

Zachary Mace,
Kevin Freyre,
Andrew Davis,
Cole Ledford

Radiology Information System

Team Responsibilities

- ❖ Zachary Mace – Team Lead
 - Features Architecture
 - Front/Back End development
 - Web Hosting
 - Debugging
 - Stress Testing

- ❖ Kevin Freyre – Scribe
 - Front/Back End development
 - Debugging
 - Stress Testing
 - Documentation

- ❖ Andrew Davis
 - User Info feature owner
 - Modeling and Design
 - Stress Testing
 - Debugging

- ❖ Cole Ledford
 - Billing feature owner
 - Modeling and Design
 - Stress testing
 - Debugging

RIS Workflow

- ❖ RIS Purpose: manage information and workflow in a radiology department
 - ❖ Schedule and register patients and appointments
 - ❖ Process patients and upload images
 - ❖ Create and format diagnostic reports
 - ❖ Store and manage diagnostic reports
 - ❖ Digitally track imaging
 - ❖ Input and manage billing information

Project Description

- ❖ Install one of the two supplied RIS systems
- ❖ Test system and identify defects
- ❖ Modify system to address the bugs discovered
- ❖ Add two additional features to the system through software engineering process
 - ❖ Requirements
 - ❖ Design
 - ❖ Implementation
 - ❖ Verification
 - ❖ Maintenance

Project Tools Used

- ❖ Development Environment
 - ❖ VisualStudio Code
 - Java Extension Pack
 - Git for VSCode
 - ❖ MySQL 8.0 Community
 - Server
 - Workbench
 - ❖ JDK 11.0.2
 - ❖ Apache-Maven 3.6.3
- ❖ Source Control
 - ❖ GitHub
- ❖ Web Development Framework
 - ❖ SpringBoot 2.4.4
 - ❖ Spring Security 5.4.6
 - ❖ Spring Email 2.3.4
 - ❖ Hibernate JPA
- ❖ Diagram Tools
 - ❖ Diagrams.net/Draw.io
 - ❖ MySQL Workbench
- ❖ Communication and Coordination
 - ❖ Discord
 - ❖ Microsoft Teams
 - ❖ Jira

Project Agenda

- ❖ Planning 06/30/2021 – 07/07/2021
 - Analyzed team member experience
 - Outlined project requirements
 - Familiarized with project tools
- ❖ System Install/Testing 07/07/2021 – 07/14/2021
 - Installed system to maintain
 - Setup database
 - Began testing and identifying issues
 - Debugging
- ❖ Feature Implementation 07/14/2021 – 07/21/2021
 - Researched potential new features
 - Feature approvals
 - Began designing/developing new features
- ❖ Feature Testing 07/21/2021 – 07/28/2021
 - Finished developing features
 - Tested and identified issues
 - debugging
- ❖ Project Finalization 07/28/2021 – 08/01/2021
 - Finished all deliverables and gathered all documentation

New Project Features

Billing

- ❖ Cost of appointment is assigned and determined by modality
- ❖ Addition of Insurance information and price in appointment scheduling.
- ❖ Billing statement sent to patient after having checked-in for their appointment

User Info

- ❖ Ability for users to manually edit their account information such as password, email and display name

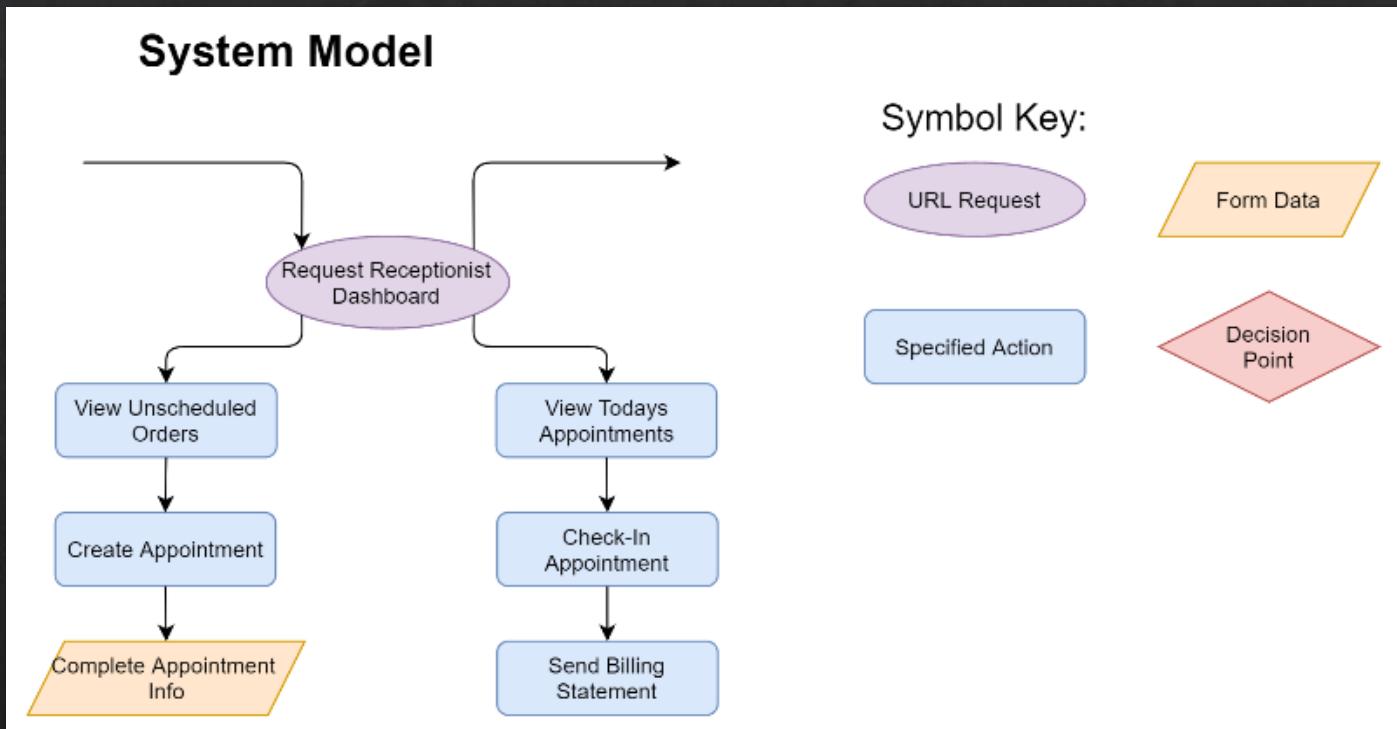
Billing Feature

Functional Requirements

- Ability to change the price based on the choosing of a modality.
- Retrieval of Insurance information at the time of appointment scheduling.
- Total cost of modality displayed on receptionist dashboard as well as at the time of appointment scheduling
- Delivery of an email with price of modality as well as insurance information provided

Nonfunctional Requirements

- Reliably provide modality price and insurance info through email.
- Maintain Data integrity by safeguarding personal information from unauthorized/uncredentialed personnel
- Provide serviceability and maintainability to future email features



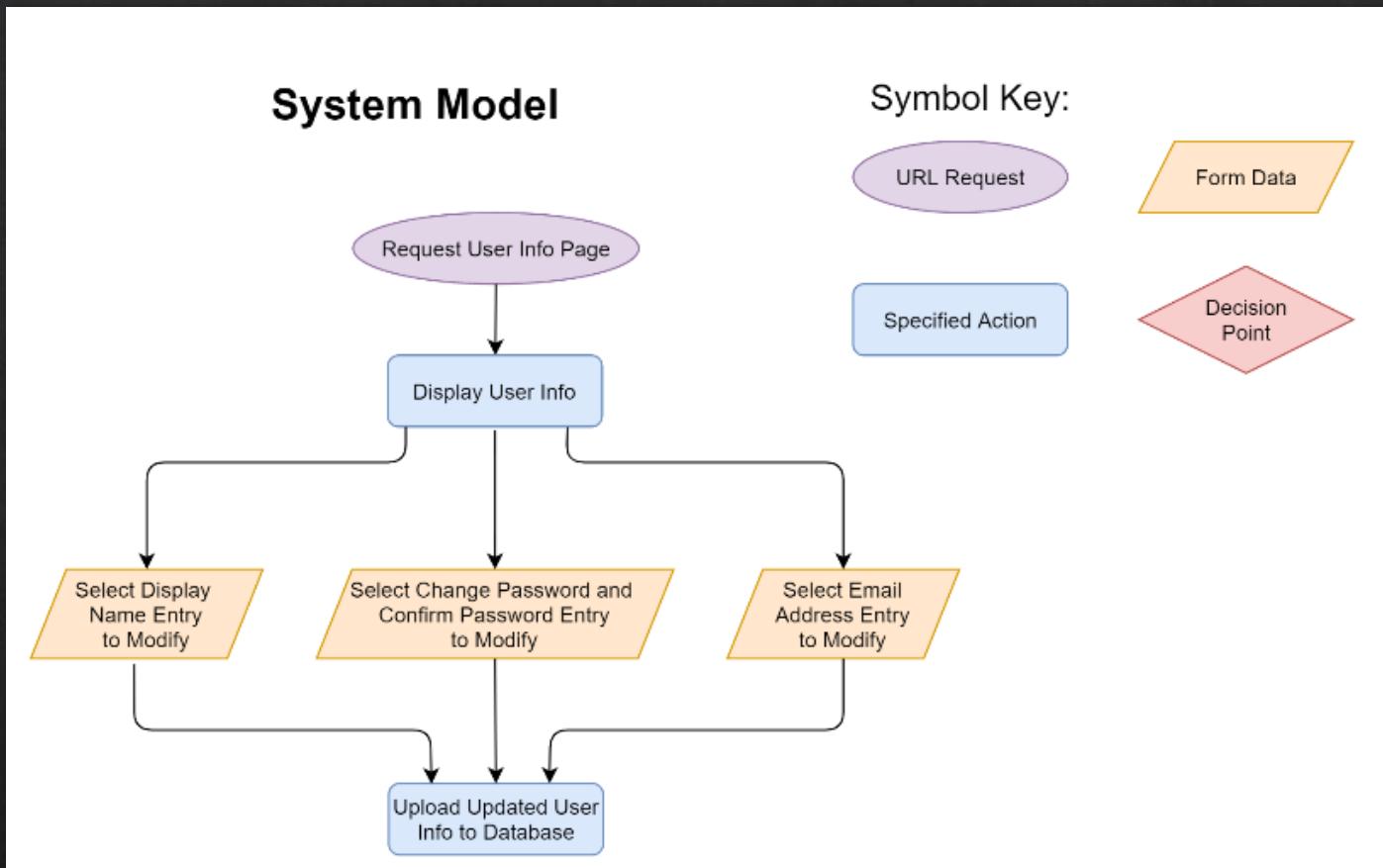
User Info Feature

Functional Requirements

- Allow user to update password, display name and email address
- Allow user to view user id and username
- Prevent user from updating user id and username

Nonfunctional Requirements

- Provide adaptability by allowing end users to change user info
- Provide accessibility by allowing user to view information that pertains to them
- Providing usability in an easy to use, simple dashboard for users



Lessons Learned

What went well:

- As a whole team communication was a strong point. We stayed on track and stayed connected on important, time-sensitive issues.
- As a team, we were very adaptable to changing environments and working in a distributed development team.
- Time management was not an issue for most of the project

What needs to be improved:

- Documentation of assigned tasks, problems, and established codebase.
- Communication with the client. Several deadlines were missed due to a misunderstanding with the wants of the client.
- Interleave design and implementation steps better. Design should be done sooner to better the team's understanding of the implementation process.