Important Functionality:
Billing Statement

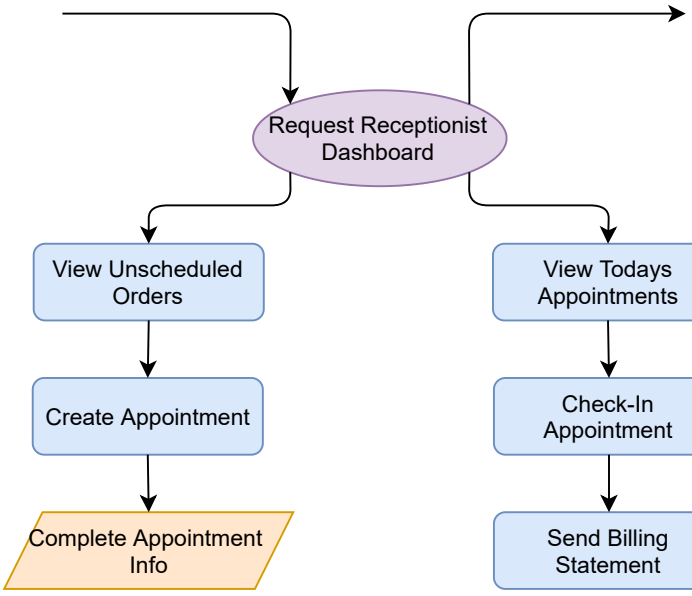
Functional Requirements

- 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

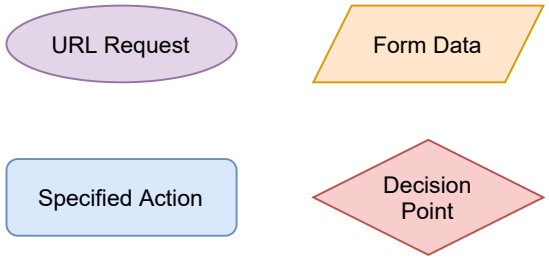
Nonfunctional Requirements

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

System Model



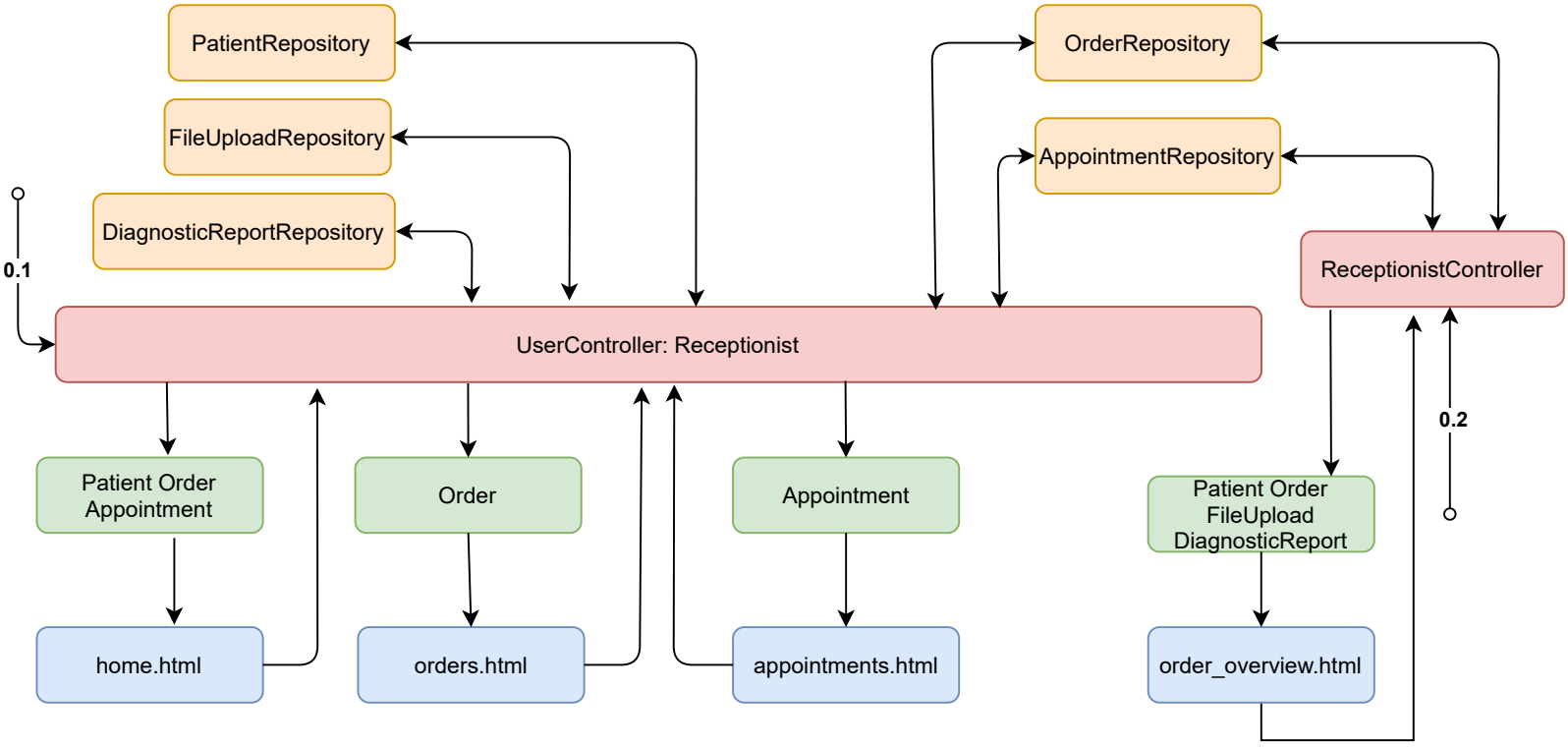
Symbol Key:



Architectural Design

Symbol Key:

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



0.1 Successful Authentication - Get Request:/Home, /Orders, /Appointments, /Order/{order_id}

0.2 Get Request: /Diagnostics/Order/{order_id}, Post Request: /Diagnostic?Submit Report

1 Receptionist Dashboard: Send User, Patient, Order, and Appointment data objects to home view, return interactions and form-data to controller.

2 Orders Data Table: Send User and Order object to orders views, return interactions to controller

3 Appointments Data Table: Send User, and Appointment object to appointments view, return interactions to controller

4 Order Overview: Send User, Patient, and Order data objects to order_overview view, return interactions to controller

1.2, 4.4 Patient Model Interactions: Query/Send Patient data object to PatientRepository, return results to conroller

1.3, 2.2, 4.2, 0.2.3 Order Model Interactions: Query/Send Order data object to OrderRepository, return results to conroller

1.4, 3.2, 0.2.2 Appointment Model Interactions: Query/Send Appointment data object to AppointmentRepository, return results to conroller

4.3 FileUpload Model Interactions: Query/Send FileUpload data object to FileUploadRepository, return results to conroller

4.4 DiagnosticReport Model Interactions: Query/Send DiagnoticReport data object to DiagnosticReportRepository, return results to conroller

Detailed Design

End User	Network	RIS Application	MySQL Database

Important Functionality: User Info Dashboard

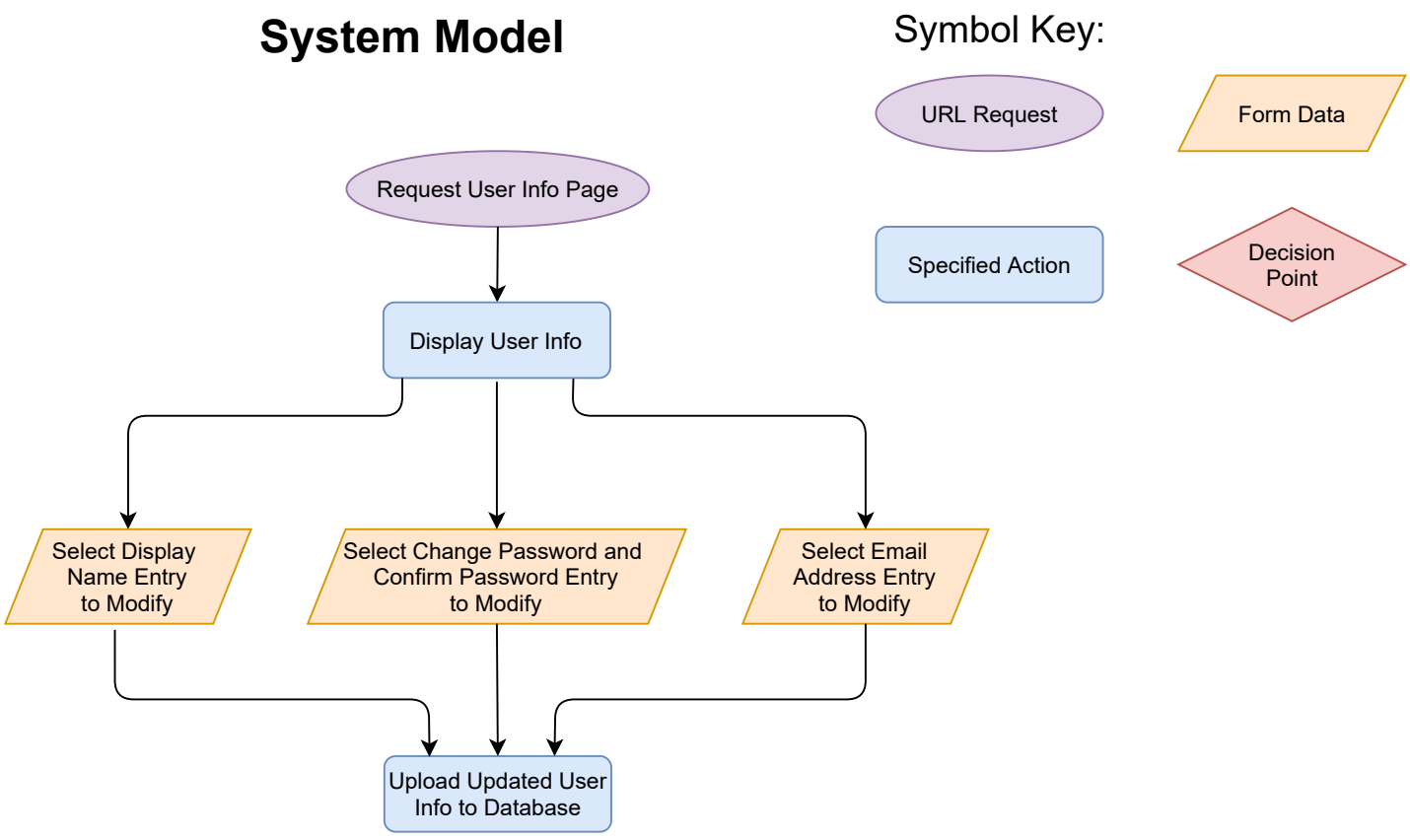
Functional Requirements

- 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 a easy to use, simple dashboard for users.

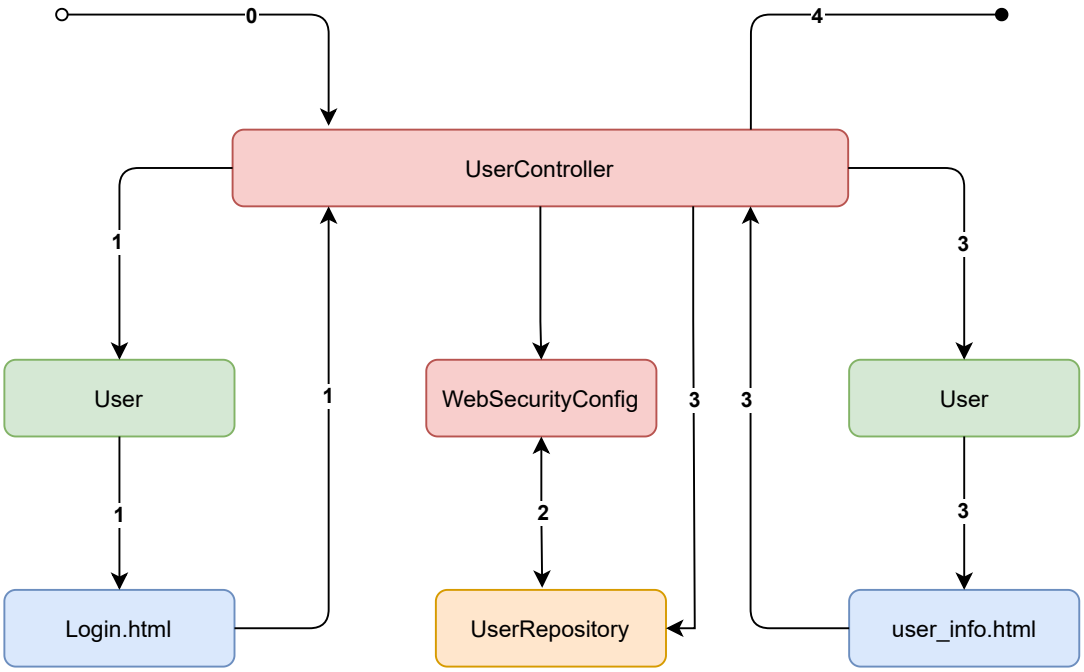
System Model



Architectural Design

Symbol Key:

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



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. Update User Info Functionality: Available to all users. Send User Data object to user_info view, send User Form data to USer Controller, Query the database with updated user information.

4. Logout

Detailed Design

End User	Network	RIS Application	MySQL Database