups | AdminMainPage, MDMainPage, LogoutMessage |
| Version | 1.0.0 |
| Author | Denton, Dockery, Donald, Knight |
| Date |  |
| Summary | This use case describes the steps necessary to sign-out of a **User**’s session and return to the basic login screen. |
| Basic Path | 1. The **User** will select the **LOGOUT** button. 2. The **User** will be asked by the system with the prompt “Are you sure you want to logout?” 3. The **User** will select yes or no 4. If yes is selected, the **User** will be logged out. 5. The system terminates itself. |
| Exception Paths | If in step 3. The **User** clicks no, they will be returned to their previous screen. |
| Triggers | The **User** decides to exit the system. |
| Pre-conditions | The **User** clicks the logout button. |
| Post-conditions | The **User** has exited the system. |

|  |  |
| --- | --- |
| UC Name | ManageReport |
| UC ID | MANAGEREPORT |
| Screen Mock-ups | SettingsPage, ManageReportPage, DidNotSaveMessage, ReportList |
| Version | 1.0.0 |
| Author | Denton, Dockery, Donald, Knight |
| Date |  |
| Summary | This use case details how the system will allow the **Admin** to make edits to the information about each report within the database. |
| Basic Path | 1. The **Admin** clicks on the Manage Report button 2. The system will then display a list of all report on the Report List Page. 3. The **Admin** will then select a report from the list. 4. The system will access the report database and display a screen with information on the selected report. 5. The **Admin** will make any desired changes to the report. 6. Once finished, **Admin** will click on Save button. 7. The system will update the report in the database. |
| Exception Paths | If in step 6. The **Admin** doesn’t click save, the system will display a message stating,”You have not saved your changes, are you sure you wish to continue?” If they select yes, no changes are made to the selected report and **Admin** is returned to Report Management page. If they select no, Admin is taken back to previous screen to click **Save** button. |
| Triggers | The **User** decides to make a change to the information describing a specific report. |
| Pre-conditions | The **Admin** has accessed the system and has navigated to the settings page for report management. |
| Post-conditions | The system has processed any changes made to the selected report and has updated the database. The **Admin** isthen returned to the settings page. |

|  |  |
| --- | --- |
| UC Name | Payment |
| UC ID | PAYMENT |
| Screen Mock-ups | ErrorMessage, UserMainPage, ReturnMainPage, PaymentSuccessMessage |
| Version | 1.0.0 |
| Author | Denton, Dockery, Donald, Knight |
| Date |  |
| Summary | This use case details the steps to view and make payments as a patient. |
| Basic Path | 1. The **System** displays a screen with the total payment required by the **patient**. 2. The **System** displays payment history including the portion that the **patient’s** health insurance provider is covering. 3. The **System** provides a payment button which allows the **patient** to input his/her debit card information along with the amount to pay. 4. The **System** processes the payment and provides the **patient** with a receipt. |
| Exception Paths | If the payment information is invalid or there are insufficient funds in the **patient**’s bank account. |
| Triggers | **Patient** wishes to make a payment and clicks the payment button listed on their main page. |
| Pre-conditions | The **patient** has logged in and payment page. |
| Post-conditions | The **System** has processed the payment and updated the **patient’s** balance. |

|  |  |
| --- | --- |
| UC Name | CreateUser |
| UC ID | CREATEUSER |
| Screen Mock-ups | SettingsPage, CreateUserPage |
| Version | 1.0.0 |
| Author | Denton, Dockery, Donald, Knight |
| Date |  |
| Summary | This use case details the steps to create a new user profile. |
| Basic Path | 1. The **System** displays a screen with the option to create or manage **Users**. 2. **Admin** selects the Create User option and is redirected to a screen with a form about the new user. 3. **Admin** fills out the form with the appropriate information about the User i.e. name, age, etc. 4. **System** will display a confirmation screen and **Admin** will either accept or decline the current information. 5. Once submitted the new **User** will be added to the database. |
| Exception Paths | If in step 4. The **Admin** decides to decline the accuracy of the current information and is redirected back to the form screen to edit information. |
| Triggers | Admin wishes to create a new **User** profile. |
| Pre-conditions | The **Admin** selects the Create User button on the settings page. |
| Post-conditions | The system has added a new **User** profile to the database. **Admin** is taken back to settings page. |

|  |  |
| --- | --- |
| UC Name | ManageUser |
| UC ID | MANAGEUSER |
| Screen Mock-ups | SettingsPage, ManageUserPage, DidNotSaveMessage, UserList |
| Version | 1.0.0 |
| Author | Denton, Dockery, Donald, Knight |
| Date |  |
| Summary | This use case details how the system will allow the **Admin** to make necessary changes/edits to the information on each **User** profile. |
| Basic Path | 1. The **Admin** clicks on the Manage User button 2. The system will then display a list of all Users on the User List Page. 3. The **Admin** will then select a **User** from the list. 4. The system will access the **User** database and display a screen with information on the selected **User**. 5. The **Admin** will make any desired changes to the **User**. 6. Once finished, **Admin** will click on Save button. 7. The system will update the **User** in the database. |
| Exception Paths | If in step 6. The **Admin** doesn’t click save, the system will display a message stating,”You have not saved your changes, are you sure you wish to continue?” If they select yes, no changes are made to the selected movie and **Admin** is returned to User List page. If they select no, Admin is taken back to previous screen to click **Save** button. |
| Triggers | The **Admin** wants to make a change to a given **User’s** profile information. |
| Pre-conditions | The **Admin** has accessed the system and has navigated to the settings page for **User** management. |
| Post-conditions | The system has processed any changes made to the selected **User** and has updated the database. The **Admin** isthen returned to the settings page. |

|  |  |
| --- | --- |
| UC Name | SearchReport |
| UC ID | SEARCHREPORT |
| Screen Mock-ups | SearchBar, ErrorMessage, UserMainPage, ReturnMainPage |
| Version | 1.0.0 |
| Author | Denton, Dockery, Donald, Knight |
| Date |  |
| Summary | This use case will detail the process the **User** will take to search for a movie within the database and display the results. |
| Basic Path | 1. The **User** will type into the search bar. 2. The System will then display a list of items that match the text entered into the search bar. |
| Exception Paths | If in step 2. There are no items matching the text entered, the System will display an ErrorMessage. |
| Triggers | A **User** wants to search for a specific movie that may not be displayed on the current screen. |
| Pre-conditions | The **User** does not see the specific movie he/she wants and elects to use the search database option. |
| Post-conditions | The system has searched for a desired movie based on the **User**’s input and has displayed a list of results. |