Radiology Information System

Seth Nurmi, Alyssa Dames, Brenden Lopez-Fulbright, Aaron Velazquez, John Rudowicz

Team Member Responsibilities

- Seth Nurmi - Project Lead
  - System architecture
  - Front-end development
  - Back-end development
- Brendan Lopez-Fulbright
  - Back-end development
  - Data management
  - Web hosting
- Alyssa Dames
  - Front-end development
  - Back-end development
  - Java development
- Aaron Velazquez
  - Java development
  - Stress testing
  - Debugging
- John Rudowicz
  - Java development
  - Stress testing
  - Debugging

Project Description

The purpose of this Radiology Information System is to:

- Connect primary care physicians with radiology specialists to place referral orders
- Schedule patient appointments to fulfill placed orders and check-in patients
- Upload and download modality imaging for remote viewing
- Produce diagnostic reports from radiologists and deliver to original referral/primary care physician
- Support custom modality options based on available systems in radiology office

Functionalities

1. Allow a network doctors to place patient referrals
   1.1. Search database of existing patients to begin order
   1.2. Create a new patient to begin order
2. Allow Referral MD to view placed orders
   2.1. Placed orders display order status, indicating radiology office progress on order fulfillment
3. Allow radiology office receptionists to schedule appointments and check-in patients
4. Allow radiology office technicians to fulfill orders
   4.1. Modality images can be upload and assigned to an order
5. Allow radiology office radiologists to compose a diagnostic report and complete an order
   5.1. Order images can be downloaded and viewed
   5.2. Once diagnostic report is submitted, order is closed
6. Allow system admin to create or modify entries in all form tables
   6.1. Includes system settings such as modalities and patient alerts

Project Plan

1. Setup development environment
    1.1. Install VisualStudio Code
    1.2. Install JDK
    1.3. Install Maven
    1.4. Install MySQL Workbench
    1.5. Configure SpringBoot starter project
2. Develop user authentication and role authorization
    2.1. Configure SpringBoot Security
    2.2. Develop system roles and privileges
        2.2.1. Admin, User, Referral MD, Receptionist, Technician, Radiologist
3. Develop patient form and patient data-tables
    3.1. Update and create new patients, display patients in searchable data-table
4. Develop order form and order data-tables
    4.1. Update and create new orders, display orders in searchable data-table

Project Plan Cont.

6. Develop file uploading and downloading
   - 6.1. Encrypt and store images locally on system server
7. Develop diagnostic report form and data-tables
   - 7.1. Update and create new diagnostic reports, display diagnostic reports in searchable data-table
8. Develop admin portal with all system forms and searchable data-tables
   - 8.1. Includes users, patients, orders, appointments, file uploads, diagnostic reports, modalities, and patient alerts
9. Develop database tables to store and retrieve data
   - 9.1. Configure Java Hibernate settings to connect to database
10. Stress test and debug system
    - 10.1. Produce test cases to reveal issues and flaws in the system
11. Finalize system functionalities
    - 11.1. Ensure that all use cases perform necessary functions without failure

Project Schedule

- Planning: 2/10/21 - 2/23/21
  - Develop RIS functional requirements
- Environment Setup 1: 2/24/21 - 3/3/21
  - Setup development environment (Initial: CodeIgniter3, XAMPP)
  - Produce project diagrams
- Environment Setup 2: 3/4/21 - 3/7/21
  - Setup development environment (Final: SpringBoot, Maven, MySQL Workbench)
  - Assign project responsibilities
- Initial Development: 3/8/21 - 3/17/21
  - Develop java persistence/repository classes for users, patients, orders
  - Develop associated web forms and data-tables
- Major Development: 3/18/21 - 3/26/21
  - Develop java persistence/repository classes for appointments, file uploads, diagnostic reports
  - Develop associated web forms and data-tables
- Final Development: 3/27/21 - 4/6/21
  - Develop admin portal, all forms and data-tables
  - Perform stress testing
    - Develop test cases
    - Resolve issues
- Post Development: 4/7/21 - 4/11/21
  - Finalize system functionalities
  - Finalize deliverable documents
  - Produce project presentation

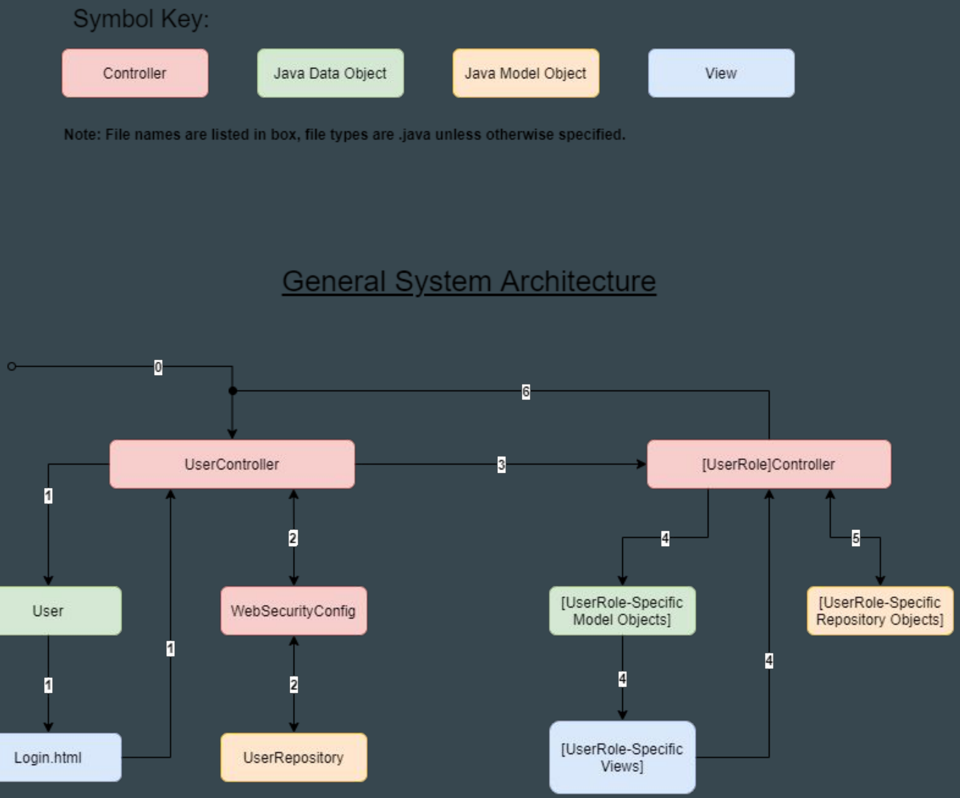
Tools Used

- Development Environment
  - VisualStudio Code
    - Java Extension Pack
    - Git for VSCode
  - MySQL 8.0
    - Server
    - Workbench
    - Command Line Client
  - Apache-Maven 3.6.3
  - JDK 14.0.2
- Source Control
  - Github

- Web Development Framework
  - SpringBoot 2.4.4
  - Spring Security 5.4.6
  - Hibernate JPA
- Diagram Tools
  - Diagrams.net
  - DBDiagram.io
  - MySQL Workbench
- Communication and Coordination
  - Discord
  - Microsoft Teams
  - Jira

System Architecture

0. Login Get Request
1. Login Process: Send User data object to Login view, on form submission send form-data to controller.
2. User Authentication: Send User data response object from form-data to WebSecurityConfig which passes the User data to the UserRepository model to check against the database. Returns authentication results if successful (User data, UserRole).
3. Successful Authentication: Pass system control to the specified UserRole controller,
4. UserRole-Specific Functionalities: Send UserRole-specific data objects to views, return form-data objects to controller.
5. Handle Form Data: Send data objects to associated model objects to interface with database, return results to controller. Go To: 4
6. Logout Get Request

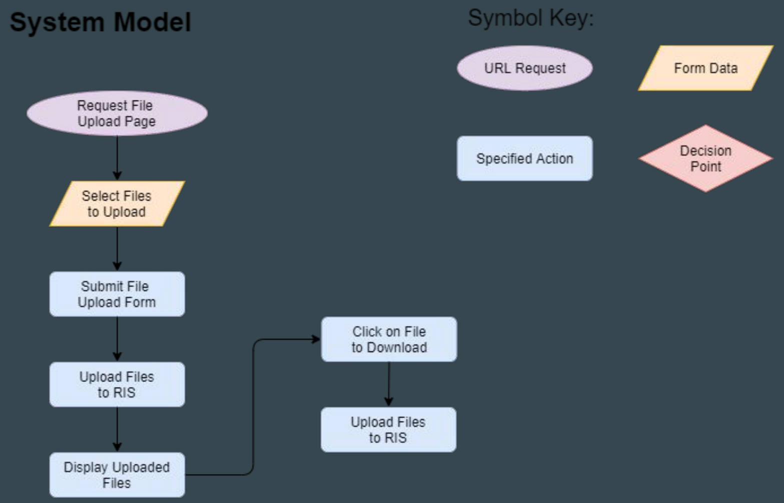

Symbol Key:
Controller
Java Data Object
Java Model Object
View
Note: File names are listed in box, file types are .java unless otherwise specified.
General System Architecture
0
6
UserController
3
[UserRole]Controller
1
2
4
5
User
WebSecurityConfig
[UserRole-Specific Model Objects]
[UserRole-Specific Repository Objects]
1
1
2
4
4
Login.html
UserRepository
[UserRole-Specific Views]

See RIS Architecture deliverable for complete system architecture

Interesting and Important Functionality

File Upload / Download System

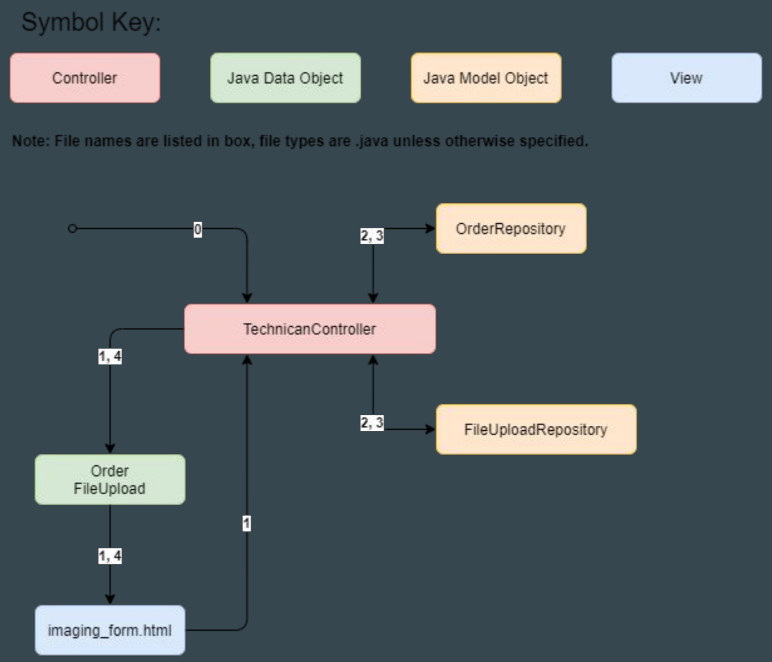
- Functional Requirements:
  - Allow user to upload files from local computer
  - Store files in specified RIS upload location
  - Track which order is associated with file uploads
  - Display file uploads associated with order
  - Download files from RIS
- Nonfunctional Requirements
  - Files must be encrypted on upload and decrypted on download to prevent sensitive information from becoming easily accessible
  - A single file upload should only be associated with one order, but multiple file uploads may be associated with a single order.



Interesting and Important Functionality

0. File Upload Page Get Request
1. Imaging Order Form: Send Order and FileUpload data objects to imaging_form view, return interactions and form-data to controller.
2. Order Model Interactions: Query/Send Order data object to OrderRepository, return results to controller.
3. File Upload Model Interactions: Query/Send Order data object to OrderRepository, return results to controller.
4. Display File Uploads: Send Order and FileUpload data objects to imaging_form view, return interactions and form-data to controller.

Architectural Design



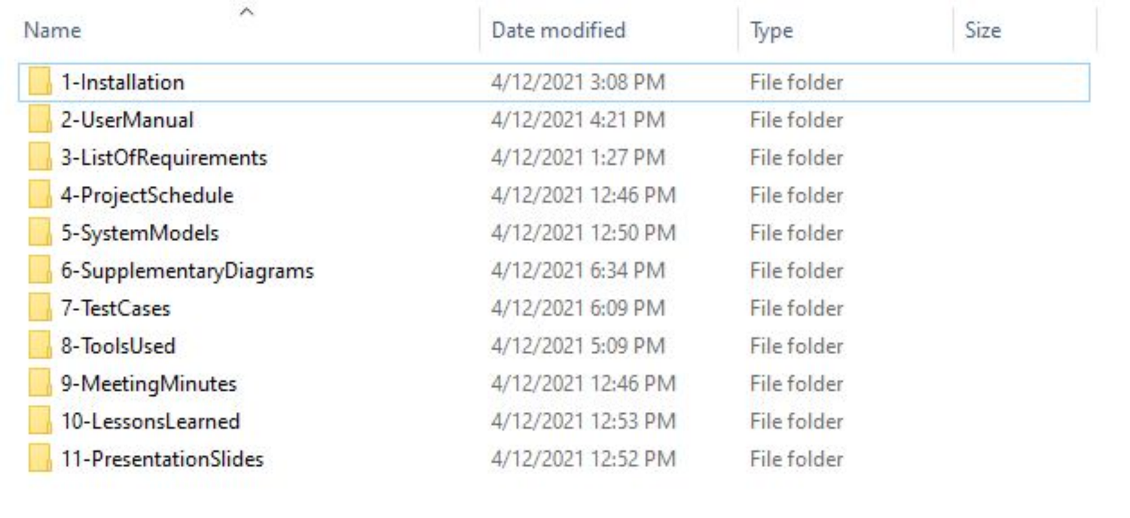
Note: File names are listed in box, file types are .java unless otherwise specified.

Lessons Learned

- Communication and planning is key
  - Working on a project with several people requires communication, especially with conflicting schedules. Planning ahead and having group discussions helps make everything run smoothly.
  - Planning is essential in order to stay on time with a project. Working with people of different skills requires proper delegation of tasks.
- You must be flexible
  - Working on this project required us to learn a new framework in order to implement the requirements of the project. We had to have an open-mind and learn something new in order to accomplish our goal.
- Take advantage of online tools to streamline workflow
  - Github, Diagrams.net, and Baeldung.com were extremely helpful in collaborating on this project, building visual diagrams, and identifying solutions to problems encounters.
- SpringBoot is a powerful web development framework
  - Using SpringBoot for this project was initially challenging, but after learning how the framework operates, development quickly gained speed.

Project Deliverables

1. Installation
2. User Manual
3. List Of Requirements
4. Project Schedule
5. System Models
6. Supplementary Diagrams
7. Test Cases
8. Tools Used
9. Meeting Minutes
10. Lessons Learned
11. Presentation Slides



Name	Date modified	Type	Size
1-Installation	4/12/2021 3:08 PM	File folder	
2-UserManual	4/12/2021 4:21 PM	File folder	
3-ListOfRequirements	4/12/2021 1:27 PM	File folder	
4-ProjectSchedule	4/12/2021 12:46 PM	File folder	
5-SystemModels	4/12/2021 12:50 PM	File folder	
6-SupplementaryDiagrams	4/12/2021 6:34 PM	File folder	
7-TestCases	4/12/2021 6:09 PM	File folder	
8-ToolsUsed	4/12/2021 5:09 PM	File folder	
9-MeetingMinutes	4/12/2021 12:46 PM	File folder	
10-LessonsLearned	4/12/2021 12:53 PM	File folder	
11-PresentationSlides	4/12/2021 12:52 PM	File folder	