**Design Model Document**

**for**

**+Health**

## Version 1.0 Approved

**Prepared by:**

Cindy Torres

Ricardo Luna

Giovanni Guzmán

**10/23/19**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 10/23/2019 | 1.0 | Design Model Document 1.0 | Ricardo Luna Cindy Torres Giovanni Guzmán |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. Introduction 4
   1. Purpose 4
   2. Scope 4
   3. Definitions, Acronyms and Abbreviations 4
   4. References 5
   5. Overview 5
2. Overall Description 5

2.1 Assumptions and Dependencies 5

1. Sequence Diagrams 6

3.1 Create an Account 7

3.2 View Patient’s Medical History 8

3.3 Create Patient’s Diagnosis 9

3.4 Create Patient’s Treatment 10

3.5 Upload Medical Studies 11

1. Class Diagrams 12

4.1 Create an Account 13

4.2 View Patient’s Medical History 14

4.3 Create Patient’s Diagnosis 15

4.4 Create Patient’s Treatment 15

4.5 Upload Medical Studies 16

1. Package Diagrams 17

5.1 +Health Package Diagram 17

1. Design Review 18

6.1 Design Checklist 18

1. **Introduction**

A Design Model Document is an object-based picture or pictures that represent the use cases for a system. It’s the means to describe a system's implementation and source code in a diagrammatic fashion. The initial work when developing the design model is to adapt to the actual implementation environment. The logical analysis model will now be adapted to reality - the implementation environment.

The aim of this document is to record the results of the system design process and to describe how the system satisfies the requirements.

* 1. **Purpose**

The purpose of the Design Model is to transform the requirements into complete and detailed system design specifications, to show an abstraction of the implementation of the system and to serve as a comprehensive, composite artifact encompassing all design classes, subsystems, packages, collaborations, and the relationships between them.

* 1. **Scope**

The purpose of +Health is to ease patients’ medical information management and to create a convenient and easy-to-use application for doctors, trying to have all their patients’ medical information in a handy way. Above all, it is expected for the application to provide a comfortable user experience.

* 1. **Definitions, Acronyms and Abbreviations**
* Patient’s Medical Information: Medical general information of the patient such as height, weight, chronic diseases and clinical information.
* Patient’s Medical History: Data related to a person’s medical history, including symptoms, diagnoses, procedures, and outcomes. It also includes patient histories, lab results, x-rays, and so on.
* Patient’s Code. Unique identifier that the system assigns to a patient.  
  1. **References**

Software Requirements Specification. (2019). 2nd ed. +Health.

Analysis Model Document. (2019). 1st ed. +Health.

* 1. **Overview**

The remaining sections of this document specify the construction details of the system, each system component’s interaction with other components and external systems, and the interface that allows end users to operate the system and its functions.

1. **Overall Description**

What is +Health? +Health is an application that pretends to gather all the patients’ medical information such as chronic diseases, medical analysis, prescriptions, treatments, diagnoses and operations.

+Health purpose is to eliminate the lack of information of a patient on their medical history by having a centralized system that: 1) is able to access medical information at any given moment. 2) give access to a third party when needed. It will give a much wider and concrete insight on a patient (speaking in medical terms).

The patient will be able to access any of their medical diagnoses and treatments, mitigating misunderstandings between the parties involved. This will also improve the verification of a diagnoses and treatments from a third party, since all the information will be presented directly through the system.

Having a centralized system with all the medical history of a patient will keep misinterpretations to a minimum.

**2.1 Assumptions and Dependencies**

This system does not have any kind of dependencies.

1. **Sequence Diagrams**

This section contains the sequence diagrams of +Health. A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

The purposes of a sequence diagrams are:

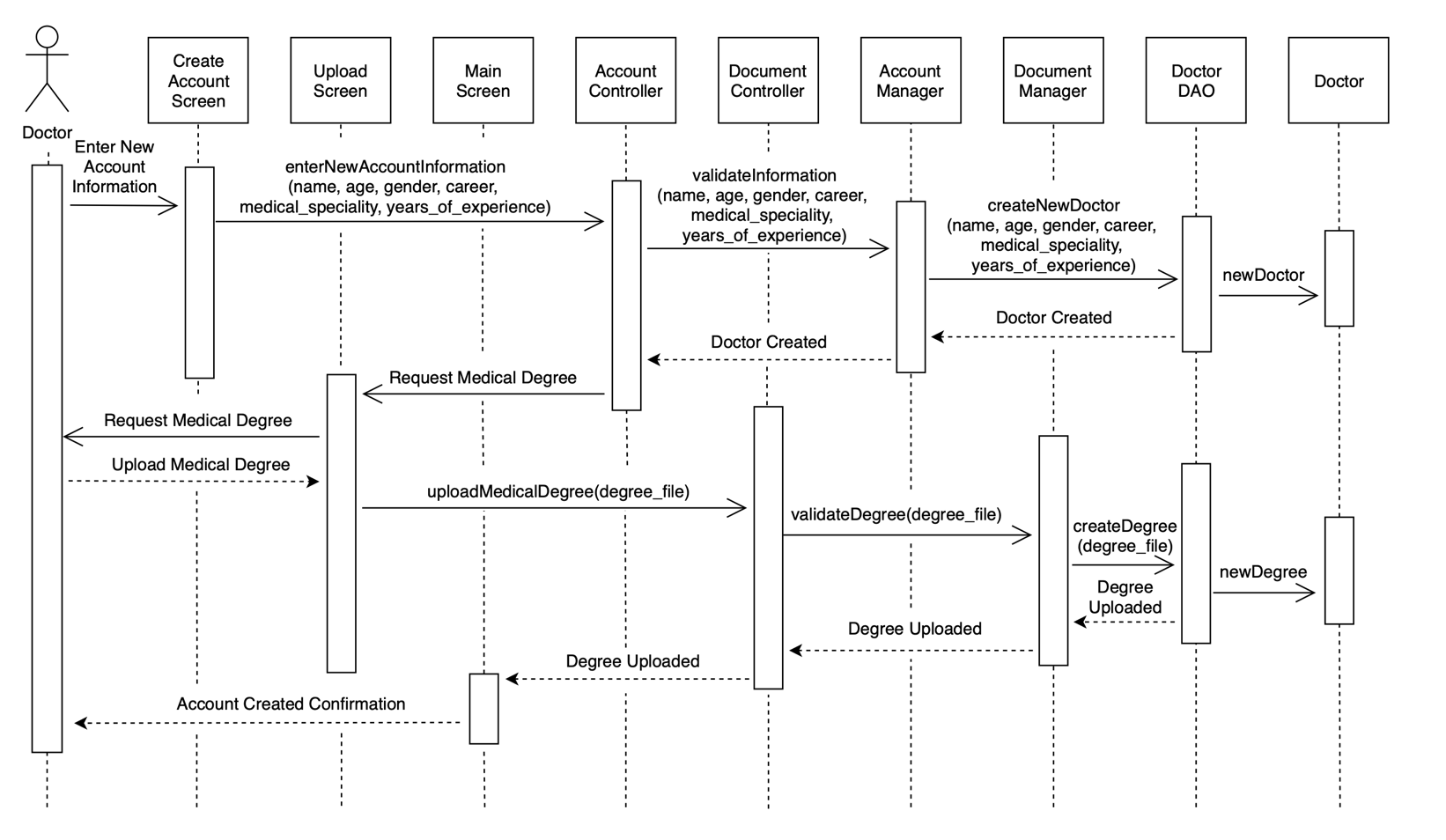
* Model high-level interaction between active objects in a system.
* Model the interaction between objects instances within a collaboration that realizes a use case.
* Model the interaction between objects within a collaboration that realizes an operation.
* Either model generic interactions (showing all possible paths through the interaction) or specific instances of an interaction (showing just one path through the interaction).

Sequence diagrams show elements as they interact over time and they are organized according to object (horizontally) and time (vertically):

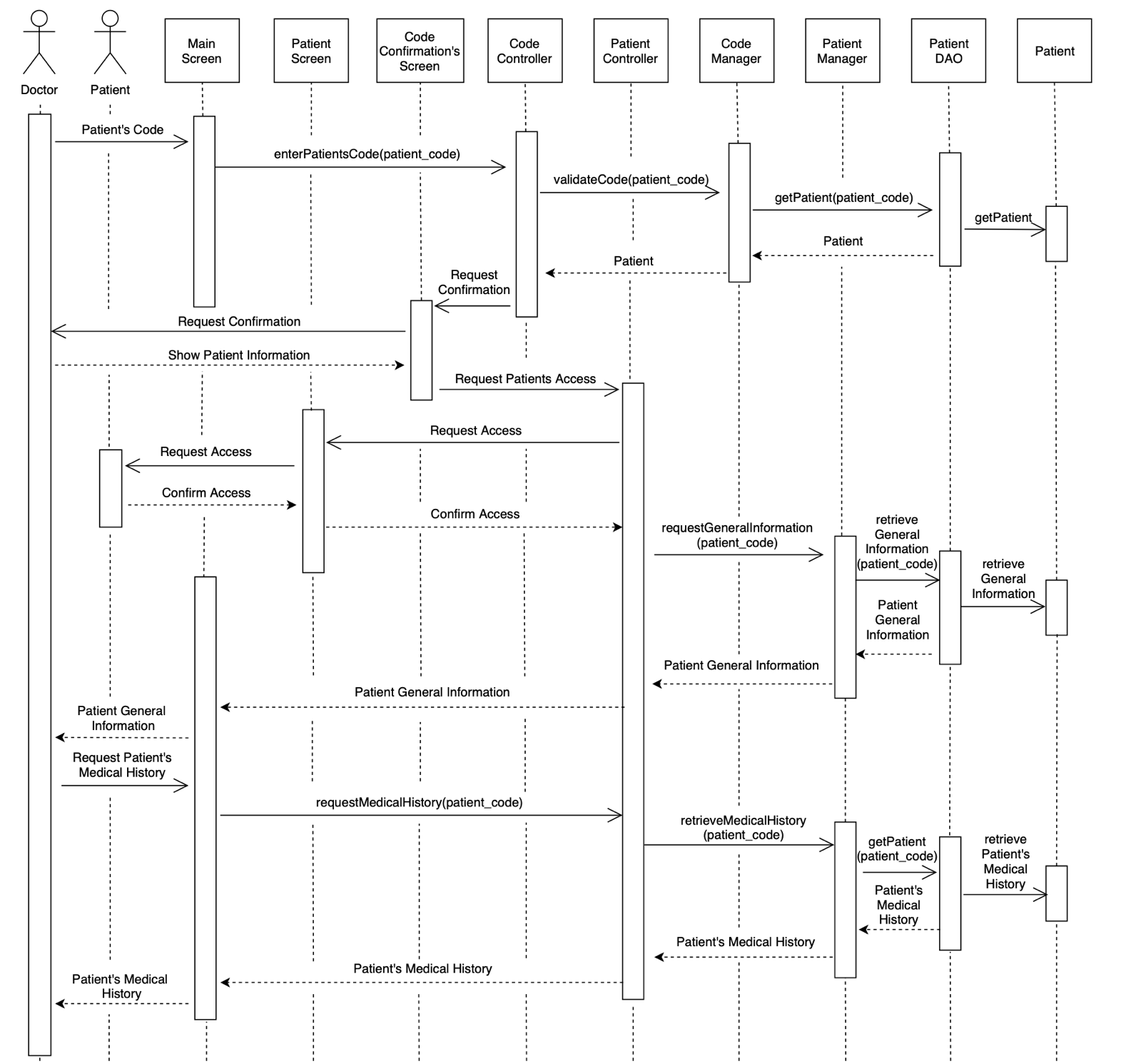
**Object Dimension:**

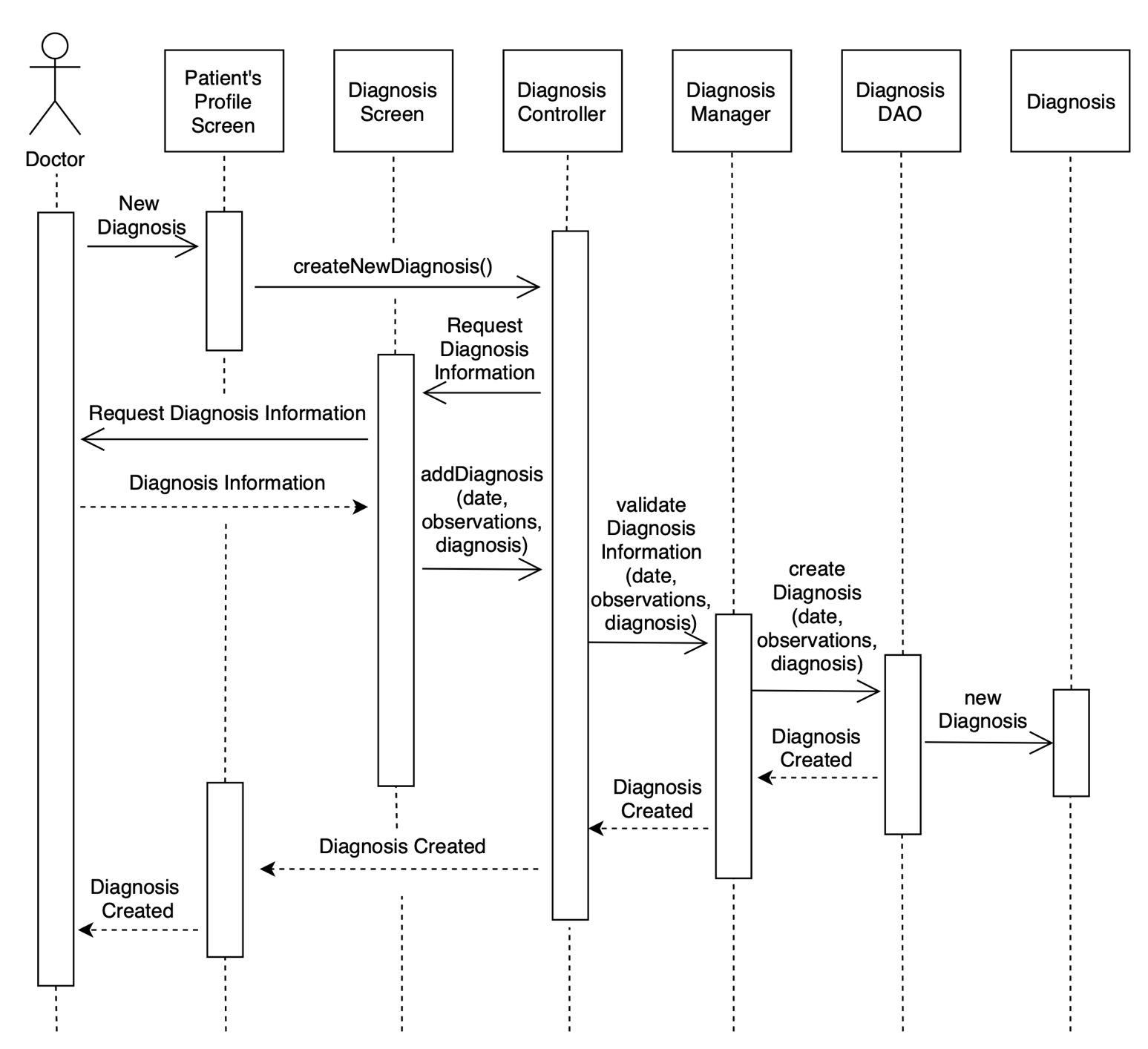
* The horizontal axis shows the elements that are involved in the interaction.
* Conventionally, the objects involved in the operation are listed from left to right according to when they take part in the message sequence. However, the elements on the horizontal axis may appear in any order.

**Time Dimension:**

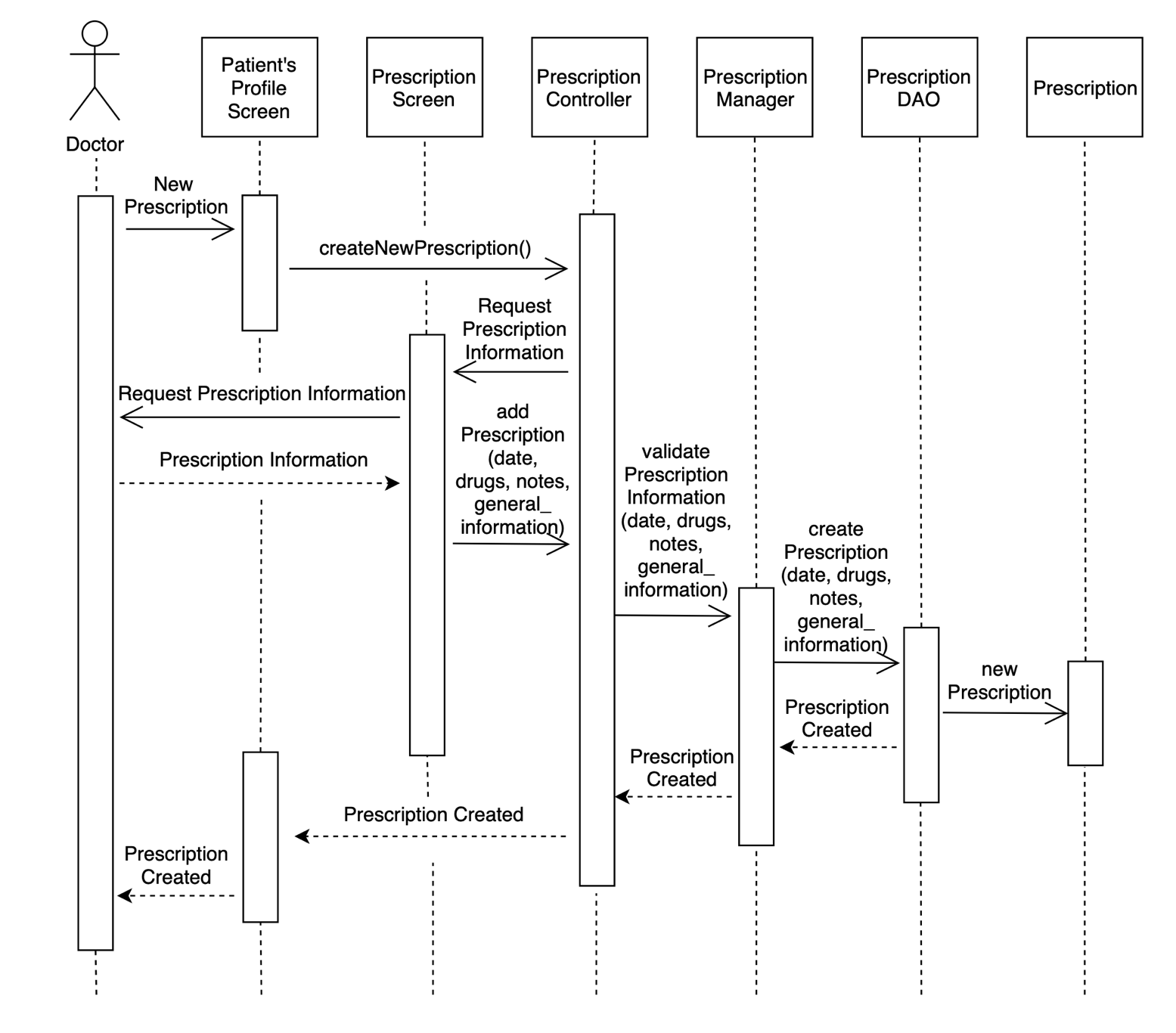
* The vertical axis represents time proceedings (or processing) down the page.
  1. ** Create an Account**

**0-9p**

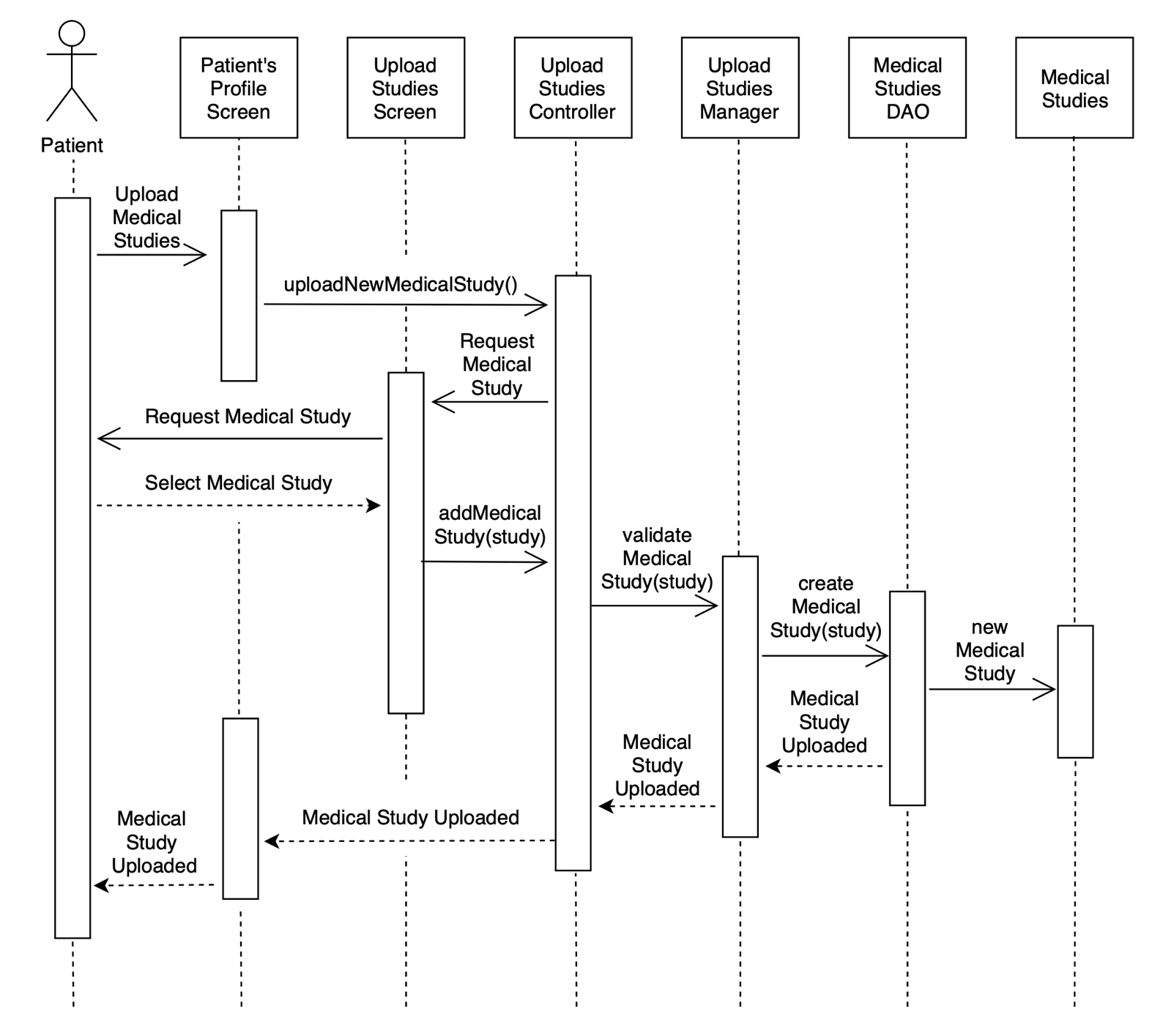
* 1. ** View Patient’s Medical History**
  2. **Create Patient’s Diagnosis**

****

* 1. **Create Patient’s Treatment**

****

* 1. **Upload Medical Studies**



1. **Class Diagrams**

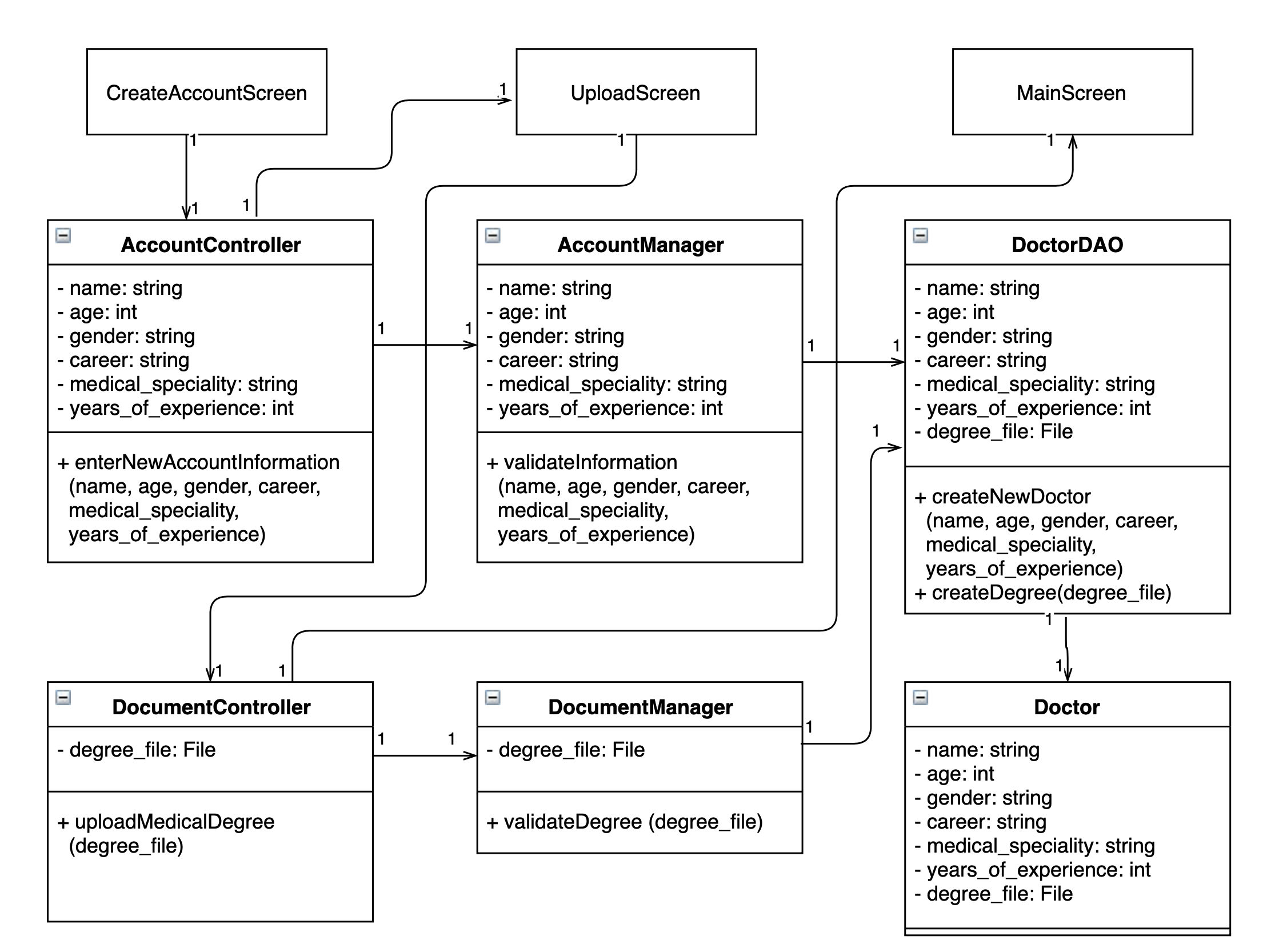
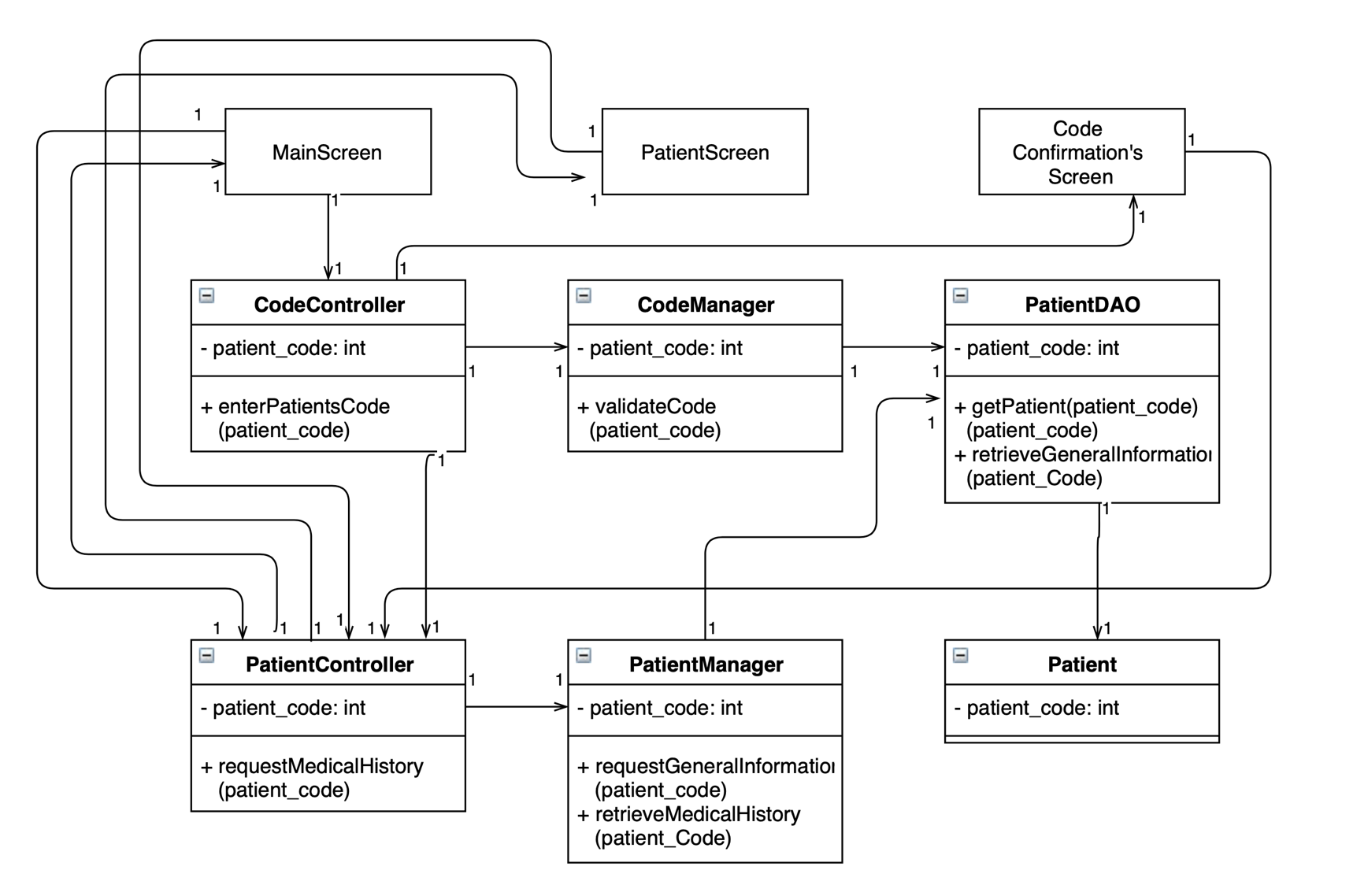
This section contains the class diagrams of +Health. A class diagram is a type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, operations (or methods), and the relationships among objects.

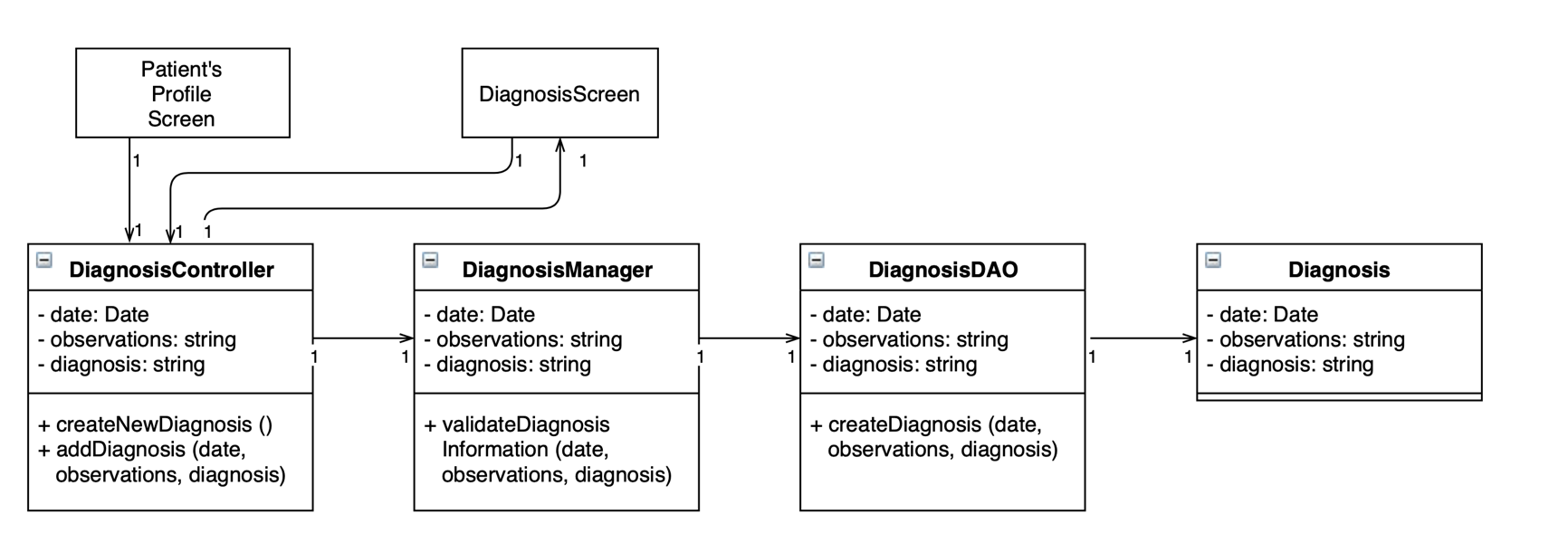
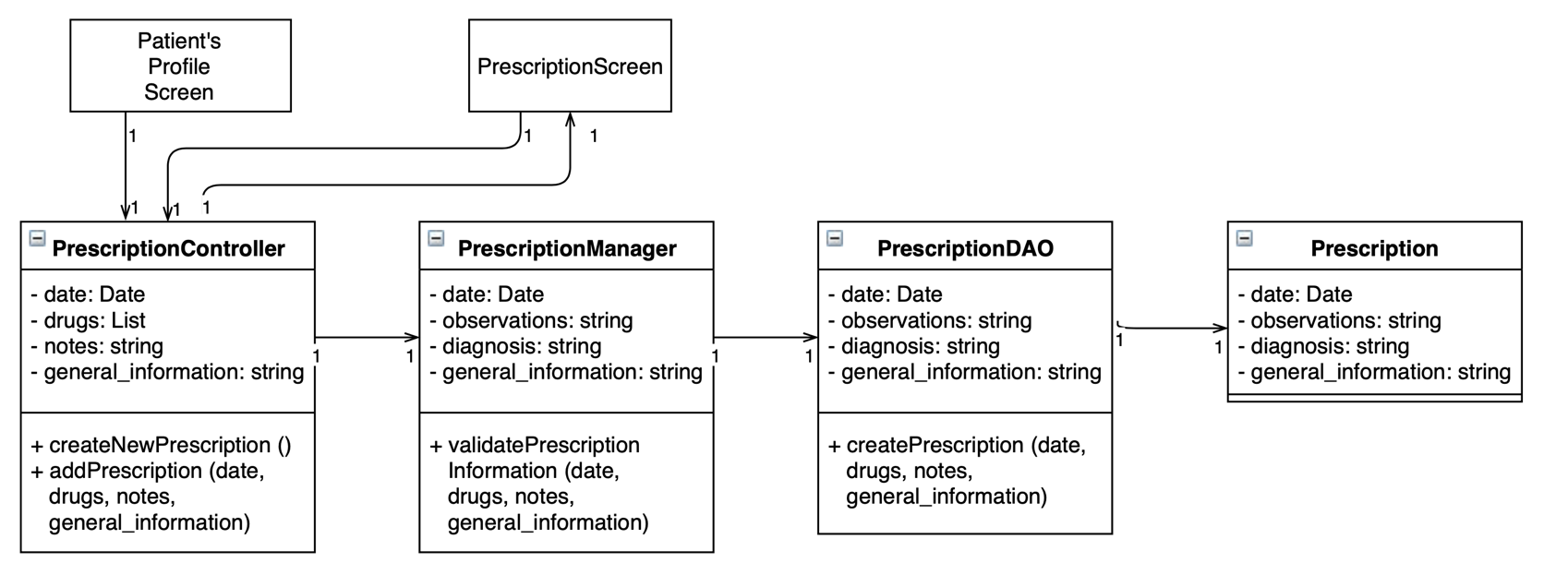
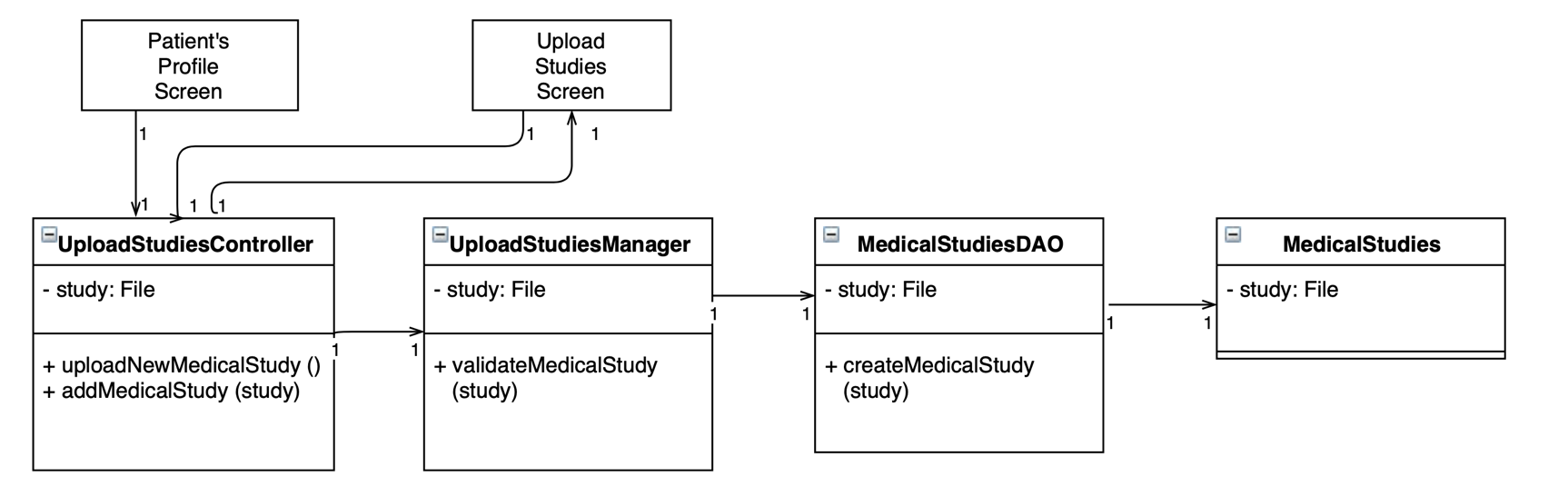
The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

* The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
* The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.
* The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

In the design of a system, a number of classes are identified and grouped together in a class diagram that helps to determine the static relations between them. With detailed modeling, the classes of the conceptual design are often split into a number of subclasses.

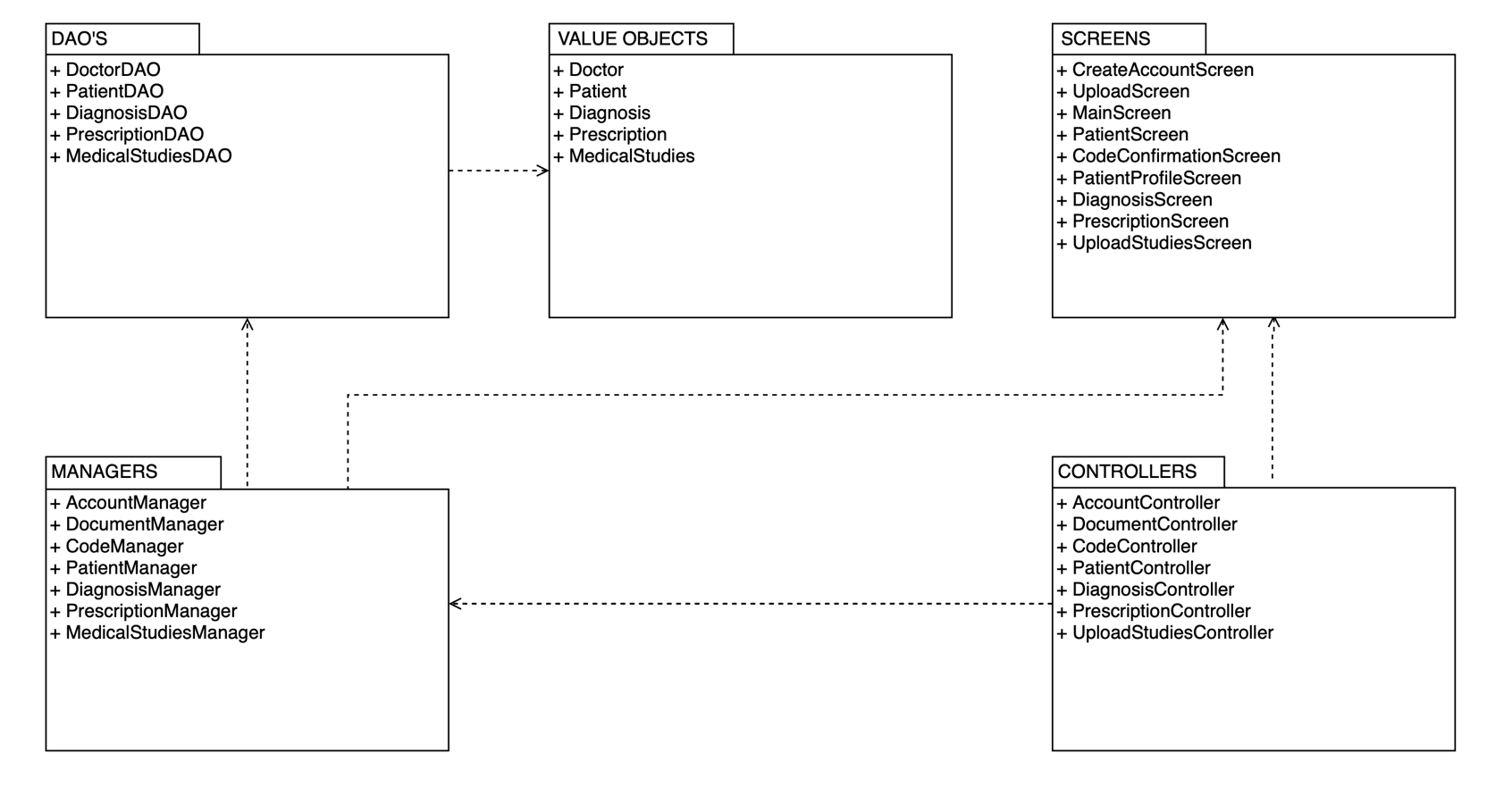
* 1.  **Create an Account**
  2.  **View Patient’s Medical History**

* 1.  **Create Patient’s Diagnosis**
  2. ** Create Patient’s Treatment**
  3. ** Upload Medical Studies**

1. **Package Diagrams**

This section contains the package diagrams of +Health. A package diagram, shows the arrangement and organization of model elements in middle to large scale projects. These can show both structure and dependencies between subsystems or modules, showing different views of a system.

Package diagrams are used to structure high level system elements. Packages are used for organizing large systems which contains diagrams, documents, and other key deliverables.

* Package diagram can be used to simplify complex class diagrams, it can group classes into packages.
* A package is a collection of logically related UML tools.
* Packages are despite as file folders and can be used on any of the UML diagrams.
  1.  **+Health Package Diagram**

1. **Design Review**

A design review is a milestone within a product development process whereby a design is evaluated against its requirements in order to verify the outcomes of previous activities and identify issues before committing to further work. The ultimate design review, if successful, therefore triggers the product launch or product release.

The purposes of design review are:

* To verify that the design model fulfills the requirements on the system, and that it serves as a good basis for its implementation.
* To ensure that the design model is consistent with respect to the general design guidelines.
* To ensure that the design guidelines fulfill their objectives.
  1. **Design Checklist**

The name of each package is unique and descriptive of the collective responsibilities of the model elements which it contains.

The publicly visible classes of the package provide a single, logically consistent set of services.

The dependencies between the package and other packages are consistent with relationships between contained classes.

The associations between the classes are correctly related.

The total number of packages is proportional to the total number of model elements.

The description of the class clearly conveys the purpose of the class

The class satisfies the behavioral requirements established by the use-case realizations.

All responsibilities of the class are related, such that it is not possible for the class to exist in a system where some of its responsibilities are used, but not others.

No two classes have essentially the same purpose.

The name of each operation is descriptive and understandable.

The parameters of each operation are correct in terms of both number, name and type.

Each operation is used by at least one use-case realization.

The name of each attribute is descriptive, and correctly conveys the information it stores.

The model is as simple as possible while still achieving the goals of the model.

The design is appropriate to task at hand (neither too complex nor too advanced).

The class satisfies the behavioral requirements established by the use-case realizations

Classes maintain a high cohesion (single responsibility principle).

Classes maintain a low coupling, increasing reuse.

Behavior has been distributed to the correct model elements, taking into consideration the responsibilities of the model elements.

Each class represents a small, consistent and unique set of responsibilities.