

System Design Document for Catalyst

Version 1.0

Authors: Amir Ali Mohaghegh, Barbod Habibi, and Kevin Nguyen

Date: 2024-11-21

Table of Contents

CRC Cards - 1

1.1 Catalyst Application	3
1.2 Navigation Controller	3
1.3 Optometrist Controller	3
1.4 Patient Controller	3
1.5 Optometrist Service	4
1.6 Patient Service	4
1.7 Optometrist Service Implementation	4
1.8 Patient Service Implementation	5
1.9 Optometrist Repository	5
1.10 Patient Repository	5
1.11 Optometrist	6
1.12 Patient.....	6
1.13 Visual Assessment	6

Software Architecture Diagram - 2

2.1 Diagram	7
-------------------	---

CRC Cards

Class Name: CatalystApplication	
Parent Class: N/A Subclasses: N/A	
Responsibilities: Runs the main program through the spring framework. It contains the main method.	Collaborators: N/A

Controllers:

Class Name: NavigationController	
Parent Class: N/A Subclasses: N/A	
Responsibilities: Map general URLs to HTML templates.	Collaborators: - Thymeleaf templates: index.html, assessments.html.

Class Name: OptometristController	
Parent Class: N/A Subclasses: N/A	
Responsibilities: Handles HTTP requests, redirections, and renders appropriate Thymeleaf templates associated with optometrist resources. It delegates CRUD operations to the appropriate service layer and returns relevant outputs to the user.	Collaborators: - OptometristService: Handles business logic and invokes data interactions. - Model: Supplies data to Thymeleaf templates. - Optometrist: Represents the entity being managed. - BindingResult: Captures validation results. - Thymeleaf templates: optometrists.html, optometrist-details.html.

Class Name: PatientController	
Parent Class: N/A Subclasses: N/A	
Responsibilities: Handles HTTP requests, redirections, and renders appropriate Thymeleaf templates associated with patient resources. It delegates CRUD operations to the appropriate service layer and returns	Collaborators: - PatientsService: Handles business logic and invokes data interactions. - Model: Supplies data to Thymeleaf templates.

relevant outputs to the user.	<ul style="list-style-type: none"> - Optometrist: Represents the entity being managed. - BindingResult: Captures validation results. - Thymeleaf templates: optometrists.html, optometrist-details.html.
-------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Services:

Interface Name: OptometristService	
Parent Class: N/A Subclasses: OptometristServiceImpl (implementation)	
Responsibilities: Provides an abstraction layer for optometrist resource CRUD operations and decouples the controller from concrete service implementations.	Collaborators: <ul style="list-style-type: none"> - OptometristController: Relies on this interface to execute business logic. - OptometristServiceImpl: Implements the interface methods and is injected into the controller.

Interface Name: PatientService	
Parent Class: N/A Subclasses: PatientServiceImpl (implementation)	
Responsibilities: Provides an abstraction layer for patient resource CRUD operations and decouples the controller from concrete service implementations.	Collaborators: <ul style="list-style-type: none"> - PatientController: Relies on this interface to execute business logic. - PatientServiceImpl: Implements the interface methods and is injected into the controller.

Service Implementations:

Class Name: OptometristServiceImpl	
Parent Class: OptometristService Subclasses: N/A	
Responsibilities: Provides concrete implementation for optometrist resource CRUD operations.	Collaborators: <ul style="list-style-type: none"> - OptometristRepository: Carries out the database operations requested by this class. - OptometristService: Provides the blueprint of the business logic that must be implemented.

Class Name: PatientServiceImpl	
Parent Class: PatientService Subclasses: N/A	
Responsibilities: Provides concrete implementation for patient resource CRUD operations.	Collaborators: - PatientRepository: Carries out the database operations requested by this class. - PatientService: Provides the blueprint of the business logic that must be implemented.

Repositories:

Interface Name: OptometristRepository	
Parent Class: JpaRepository<Optometrist, Integer> Subclasses: N/A	
Responsibilities: Provides database access methods for optometrist CRUD.	Collaborators: - OptometristServiceImpl: Uses this repository from Spring Data JPA to interact with the database. - Optometrist: The entity managed by this repository.

Interface Name: PatientRepository	
Parent Class: JpaRepository<Patient, Integer> Subclasses: N/A	
Responsibilities: Provides database access methods for patient CRUD.	Collaborators: - PatientServiceImpl: Uses this repository from Spring Data JPA to interact with the database. - Patient: The entity managed by this repository.

Models:

Class Name: Optometrist	
Parent Class: N/A Subclasses: N/A	
Responsibilities: Represents the optometrist entity with fields for storing personal and professional information. Provides a data model for database persistence and business logic interactions.	Collaborators: - OptometristRepository: Maps this entity to the database and performs CRUD operations. - OptometristServiceImpl: Uses this model to implement business logic. - OptometristController: Passes optometrist data between the front-end and back-end.

Class Name: Patient	
Parent Class: N/A Subclasses: N/A	
Responsibilities: Represents the patient entity with fields for storing personal and professional information. Provides a data model for database persistence and business logic interactions.	Collaborators: - PatientRepository: Maps this entity to the database and performs CRUD operations. - PatientServiceImpl: Uses this model to implement business logic. - PatientController: Passes patient data between the front-end and back-end.

Class Name: VisualAssessment	
Parent Class: N/A Subclasses: N/A	
Responsibilities: Represents the visual assessment entity with fields for storing relevant information. Provides a data model for database persistence and business logic interactions.	Collaborators: - NavController: Invokes the template associated with this model. - More collaborators will be implemented in the next sprint.

Software Architecture Diagram

