**MINISTRY OF EDUCATION AND TRAINING**

|  |
| --- |
| **FPT UNIVERSITY** |
| **Capstone Project Document** |
| **Olives health care system** |
|  |
| |  |  | | --- | --- | | **TrendyPlus Team** | | | **Group Member** | * Nguyễn Duy Linh (SE03150) * Bùi Đức Trọng * Trần Ánh Dương (SE02797) * Phạm Đức Thắng * Phạm Minh Tuấn * Nguyễn Văn Hưng | | **Supervisor** | Lecturer : | | **Capstone Project code** | Olives |   Hanoi, 24/4/2015 |

**Contents**

[Acknowledgements 9](#_Toc459493925)

[Definitions and Acronyms 10](#_Toc459493926)

[CHAPTER 4: SOFTWARE DESIGN 11](#_Toc459493927)

[1. Purpose 11](#_Toc459493928)

[2. Architecture Overview 11](#_Toc459493929)

[2.1. System Architecture 11](#_Toc459493930)

[2.2. System Architecture Explanation 12](#_Toc459493931)

[3. Architecture Design 12](#_Toc459493932)

[3.1. Olives Administration sub-system 12](#_Toc459493933)

[i. API 12](#_Toc459493934)

[3.2. Olives sub-system 14](#_Toc459493935)

[ii. API layer 14](#_Toc459493936)

[4. Database Design 17](#_Toc459493937)

[4.1. ERD 17](#_Toc459493938)

[4.2. Database dictionary 17](#_Toc459493939)

[4.2.1. Person 17](#_Toc459493940)

[4.2.2. Patient 18](#_Toc459493941)

[4.2.3. Doctor 18](#_Toc459493942)

[4.2.4. Place 18](#_Toc459493943)

[4.2.5. Specialty 18](#_Toc459493944)

[4.2.6. Diary 19](#_Toc459493945)

[4.2.7. Rating 19](#_Toc459493946)

[4.2.8. Relationship 19](#_Toc459493947)

[4.2.9. RelationshipRequest 20](#_Toc459493948)

[4.2.10. Allergy 20](#_Toc459493949)

[4.2.11. Addiction 20](#_Toc459493950)

[4.2.12. BloodPressure 21](#_Toc459493951)

[4.2.13. Heartbeat 21](#_Toc459493952)

[4.2.14. BloodSugar 21](#_Toc459493953)

[4.2.15. MedicalRecord 22](#_Toc459493954)

[4.2.16. Prescription 22](#_Toc459493955)

[4.2.17. PrescriptionImage 23](#_Toc459493956)

[4.2.18. MedicalNote 23](#_Toc459493957)

[4.2.19. MedicalCategory 24](#_Toc459493958)

[4.2.20. MedicalImage 24](#_Toc459493959)

[4.2.21. MedicalRecord 24](#_Toc459493960)

[4.2.22. Notification 25](#_Toc459493961)

[4.2.23. RealtimeConnection 26](#_Toc459493962)

[4.2.24. Message 26](#_Toc459493963)

[4.2.25. AccountToken 26](#_Toc459493964)

[4.2.26. Appointment 27](#_Toc459493965)

[5. References 28](#_Toc459493966)

[6. Detail design 28](#_Toc459493967)

[6.1. General sequence diagram 28](#_Toc459493968)

[6.1.1 Olives back-end general sequence diagrams 28](#_Toc459493969)

[6.1.1.1. Find record 28](#_Toc459493970)

[6.1.1.2. Initialize record 28](#_Toc459493971)

[6.1.1.3. Edit record 29](#_Toc459493972)

[6.1.1.4. Delete record 29](#_Toc459493973)

[6.1.1.5. Filter record 30](#_Toc459493974)

[6.1.2. Olives front-end general sequence diagrams (Patient application) 30](#_Toc459493975)

[6.1.2.1. Non-medical records data sequence 30](#_Toc459493976)

[6.1.2.2. Medical records data sequence 31](#_Toc459493977)

[6.1.3. Example of ordinary sequence diagrams 31](#_Toc459493978)

[6.1.3.1. Initialize blood pressure note 31](#_Toc459493979)

[6.1.3.3. Initialize appointment 32](#_Toc459493980)

[6.2. Detail design & use case 32](#_Toc459493981)

[6.2.1. Olives Administration sub-system 32](#_Toc459493982)

[6.2.1.1. Login 32](#_Toc459493983)

[a. Class diagram 32](#_Toc459493984)

[b. Sequence diagram 33](#_Toc459493985)

[c. Class definition 34](#_Toc459493986)

[ApiParentController class 34](#_Toc459493987)

[*[Method]RetrieveValidationErrors()* 34](#_Toc459493988)

[AdminController class 34](#_Toc459493989)

[*Attributes* 34](#_Toc459493990)

[*[Method]LoginAsync()* 35](#_Toc459493991)

[*[Method]EditAdminProfileAsync()* 35](#_Toc459493992)

[RepositoryAccountExtended class 35](#_Toc459493993)

[*Attributes* 35](#_Toc459493994)

[*[Method]FilterDoctorAsync()* 36](#_Toc459493995)

[*[Method]FilterPatientsAsync()* 36](#_Toc459493996)

[*[Method]SummarizePersonRoleAsync()* 36](#_Toc459493997)

[6.2.1.2. Logout 37](#_Toc459493998)

[a. Sequence diagram 37](#_Toc459493999)

[6.2.1.3. Create medical category 37](#_Toc459494000)

[a. Class diagram 37](#_Toc459494001)

[b. Class definition 38](#_Toc459494002)

[MedicalCategoryController class 38](#_Toc459494003)

[Attribute 38](#_Toc459494004)

[Method: InitializeMedicalCategory() 38](#_Toc459494005)

[Method: ModifyMedicalCategory() 39](#_Toc459494006)

[Method: FilterCategories() 39](#_Toc459494007)

[6.2.1.4. Edit medical category 40](#_Toc459494008)

[a. Class diagram 40](#_Toc459494009)

[6.2.1.5. Filter medical category 40](#_Toc459494010)

[a. Class diagram 40](#_Toc459494011)

[6.2.1.6. Activate account 40](#_Toc459494012)

[a. Class diagram 40](#_Toc459494013)

[b. Sequence diagram 41](#_Toc459494014)

[6.2.1.7. Find doctor 42](#_Toc459494015)

[a. Class diagram 42](#_Toc459494016)

[b. Class definition 43](#_Toc459494017)

[DoctorController class 43](#_Toc459494018)

[Attribute 43](#_Toc459494019)

[Method: Get() 44](#_Toc459494020)

[Method: Filter() 44](#_Toc459494021)

[6.2.1.8. Filter doctors 44](#_Toc459494022)

[a. Class diagram 44](#_Toc459494023)

[6.2.1.9. Find patients 45](#_Toc459494024)

[a. Class diagram 45](#_Toc459494025)

[6.2.1.10. Statistic accounts by role 46](#_Toc459494026)

[a. Class diagram 46](#_Toc459494027)

[b. Sequence diagram 47](#_Toc459494028)

[c. Class diagram 47](#_Toc459494029)

[PersonController class 48](#_Toc459494030)

[Attribute 48](#_Toc459494031)

[Method: Status() 48](#_Toc459494032)

[6.2.1.11. Initialize place 48](#_Toc459494033)

[a. Class diagram 48](#_Toc459494034)

[b. Class definition 49](#_Toc459494035)

[PlaceController class 49](#_Toc459494036)

[Attribute 49](#_Toc459494037)

[Method: FindPlace() 49](#_Toc459494038)

[Method: InitializePlace() 50](#_Toc459494039)

[Method: ModifyPlace() 50](#_Toc459494040)

[Method: FilterPlace() 50](#_Toc459494041)

[6.2.1.12. Edit place 51](#_Toc459494042)

[a. Class diagram 51](#_Toc459494043)

[6.2.2. Olives sub-system 51](#_Toc459494044)

[6.2.1.1. Login 51](#_Toc459494045)

[a. Sequence diagram 51](#_Toc459494046)

[b. Class diagram 52](#_Toc459494047)

[c. Class definition 52](#_Toc459494048)

[PlaceController class 52](#_Toc459494049)

[Attribute 53](#_Toc459494050)

[Method: FilterDoctorsAsync() 53](#_Toc459494051)

[Method: FilterPatientsAsync() 53](#_Toc459494052)

[Method: EditPersonProfileAsync() 53](#_Toc459494053)

[AccountController class 54](#_Toc459494054)

[Attribute 54](#_Toc459494055)

[Method: InitializeAvatarAsync() 54](#_Toc459494056)

[Method: SubmitLostPassword() 54](#_Toc459494057)

[Method: Login() 55](#_Toc459494058)

[LoginViewModel class 55](#_Toc459494059)

[Attribute 55](#_Toc459494060)

[6.2.1.2. Patient register account 55](#_Toc459494061)

[a. Class diagram 55](#_Toc459494062)

[b. Class definition 56](#_Toc459494063)

[InitializePatientViewModel class 56](#_Toc459494064)

[Attribute 56](#_Toc459494065)

[6.2.1.3. Patient register account 56](#_Toc459494066)

[a. Class diagram 56](#_Toc459494067)

[b. Class definition 57](#_Toc459494068)

[InitializePatientViewModel class 57](#_Toc459494069)

[Attribute 57](#_Toc459494070)

[6.2.1.4. Forgot password 57](#_Toc459494071)

[a. Sequence diagram 57](#_Toc459494072)

[<TODO> Fill fill fill 58](#_Toc459494073)

[6.2.1.5. Request change password 58](#_Toc459494074)

[a. Class diagram 58](#_Toc459494075)

[b. Sequence diagram 58](#_Toc459494076)

[c. Class definition 58](#_Toc459494077)

[PatientController class 58](#_Toc459494078)

[Attribute 58](#_Toc459494079)

[Method: FindPatientAsync() 59](#_Toc459494080)

[Method: InitializePatientAsync() 59](#_Toc459494081)

[Method: EditPatientAsync() 59](#_Toc459494082)

[Method: FilterPatientAsync() 60](#_Toc459494083)

[Method: RequestActivationCode() 60](#_Toc459494084)

[EditPatientViewModel class 60](#_Toc459494085)

[Attribute 60](#_Toc459494086)

[6.2.1.6. Find doctor 61](#_Toc459494087)

[a. Class diagram 61](#_Toc459494088)

[b. Class definition 61](#_Toc459494089)

[DoctorController class 61](#_Toc459494090)

[Attribute 61](#_Toc459494091)

[Method: FindDoctorAsync() 62](#_Toc459494092)

[Method: InitializeDoctorAsync() 62](#_Toc459494093)

[Method: EditDoctorAsync() 62](#_Toc459494094)

[Method: FilterDoctorAsync() 63](#_Toc459494095)

[6.2.1.7. Filter doctor rate 63](#_Toc459494096)

[a. Class diagram 63](#_Toc459494097)

[b. Class definition 64](#_Toc459494098)

[DoctorController class 64](#_Toc459494099)

[Attribute 64](#_Toc459494100)

[Method: InitializeRatingAsync() 64](#_Toc459494101)

[Method: FilterRatingAsync() 65](#_Toc459494102)

[RepositoryRating class 65](#_Toc459494103)

[Attribute 65](#_Toc459494104)

[Method: InitializeRatingAsync() 66](#_Toc459494105)

[Method: FilterRatingAsync() 66](#_Toc459494106)

[RepositoryRating class 66](#_Toc459494107)

[Attribute 66](#_Toc459494108)

[6.2.1.8. Initialize doctor rating 67](#_Toc459494109)

[a. Class diagram 67](#_Toc459494110)

[b. Class definition 68](#_Toc459494111)

[InitializeRatingViewModel class 68](#_Toc459494112)

[Attribute 68](#_Toc459494113)

[6.2.1.9. Patient finds personal profile 68](#_Toc459494114)

[a. Class diagram 68](#_Toc459494115)

[6.2.1.10. Patient edit profile 69](#_Toc459494116)

[a. Class diagram 69](#_Toc459494117)

[EditPatientProfileViewModel class 70](#_Toc459494118)

[Attribute 70](#_Toc459494119)

[6.2.1.11. Filter daily record 71](#_Toc459494120)

[a. Class diagram 71](#_Toc459494121)

[b. Class definition 72](#_Toc459494122)

[<name>Controller class 72](#_Toc459494123)

[Attribute 72](#_Toc459494124)

[Method: Get() 73](#_Toc459494125)

[Method: Post() 73](#_Toc459494126)

[Method: Put() 73](#_Toc459494127)

[Method: Delete() 74](#_Toc459494128)

[Method: Filter() 74](#_Toc459494129)

[Repository<name> class 74](#_Toc459494130)

[Attribute 74](#_Toc459494131)

[Method: Initialize<name>Async() 74](#_Toc459494132)

[Method: Find<name>Async () 75](#_Toc459494133)

[Method: Delete<name>Async () 75](#_Toc459494134)

[Method: Filter<name>Async () 75](#_Toc459494135)

[FilterHeatbeatViewModel class 75](#_Toc459494136)

[Attribute 75](#_Toc459494137)

[FilterAddictionViewModel class 76](#_Toc459494138)

[Attribute 76](#_Toc459494139)

[FilterAllergyViewModel class 77](#_Toc459494140)

[Attribute 77](#_Toc459494141)

[FilterBloodPressureViewModel class 77](#_Toc459494142)

[Attribute 78](#_Toc459494143)

[FilterBloodSugarViewModel class 78](#_Toc459494144)

[Attribute 78](#_Toc459494145)

[6.2.1.12. Edit daily record 79](#_Toc459494146)

[a. Class diagram 79](#_Toc459494147)

[b. Class definition 80](#_Toc459494148)

[EditHeartbeatViewModel class 80](#_Toc459494149)

[Attribute 80](#_Toc459494150)

[EditAddictionViewModel class 80](#_Toc459494151)

[Attribute 80](#_Toc459494152)

[EditAllergyViewModel class 80](#_Toc459494153)

[Attribute 80](#_Toc459494154)

[EditBloodPressureViewModel class 80](#_Toc459494155)

[Attribute 80](#_Toc459494156)

[EditBloodSugarViewModel class 81](#_Toc459494157)

[Attribute 81](#_Toc459494158)

[6.2.1.13. Delete daily record 81](#_Toc459494159)

[a. Class diagram 81](#_Toc459494160)

[6.2.1.14. Initialize daily record 82](#_Toc459494161)

[a. Class diagram 82](#_Toc459494162)

[b. Class definition 83](#_Toc459494163)

[InitializeHeartbeatViewModel class 83](#_Toc459494164)

[Attribute 83](#_Toc459494165)

[InitializeHeartbeatViewModel class 83](#_Toc459494166)

[Attribute 83](#_Toc459494167)

[InitializeAddictionViewModel class 84](#_Toc459494168)

[Attribute 84](#_Toc459494169)

[InitializeAllergyViewModel class 84](#_Toc459494170)

[Attribute 84](#_Toc459494171)

[InitializeBloodPressureViewModel class 84](#_Toc459494172)

[Attribute 84](#_Toc459494173)

[InitializeBloodPressureViewModel class 84](#_Toc459494174)

[Attribute 84](#_Toc459494175)

[6.2.1.15. View Prescription 85](#_Toc459494176)

[a. Class diagram 85](#_Toc459494177)

[b. Class definition 86](#_Toc459494178)

[<name>Controller class 86](#_Toc459494179)

[Method: Filter<name>Async () 86](#_Toc459494180)

[Method: Initialize<name>Async () 86](#_Toc459494181)

[Method: Edit<name>Async () 87](#_Toc459494182)

[Method: Delete<name>Async () 87](#_Toc459494183)

[Repository<name> class 87](#_Toc459494184)

[Method: Filter<name>Async () 87](#_Toc459494185)

[Method: Find<name>Async () 88](#_Toc459494186)

[Method: Delete<name>Async () 88](#_Toc459494187)

[Method: Initialize<name>Async () 88](#_Toc459494188)

[InitializePrescriptionViewModel class 89](#_Toc459494189)

[Attribute 89](#_Toc459494190)

[InitializePrescriptionImageViewModel class 89](#_Toc459494191)

[Attribute 89](#_Toc459494192)

[6.2.1.16. Edit Prescription 89](#_Toc459494193)

[a. Class diagram 89](#_Toc459494194)

[b. Class definition 90](#_Toc459494195)

[EditPrescriptionViewModel class 90](#_Toc459494196)

[Attribute 90](#_Toc459494197)

[6.2.1.17. Delete Prescription 90](#_Toc459494198)

[a. Class diagram 90](#_Toc459494199)

[6.2.1.18. View medical record 91](#_Toc459494200)

[a. Class diagram 91](#_Toc459494201)

[b. Class definition 93](#_Toc459494202)

[FilterMedicalNoteViewModel class 93](#_Toc459494203)

[Attribute 93](#_Toc459494204)

[FilterMedicalRecordViewModel class 94](#_Toc459494205)

[Attribute 94](#_Toc459494206)

[FilterMedicalImageViewModel class 95](#_Toc459494207)

[Attribute 95](#_Toc459494208)

[FilterExperimentNoteViewModel class 95](#_Toc459494209)

[Attribute 95](#_Toc459494210)

[6.2.1.19. Edit medical record 96](#_Toc459494211)

[a. Class diagram 96](#_Toc459494212)

[b. Class definition 96](#_Toc459494213)

[EditExperimentNote class 97](#_Toc459494214)

[Attribute 97](#_Toc459494215)

[EditMedicalNoteViewModel class 97](#_Toc459494216)

[Attribute 97](#_Toc459494217)

[EditMedicalRecordViewModel class 97](#_Toc459494218)

[Attribute 97](#_Toc459494219)

[6.2.1.20. Add medical record 97](#_Toc459494220)

[InitializeMedicalImageViewModel class 97](#_Toc459494221)

[Attribute 97](#_Toc459494222)

[InitializeMedicalNoteViewModel class 98](#_Toc459494223)

[Attribute 98](#_Toc459494224)

[InitializeMedicalRecordViewModel class 98](#_Toc459494225)

[Attribute 98](#_Toc459494226)

[6.2.1.21. Delete medical record 98](#_Toc459494227)

[6.2.1.22. Create appointment 100](#_Toc459494228)

[a. Class diagram 100](#_Toc459494229)

[b. Class definition 101](#_Toc459494230)

[AppointmentController class 101](#_Toc459494231)

[Attribute 101](#_Toc459494232)

[Method: FindAppointmentAsync() 102](#_Toc459494233)

[Method: InitializeAppointmentAsync() 102](#_Toc459494234)

[Method: EditAppointmentAsync() 102](#_Toc459494235)

[Method: FilterAppointmentAsync() 102](#_Toc459494236)

[AppointmentController class 103](#_Toc459494237)

[Attribute 103](#_Toc459494238)

[6.2.1.23. View appointment 103](#_Toc459494239)

[a. Class diagram 103](#_Toc459494240)

[b. Class definition 104](#_Toc459494241)

[AppointmentController class 104](#_Toc459494242)

[Attribute 104](#_Toc459494243)

[6.2.1.24. Accept appointment 105](#_Toc459494244)

[a. Class diagram 105](#_Toc459494245)

[EditAppointmentViewModel class 105](#_Toc459494246)

[Attribute 105](#_Toc459494247)

[6.2.1.25. Cancel appointment 106](#_Toc459494248)

[a. Class diagram 106](#_Toc459494249)

[6.2.1.26. Connect/disconnect to/from doctor 106](#_Toc459494250)

[RepositoryRelationship class 107](#_Toc459494251)

[Attribute 107](#_Toc459494252)

[Method: DeleteRelationship() 107](#_Toc459494253)

[Method: FilterRalatedDoctorAsync() 108](#_Toc459494254)

[Method: IsPeopleConnected() 108](#_Toc459494255)

[6.2.1.27. Chat 108](#_Toc459494256)

# Acknowledgements

*The success and final outcome of this project required a lot of guidance and assistance from many people and wish to express our deepest gratitude to them. Without them, this project would have been done with much more obstacles than it was.*

*First, we want to say thanks to Mr. Nguyen Tat Trung for his continuously sharing and motivating throughout the project. Mr. Nguyen Tat Trung has always been very responsive in providing necessary information, and without his generous support project would lack in accurate information on current developments. Under the instructions of Mr. Nguyen, the project team has always felt comfortable and confident. Apart from all the technologies and methodologies, Mr. Nguyen shared with us the philosophy of keeping the right attitude towards problems.*

*Second, we would like to say thanks to the instructors at FPT University for all the classes. Without the knowledge that university teaches us through each class, semester, we cannot finish this project.*

*Third, we truly appreciate the random fellows on the internet who has discussed with us. Every our tiny problem was answered clearly.*

*Finally, we want to thank our family for their help, inspiring during this project.*

# Definitions and Acronyms

|  |  |  |
| --- | --- | --- |
| **Acronym** | **Definition** | **Note** |
| SRS | Software Requirement Specification |  |
| API | Application Programming Interface |  |
| HTML | Hypertext Markup Language |  |
| CSS | Cascading Style Sheets |  |
| IDE | Integrated Development Environment |  |
| DBMS | Database Management System |  |
| UML | Unified Modeling Language |  |
| PM | Project Manager |  |
| UC | Use Case |  |
| N/A | Not Available |  |
| ORM | Object Relational Mapping |  |
| GUI | Graphical User Interface |  |
| HTTP | Hypertext Transfer Protocol |  |
| PHP | Hypertext Preprocessor |  |
| REST | Representational State Transfer |  |
| RAM | Random Access Memory |  |
| URL | Uniform Resource Locator |  |

# CHAPTER 4: SOFTWARE DESIGN

## Purpose

This chapter is to give the developer team a guidance of what the system’s architecture is, and how they should be implemented. This chapter consists of:

* Architecture overview
* Component diagram
* Detailed design
* Detailed description of components
* Database design

## Architecture Overview

### 2.1. System Architecture

<TODO: Import system data architecture>

**

*Figure 4-1: Olives System Architecture*

### 2.2. System Architecture Explanation

The Olives System is divided into 2 sub-systems hosted on Azure cloud services, use one SQL in common:

* Olives health care sub-system :
  + This sub-system is for ordinary users in the system: guests, doctors and patients. It includes 2 mobile applications and 1 web application and 1 API service.
    - Mobile applications for users who are doctors and patients.
      * These 2 applications are design for 2 specific roles: doctors and owners. They're built by using Objective C and run on iOS devices. These applications help users to be more flexible in using system. Messages and Notifications are sent instantly to bring Real-Time experience.
    - One web application for users of both role.
      * An application which has been built on AngularJS. It helps users interact and consume system services. It can be connected any time, any places by using browsers. Notifications are sent instantly to bring Real-Time experience.
    - API service:
      * A service which is built on ASP.NET Web API 2 framework and hosted on Azure cloud service. It takes responsibility in taking requests come from user applications, connecting to database and processing the received data and respond processed result to user applications.
* Olives administration system:
  + This sub-system is for administrators to manage system. It includes one web application and one API service application.
    - Web application: an application which helps administrators to manage users in system, to track their statuses, to activate/deactivate ordinary users …
    - API Service: A service which is built on ASP.NET Web API 2 framework and hosted on Azure cloud service. It takes responsibility in taking requests come from user applications, connecting to database and processing the received data and respond processed result to user applications.

## Architecture Design

### 3.1. Olives Administration sub-system

#### API

**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Namespace/Directory** | **Description** | **Convention** |
| 1 | Controllers | Contains all controller source files which receive requests come from clients, processs data and return results back to requesters. | <name>Controller |
| 2 | Interfaces | Contains all interfaces definition files which are inherited by another classes, such as repositories, services, ... | **I**<name> |
| 3 | Attributes | Contains all validation attributes source file which are used for data validation purposes. | <name>Attribute |
| 4 | Enumerations | Contains enumerations which are used in service. |  |
| 5 | Repositories | Contains classes which is used for access database to process data and return data to controller. Controllers don’t access database directly. | **Repository**<name> |
| 7 | Scripts | Contains javascript files which are used in Views. |  |
| 8 | ViewModels | Contains classes which define data structure sent from clients and vice versa. | <name>ViewModel |

*Table 4-1: Namespace Explanation*

### 3.2. Olives sub-system

#### API layer

*Figure 4-13: TrendyPlus Layer Design*

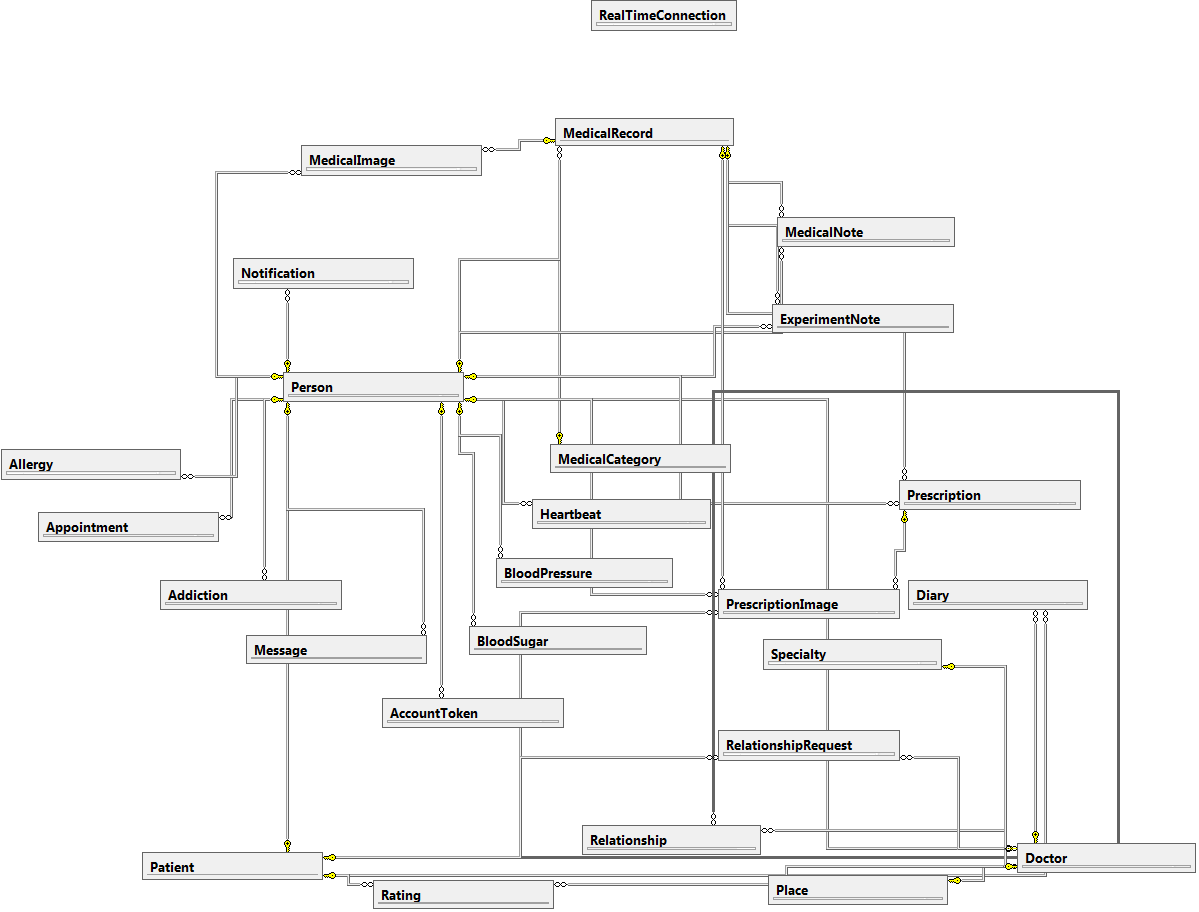


|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Namespace/Directory** | **Description** | **Convention** |
| 1 | Controllers | Contains all controller source files which receive requests come from clients, processs data and return results back to requesters. | <name>Controller |
| 2 | Interfaces | Contains all interfaces definition files which are inherited by another classes, such as repositories, services, ... | **I<name>** |
| 3 | Attributes | Contains all validation attributes source file which are used for data validation purposes. | <name>Attribute |
| 4 | Enumerations | Contains enumerations which are used in service. |  |
| 5 | Repositories | Contains classes which is used for access database to process data and return data to controller. Controllers don’t access database directly. | **Repository<name>** |
| 7 | Scripts | Contains javascript files which are used in Views. |  |
| 8 | ViewModels | Contains classes which define data structure sent from clients and vice versa. | <name>ViewModel |
| 9 | Views | Contains csthml files which are used for being displayed on client browsers. |  |

*Table 4-2: Namespace Explanation*

## Database Design

### 4.1. Table relationship



### 4.2. Table purposes

|  |  |  |
| --- | --- | --- |
| **No.** | **Table** | **Description** |
| 1 | Person | Stores general information of users in system. |
| 1 | Patient | Stores specific information of users who are patients in system. |
| 1 | Doctor | Stores specific information of users who are doctors in system. |
| 1 | Place | Stores information of places list in system where doctor lives or works. |
| 1 | Specialty | Stores information of specialties in which doctor operates. |
| 1 | Diary | Stores information of personal diary of a doctor about a specific patient. |
| 1 | Rating | Stores information of rating between patients and doctors. |
| 1 | Relationship | Stores information of connection between patients and doctors. |
| 1 | RelationshipRequest | Stores information of relationship requests from a patients to doctors. |
| 1 | Allergy | Stores personal notes of patients about their allergies. |
| 1 | Addiction | Stores personal notes of patients about their addictions. |
| 1 | BloodPressure | Stores personal notes of patients about their blood pressure measurements. |
| 1 | Heartbeat | Stores personal notes of patients about their heart rate measurements. |
| 1 | BloodSugar | Stores personal notes of patients about their blood sugars. |
| 1 | MedicalRecord | Stores medical records of patients. |
| 1 | Prescription | Store prescriptions of patients. |
| 1 | PrescriptionImage | Store images about prescriptions. |
| 1 | MedicalNote | Store notes about medical records. |
| 1 | MedicalCategory | Store list of categories for categorizing medical records. |
| 1 | MedicalImage | Store list of images related to medical records. |
| 1 | ExperimentNote | Store list of experiment notes of medical records. |
| 1 | Notification | Store list of notifications broadcasted from users to users. |
| 1 | RealTimeConnection | Store list of realtime connection indexes which are used for broadcasting notification messages. |
| 1 | Message | Store list of messages. |
| 1 | AccountToken | Store list of account tokens. |
| 1 | Appointment | Store list of appointment created by patients to doctor or vice versa. |

### 4.3. Table design

#### 4.3.1. Person table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | The id of user | No |
| 2 | Email | VARCHAR(64) |  | Email which is used for account registration. This can be understood as account name | No |
| 3 | Password | VARCHAR(16) |  | Password of user | No |
| 4 | FirstName | NVARCHAR(32) |  | User first name | No |
| 5 | LastName | NVARCHAR(32) |  | User last name | No |
| 6 | FullName | NVARCHAR(96) |  | Full name of user, this is the combination of First Name and Last Name. It is created for quick access. | No |
| 7 | Birthday | FLOAT |  | Time when user was born. Its format is UTC. | No |
| 8 | Phone | VARCHAR(15) |  | User phone number | Yes |
| 9 | Gender | TINYINT |  | User gender | No |
| 10 | Role | TINYINT |  | Role in the system: Admin, patient or doctor | No |
| 11 | Created | FLOAT |  | Time when account was signed up | No |
| 12 | LastModified | FLOAT |  | Time when the account information was lastly modified | Yes |
| 13 | Status | TINYINT |  | Status of account : pending, disabled or active | No |
| 14 | Address | NVARCHAR(128) |  | Address where user lives or works | Yes |
| 15 | PhotoUrl | NVARCHAR(MAX) |  | Relative path of user photo, this is used for render hyperlink. | Yes |
| 16 | PhotoPhysicPath | NVARCHAR(MAX) |  | The physical path where photo is stored, this is used for file access. | Yes |

#### 4.3.2. Patient table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK/FK | The id of patient. This is the same as person’s id | No |
| 2 | Height | FLOAT |  | Height of user | Yes |
| 3 | Weight | FLOAT |  | User weight | Yes |

#### 4.3.3. Doctor table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK/FK | The id of doctor. This is the same as person’s id | No |
| 2 | Rank | FLOAT |  | Doctor rank, this indicates how good the doctor is | Yes |
| 3 | SpecialtyId | INT | FK | Id of doctor specialty | No |
| 4 | Voters | INT |  | Number of people who voted for this doctor | Yes |
| 5 | PlaceId | INT | FK | Id of place where doctor lives or works | No |
| 6 | ProfileUrl | NVARCHAR(MAX) |  | Url of doctor profile file. | Yes |
| 7 | ProfilePhysicalPath | NVARCHAR(MAX) |  | Storage location on the server where real file is stored. | Yes |

#### 4.3.4. Place table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK/FK | The id of doctor. This is the same as person’s id | No |
| 2 | City | FLOAT |  | City name. | No |
| 3 | Country | INT | FK | Country name | No |

#### 4.3.5. Specialty table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id which is used for identifying specialties. | No |
| 2 | Name | NVARCHAR(32) |  | Specialty name | No |

#### 4.3.6. Diary table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Diary unique index. | No |
| 2 | Owner | INT | FK | Id of doctor who owns the diary. | No |
| 3 | Target | INT | FK | Id of patient who diary notes about. | No |
| 4 | Time | FLOAT |  | Time when the diary was noted. | No |
| 5 | Note | NVARCHAR(512) |  | Diary note. | No |
| 6 | Created | FLOAT |  | Time when the diary was created. | No |
| 7 | LastModified | FLOAT |  | Time when the diary was lastly modified. | No |

#### 4.3.7. Rating table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Maker | INT | PK/FK | Id of patient who made the rating, with Target, they combine to one primary key. | No |
| 2 | Target | INT | PK/FK | Id of doctor who was rated, with Maker, they combine to one primary key. | No |
| 3 | Value | INT |  | Rating point. | No |
| 4 | Comment | FLOAT |  | Comment of patient when rating was made. | No |
| 5 | Created | FLOAT |  | When the rating was created. | No |

#### 4.3.8. Relationship table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id of relationship. | No |
| 2 | Source | INT | FK | Id of patient who started the relationship. | No |
| 3 | Target | INT | FK | Id of doctor who accepted the relationship. | No |
| 4 | Created | FLOAT |  | When the relationship was created. | No |

#### 4.3.9. RelationshipRequest table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id of relationship. | No |
| 2 | Source | INT | FK | Id of patient who started the relationship request. | No |
| 3 | Target | INT | FK | Id of doctor who accepted the relationship request. | No |
| 4 | Content | NVARCHAR(128) |  | Message sent from patient to doctor about this request. | No |
| 5 | Created | FLOAT |  | When the relationship was sent. | Yes |

#### 4.3.10. Allergy table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id of allergy note. | No |
| 2 | Owner | INT | FK | Id of patient who owns the allergy note. | No |
| 3 | Name | INT | FK | Name of allergy. | No |
| 4 | Cause | NVARCHAR(128) |  | Which patient is allergic with. | No |
| 5 | Note | FLOAT |  | Note information of allergy. | Yes |
| 6 | Created | FLOAT |  | When the note was created. | No |
| 7 | LastModified | FLOAT |  | When the note was lastly modified. | Yes |

#### 4.3.11. Addiction table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id of addiction note. | No |
| 2 | Owner | INT | FK | Id of patient who owns the addiction note. | No |
| 3 | Cause | NVARCHAR(128) |  | Which patient is addictive with. | No |
| 4 | Note | FLOAT |  | Note information of addiction. | Yes |
| 5 | Created | FLOAT |  | When the note was created. | No |
| 6 | LastModified | FLOAT |  | When the note was lastly modified. | Yes |

#### 4.3.12. BloodPressure table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id of blood pressure note. | No |
| 2 | Owner | INT | FK | Id of patient who owns the note. | No |
| 3 | Systolic | INT |  | Max pressure of blood. | No |
| 4 | Diastolic | INT |  | Min pressure of blood. | No |
| 5 | Time | FLOAT |  | When the measurement was done. | No |
| 6 | Note | FLOAT |  | Note information of addiction. | Yes |
| 7 | Created | FLOAT |  | When the note was created. | No |
| 8 | LastModified | FLOAT |  | When the note was lastly modified. | Yes |

#### 4.3.13. Heartbeat table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id of blood pressure note. | No |
| 2 | Owner | INT | FK | Id of patient who owns the note. | No |
| 3 | Rate | FLOAT |  | Heart rate at the time measurement was done. | No |
| 4 | Time | FLOAT |  | When the measurement was done. | No |
| 5 | Note | FLOAT |  | Note information of addiction. | Yes |
| 6 | Created | FLOAT |  | When the note was created. | No |
| 7 | LastModified | FLOAT |  | When the note was lastly modified. | Yes |

#### 4.3.14. BloodSugar table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id of blood sugar note. | No |
| 2 | Owner | INT | FK | Id of patient who owns the note. | No |
| 3 | Value | FLOAT |  | Blood sugar level at the time measurement was done. | No |
| 4 | Time | FLOAT |  | When the measurement was done. | No |
| 5 | Note | FLOAT |  | Note information of addiction. | Yes |
| 6 | Created | FLOAT |  | When the note was created. | No |
| 7 | LastModified | FLOAT |  | When the note was lastly modified. | Yes |

#### 4.3.15. MedicalRecord table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id of medical record. | No |
| 2 | Owner | INT | FK | Who owned the record. | No |
| 3 | Creator | INT | FK | Who created medical record. | No |
| 4 | Category | INT |  | Category which medical record belongs to. | No |
| 5 | Name | NVARCHAR(64) |  | Name of medical record. | Yes |
| 6 | Info | NVARCHAR(MAX) |  | Key-value of necessary information.I.e : “Eye”:”good” | Yes |
| 7 | Time | FLOAT |  | When the record was about. | Yes |
| 8 | Created | FLOAT |  | When the record was created. | No |
| 9 | LastModified | FLOAT |  | When the note was lastly modified. | Yes |

#### 4.3.16. Prescription table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id of prescription. | No |
| 2 | Owner | INT | FK | Who owns the prescription. | No |
| 3 | Creator | INT | FK | Who created prescription. | No |
| 4 | MedicalRecordId | INT | FK | Id of medical record which prescription belongs to. | No |
| 5 | From | FLOAT |  | When the prescription should be available. | No |
| 6 | To | FLOAT |  | When the prescription should be expired | No |
| 7 | Name | NVARCHAR(32) |  | Name of prescription | No |
| 8 | Medicine | NVARCHAR(MAX) |  | Stores serialized information of medicine. Refer to API documentation for more detail (api/medical/prescription) | Yes |
| 9 | Note | NVARCHAR(128) |  | Note of prescription. | Yes |
| 10 | Created | FLOAT |  | When the prescription was created. | No |
| 11 | LastModified | FLOAT |  | When the prescription was lastly modified. | Yes |

#### 4.3.17. PrescriptionImage table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id of prescription image. | No |
| 2 | Owner | INT | FK | Who owns the prescription image. | No |
| 3 | Creator | INT | FK | Who created prescription image. | No |
| 4 | MedicalRecordId | INT | FK | Id of medical record which prescription image belongs to. | No |
| 5 | PrescriptionId | INT | FK | Id of prescription which prescription image belongs to. | No |
| 6 | Image | NVARCHAR(32) |  | Name of image. | No |
| 7 | FullPath | NVARCHAR(MAX) |  | Physical path where image is stored on server. | No |
| 8 | Created | FLOAT |  | When the prescription image was created. | No |
| 9 | LastModified | FLOAT |  | When the prescription image was lastly modified. | Yes |

#### 4.3.18. MedicalNote table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id of prescription image. | No |
| 2 | Owner | INT | FK | Who owns the prescription image. | No |
| 3 | Creator | INT | FK | Who created prescription image. | No |
| 4 | MedicalRecordId | INT | FK | Id of medical record which note belongs to. | No |
| 5 | Note | NVARCHAR(128) |  | Note of medical record. | No |
| 6 | Time | INT |  | Time when the note was about. | No |
| 7 | Created | FLOAT |  | When the note was created. | No |
| 8 | LastModified | FLOAT |  | When the note was lastly modified. | Yes |

#### 4.3.19. MedicalCategory table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id of prescription image. | No |
| 2 | Name | NVARCHAR(32) |  | Name of medical category. | No |
| 3 | Created | FLOAT |  | When the category was created. | No |
| 4 | LastModified | FLOAT |  | When the category was lastly modified. | Yes |

#### 4.3.20. MedicalImage table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id of .medical image. | No |
| 2 | MedicalRecordId | INT | FK | Id of medical record which medical image belongs to. | No |
| 3 | Owner | INT | FK | Id of person who owns the medical image. | No |
| 4 | Creator | INT | FK | Id of person who created medical image. | No |
| 5 | Image | NVARCHAR(32) |  | Name of image. | No |
| 6 | FullPath | NVARCHAR(MAX) |  | Physical path on server where image is stored. | No |
| 7 | Created | FLOAT |  | When medical image was created. | No |

#### 4.3.21. ExperimentNote table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id of of experiment note. | No |
| 2 | MedicalRecordId | INT | FK | Id of medical record which medical record belongs to. | No |
| 3 | Owner | INT | FK | Id of person who owns the experiment note. | No |
| 4 | Creator | INT | FK | Id of person who created experiment note. | No |
| 5 | Name | INT |  | Name of experiment note. | No |
| 6 | Info | INT |  | Stores serialized information of experiment measurement. | No |
| 7 | Time | INT |  | Time when the experiment note is about. | No |
| 8 | Created | FLOAT |  | When medical image was created. | No |
| 9 | LastModified | FLOAT |  | When medical image was lastly modified. | No |

#### 4.3.22. Notification table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id of of notification. | No |
| 2 | Type | INT |  | Type of notification : create/edit/remove/… | No |
| 3 | Topic | INT |  | Topic of notification : Medical record/ Prescription/… | No |
| 4 | Container | INT |  | Id of the container (Medical Record ID, Prescription ID, …). | No |
| 5 | ContainerType | INT |  | Type of container (Medical Record, Prescription, …). | No |
| 6 | Broadcaster | INT |  | Who causes the notification broadcasted. | No |
| 7 | Recipient | INT |  | Who should receive the notification. | No |
| 8 | Record | FLOAT |  | Id of record which is targeted. | No |
| 9 | Message | FLOAT |  | Message from service. | No |
| 10 | Created | FLOAT |  | When the notification is created. | No |
| 11 | IsSeen | FLOAT |  | Whether recipient has seen the notification or not. | No |

#### 4.3.23. RealtimeConnection table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id of of realtime connection. | No |
| 2 | Owner | INT |  | Id of person who connects to service. | No |
| 3 | ConnectionId | INT |  | Connection index which generated by service. | No |
| 4 | Created | INT |  | Time when the note was created. | No |

#### 4.3.24. Message table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK | Id of of message. | No |
| 2 | Broadcaster | INT | FK | Id of person who broadcasted the message. | No |
| 3 | Recipient | INT | FK | Id of person who receives the message. | No |
| 4 | Content | NVARCHAR(512) |  | Message content. | No |
| 5 | Created | FLOAT |  | Time when the message was broadcasted. | No |
| 6 | IsSeen | BIT |  | Whether message has been seen or not. | No |

#### 4.3.25. AccountToken table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Owner | INT | PK/FK | Id of patient who has the code. | No |
| 2 | Code | NVARCHAR(36) |  | Code for account activation/account password modification. | No |
| 3 | Type | INT | PK/FK | Id of person who receives the message. | No |
| 4 | Expired | DATETIME |  | Time when the account token is expired. | No |

#### 4.3.26. Appointment table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Field** | **Type** | **Key** | **Description** | **Null** |
| 1 | Id | INT | PK/FK | Appointment index. | No |
| 2 | Maker | INT | FK | Id of person who made the appointment request. | No |
| 3 | Dater | INT | FK | Id of person who received the appointment request. | No |
| 4 | From | FLOAT |  | Time when the appointment should be started. | No |
| 5 | To | FLOAT |  | Time when the appointment should be emded. | No |
| 6 | Note | NVARCHAR(128) |  | Note of appointment | No |
| 7 | Created | FLOAT |  | Time when the appointment was created. | No |
| 8 | LastModified | FLOAT |  | Time when the appointment was lastly modified. | No |
| 9 | LastModifiedNote | NVARCHAR(128) |  | Note of the time appointment was lastly modified. | Yes |
| 10 | Status | TINYINT |  | Status of appointment | No |

## References

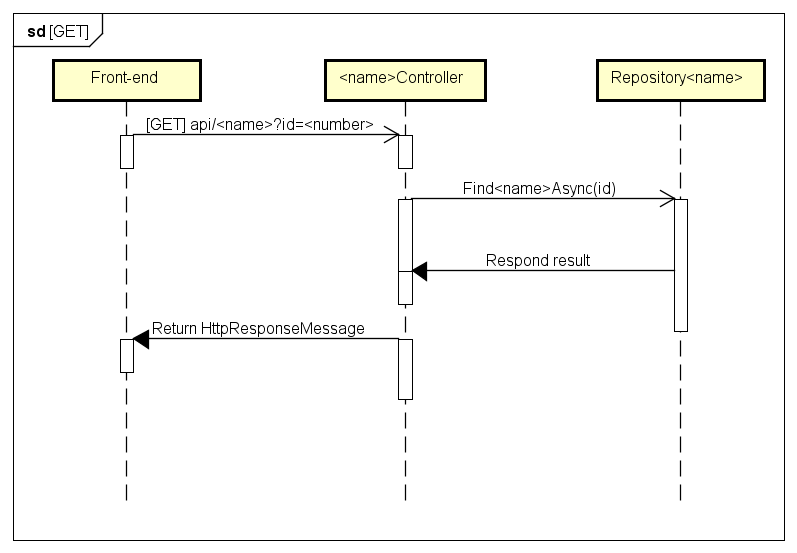
## Detail design

### 6.1. General sequence diagram

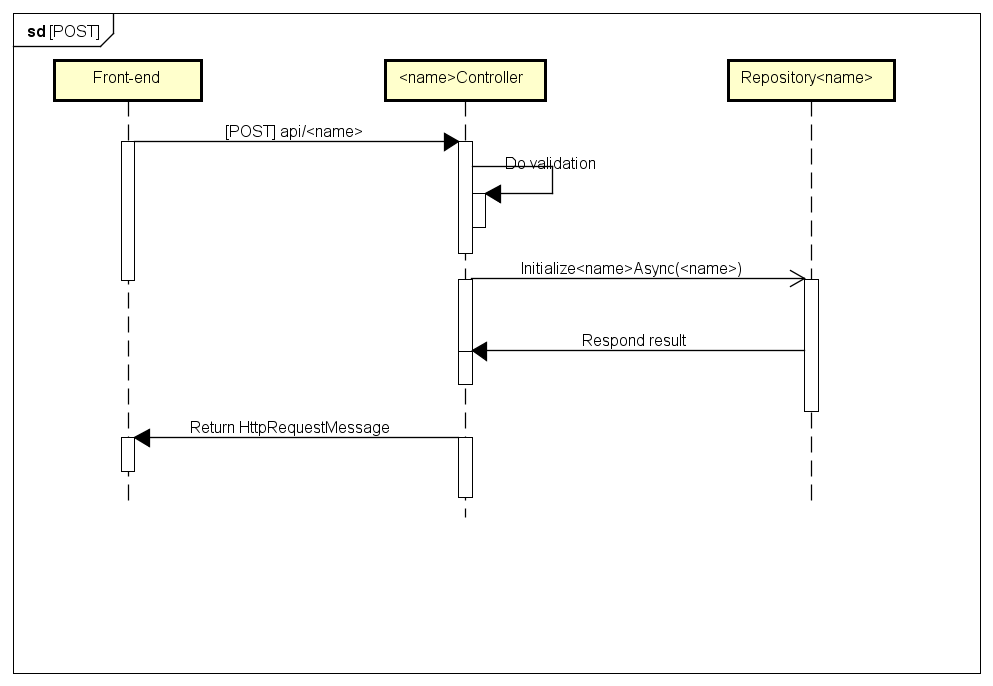
Olives back-end services and front-end applications all have some general sequence diagram which will be used by Components Assemblers in development stage. As in a specific function, sequence diagram is not defined, that function should follows the general sequence diagram.

#### 6.1.1 Olives back-end general sequence diagrams

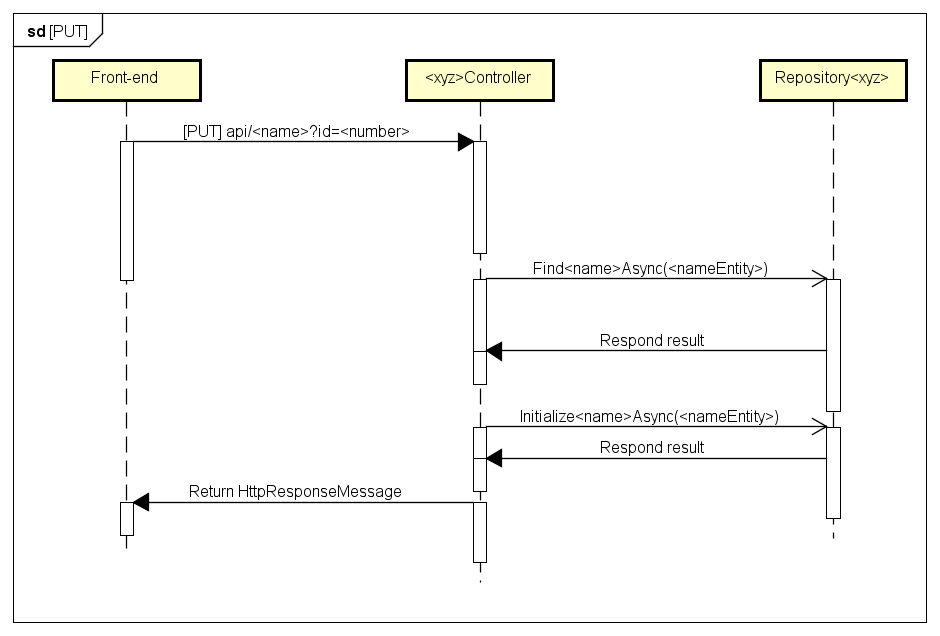
##### 6.1.1.1. Find record



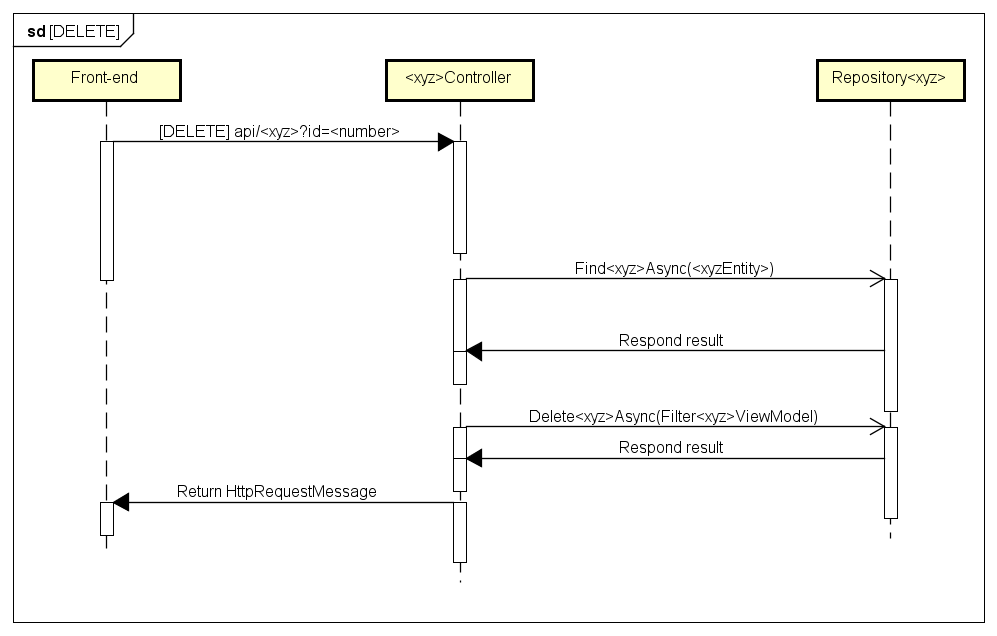
##### 6.1.1.2. Initialize record



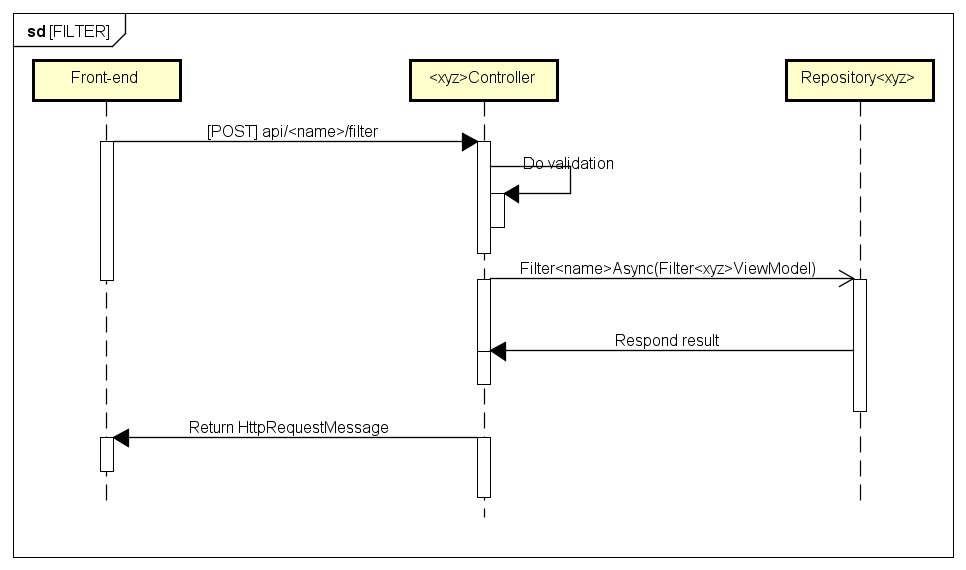
##### 6.1.1.3. Edit record



##### 6.1.1.4. Delete record



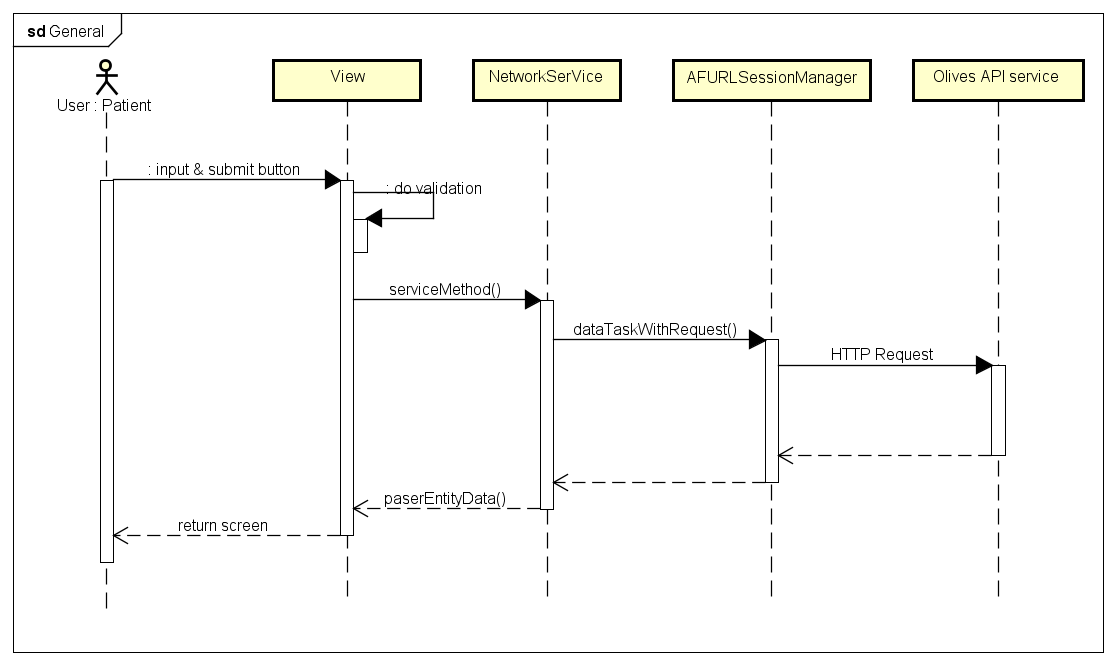
##### 6.1.1.5. Filter record



#### 6.1.2. Olives front-end general sequence diagrams (Patient application)

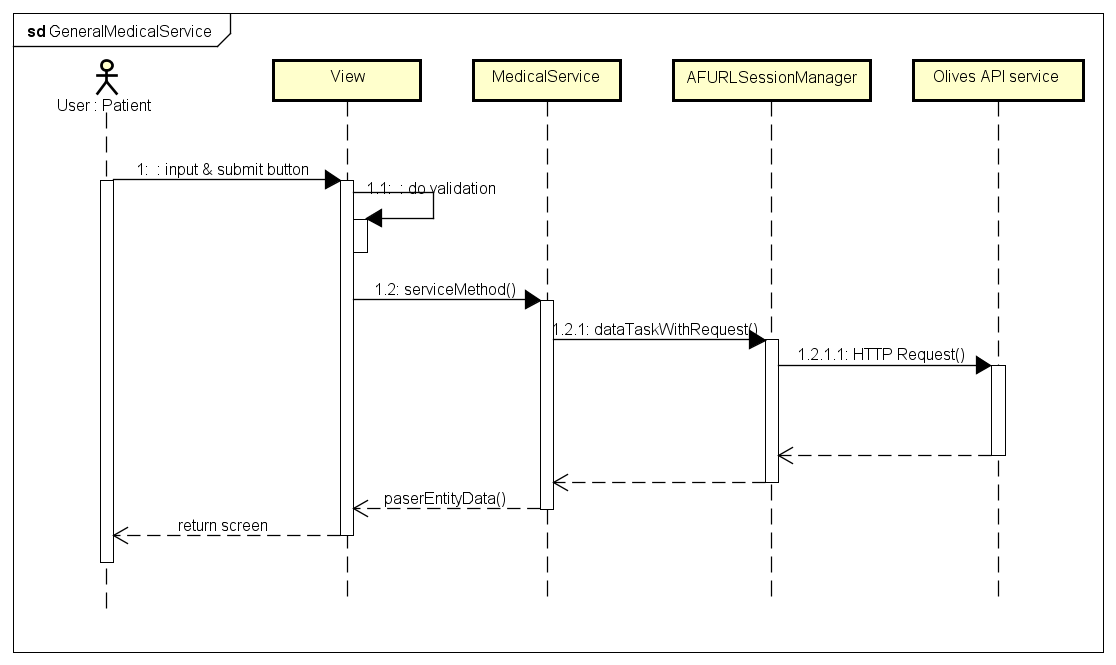
##### 6.1.2.1. Non-medical records data sequence

This type of sequence is for data which doesn’t belong to these types: Medical Record, Experiment Note, Medical Note, Medical Category, Medical Image, Prescription, Prescription Image.



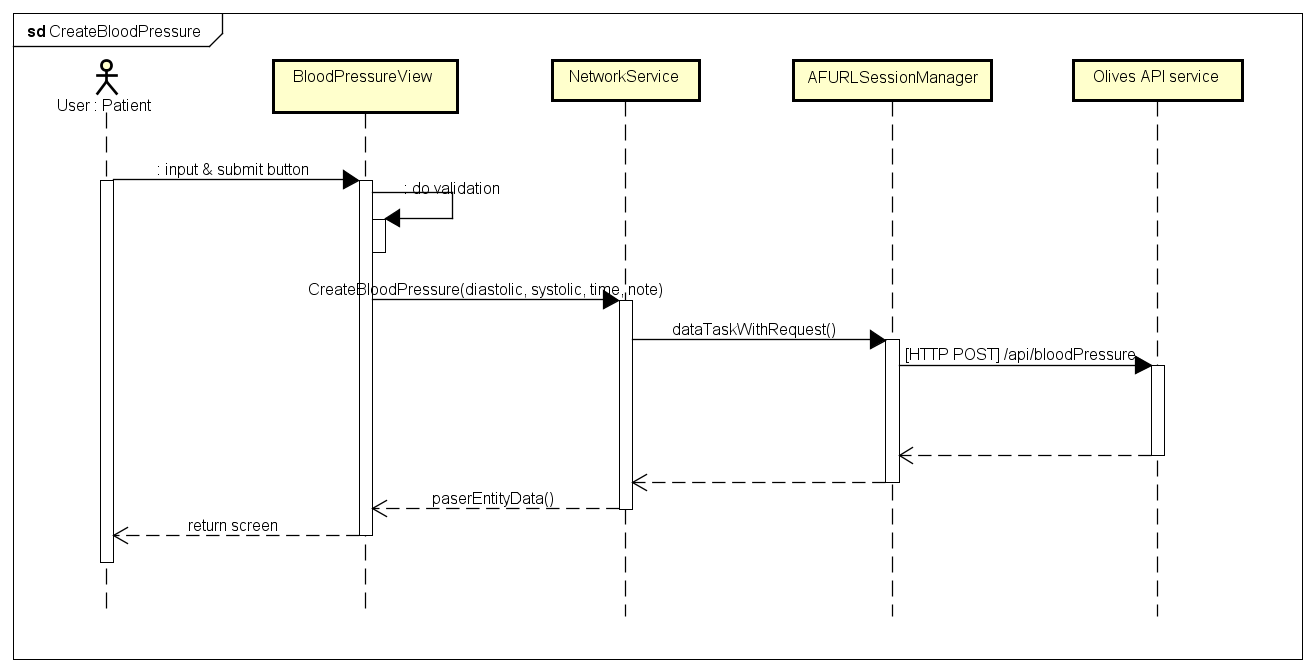
##### 6.1.2.2. Medical records data sequence

This type of sequence is for data which belongs to those excluded data types above.

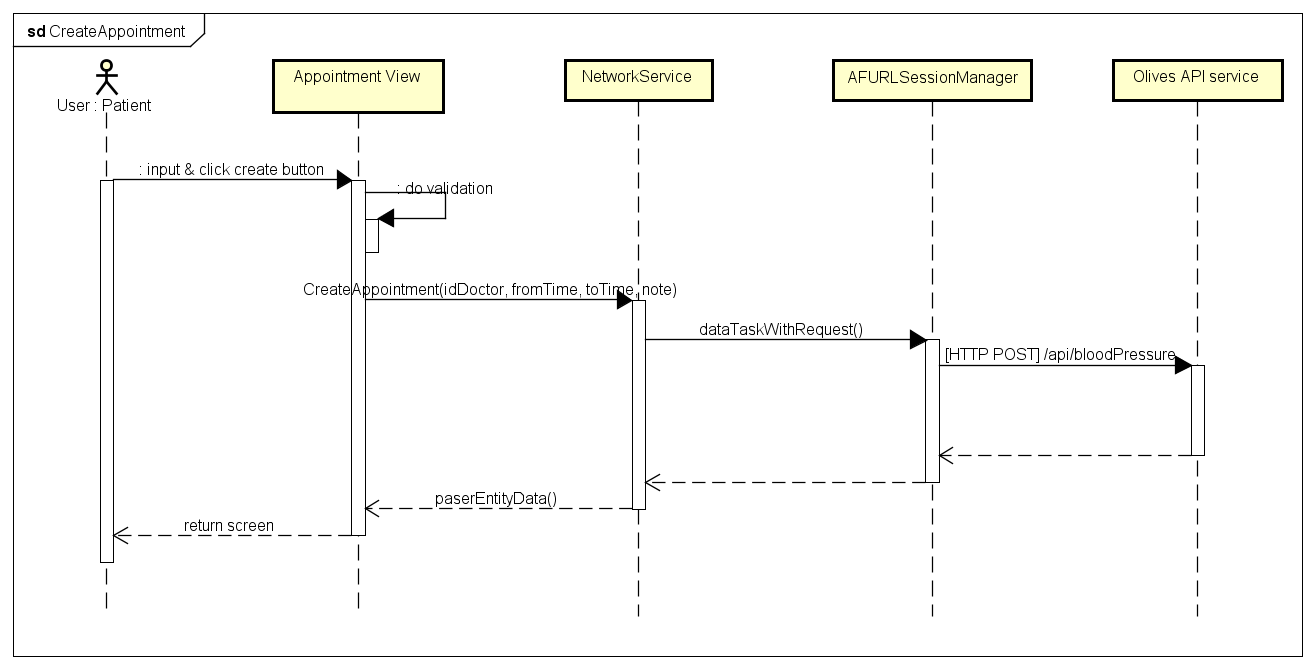


#### 6.1.2.3. Example of ordinary sequence diagrams

##### 6.1.2.3.1. Initialize blood pressure note

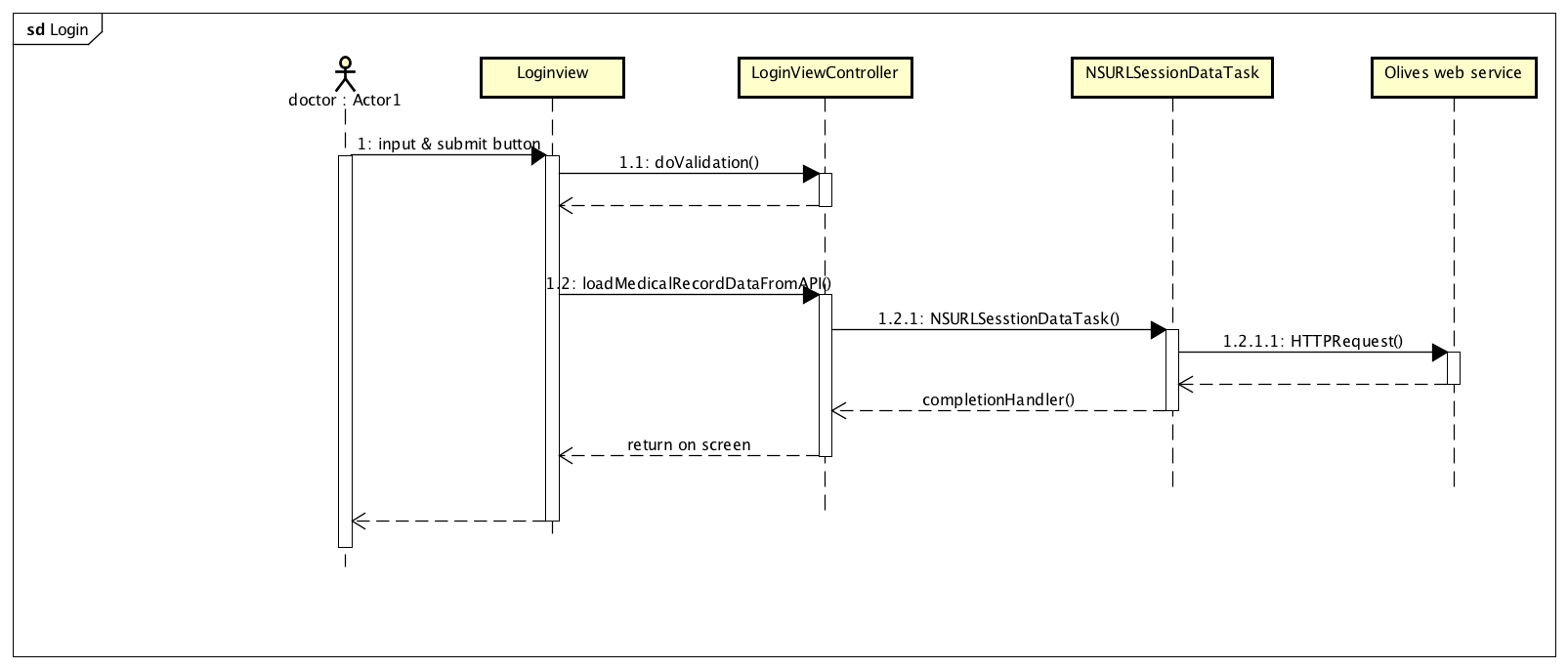


##### 6.1.2.3.2 Initialize appointment

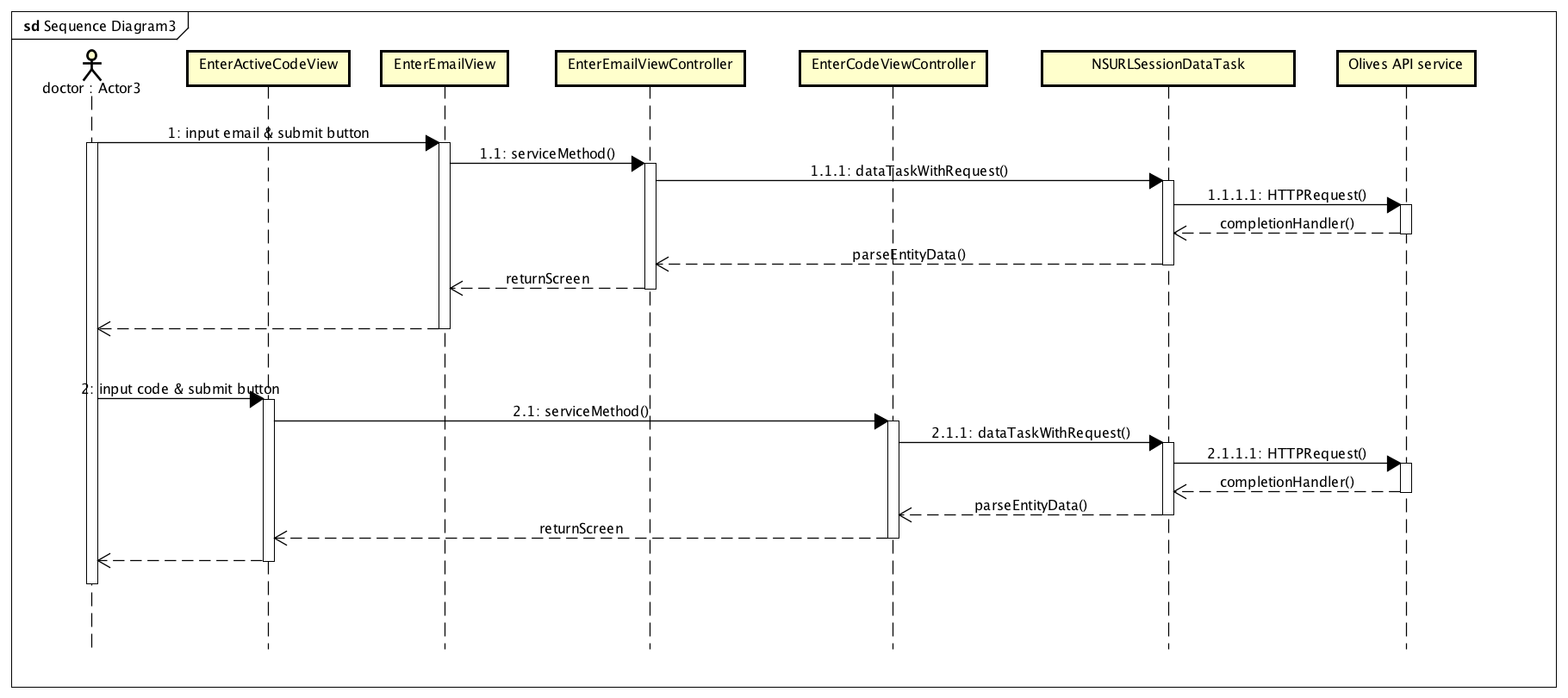


**6.1.3 Olives App doctor sequence diagram**

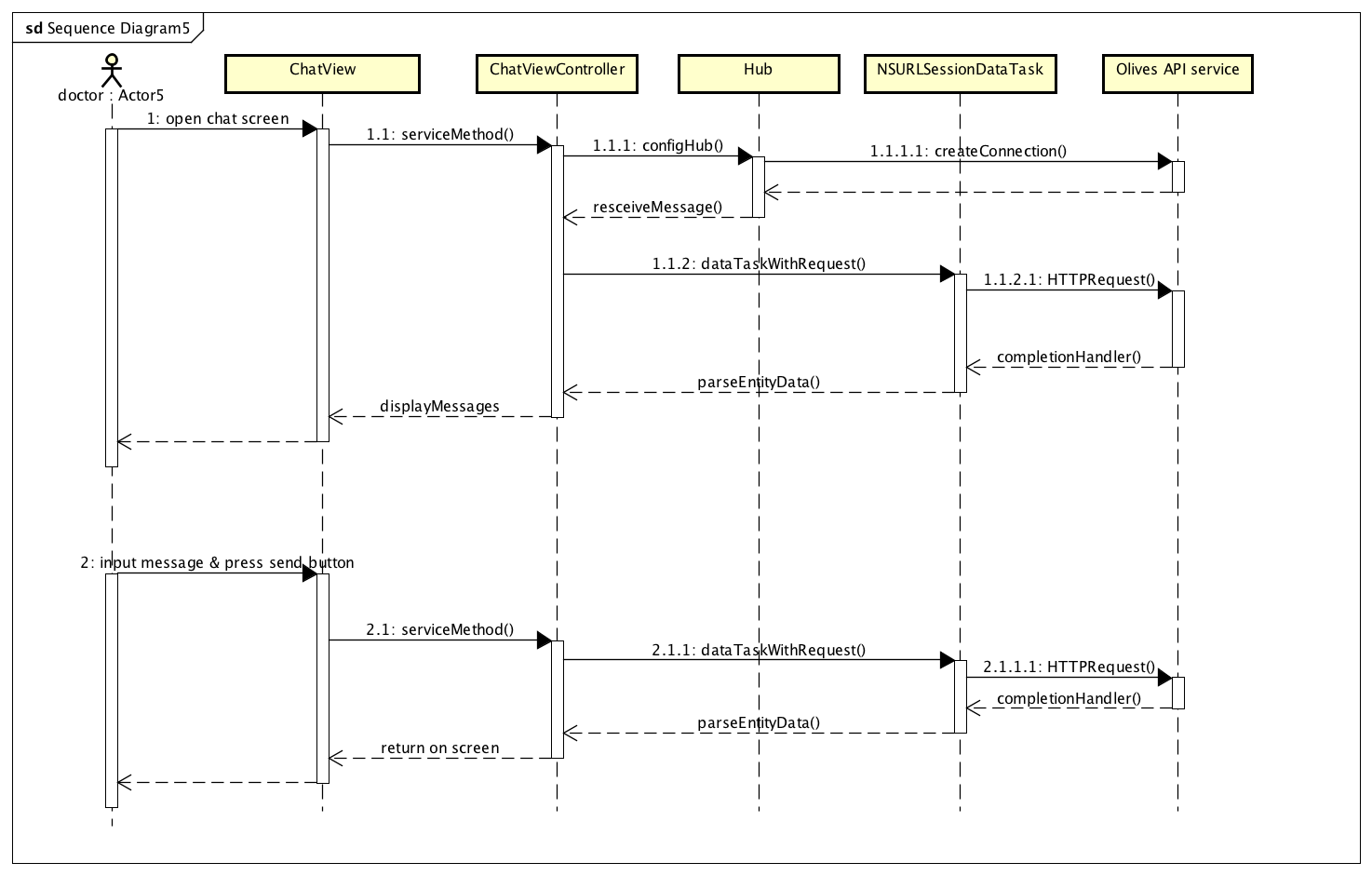
6.1.3.1 Login sequence diagram



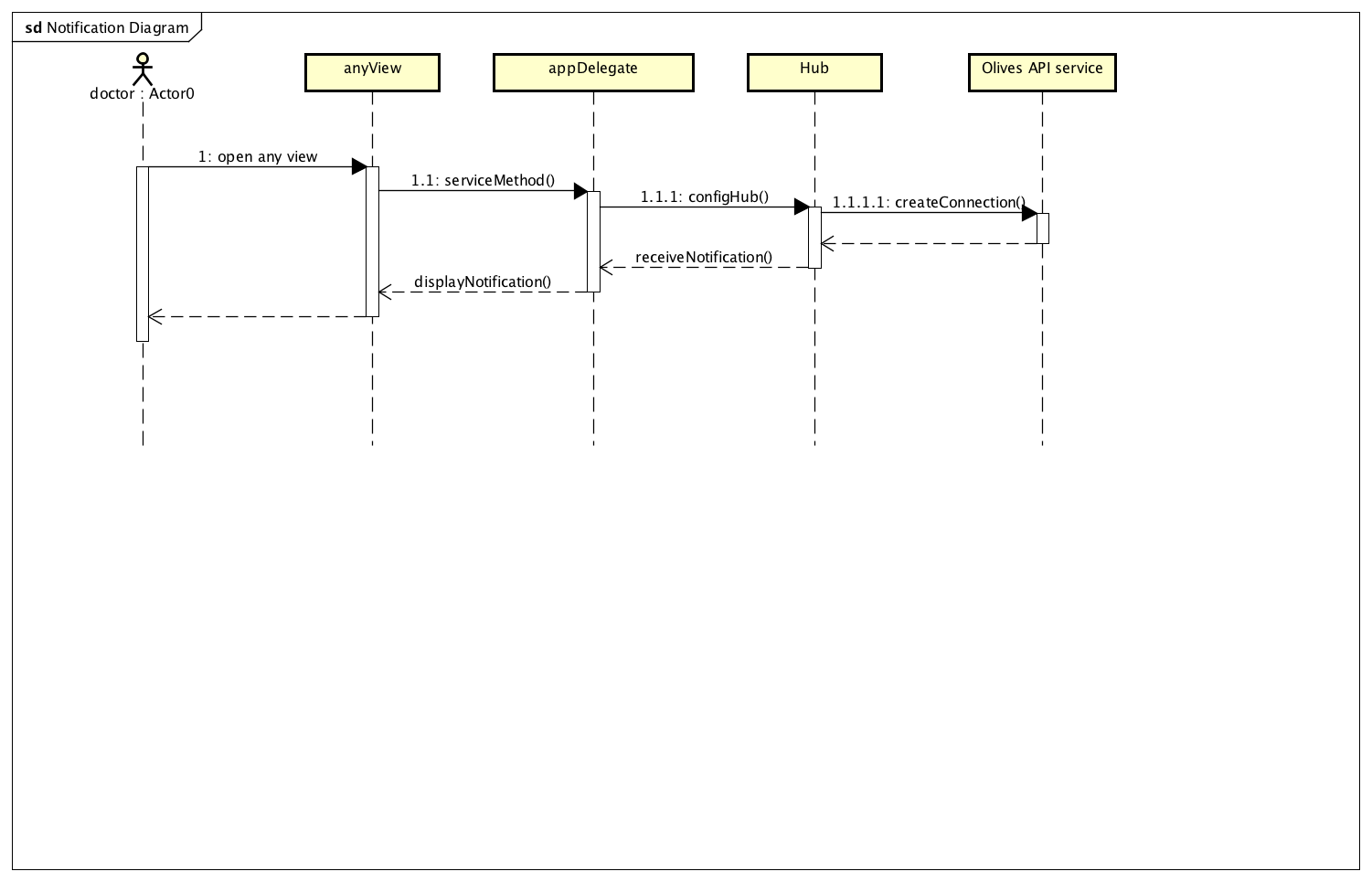
6.1.3.2 Forgot password sequence diagram



6.1.3.3 Chat sequence diagram



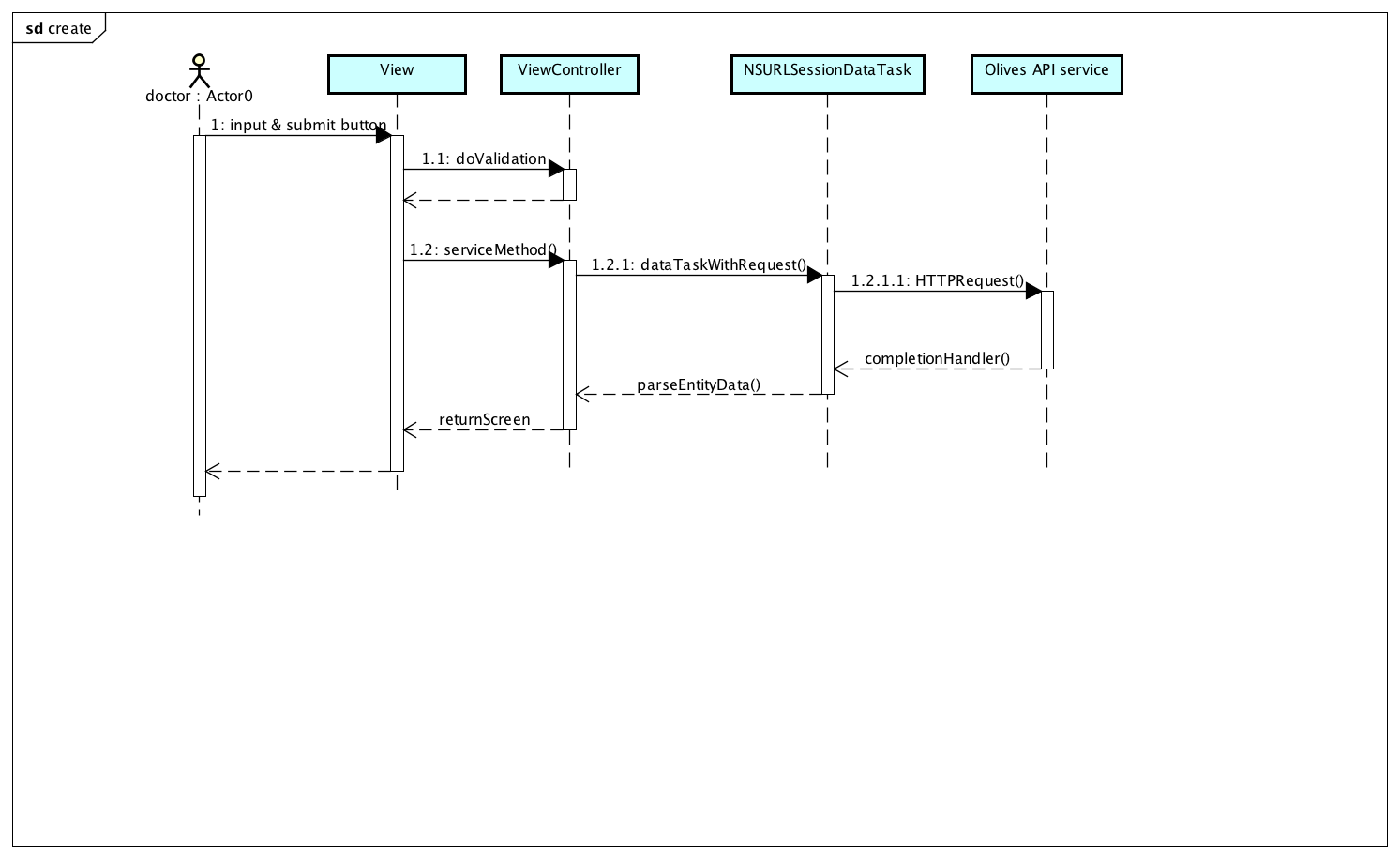
6.1.3.4 Notification sequence diagram



6.1.3.5 Create/add sequence diagram

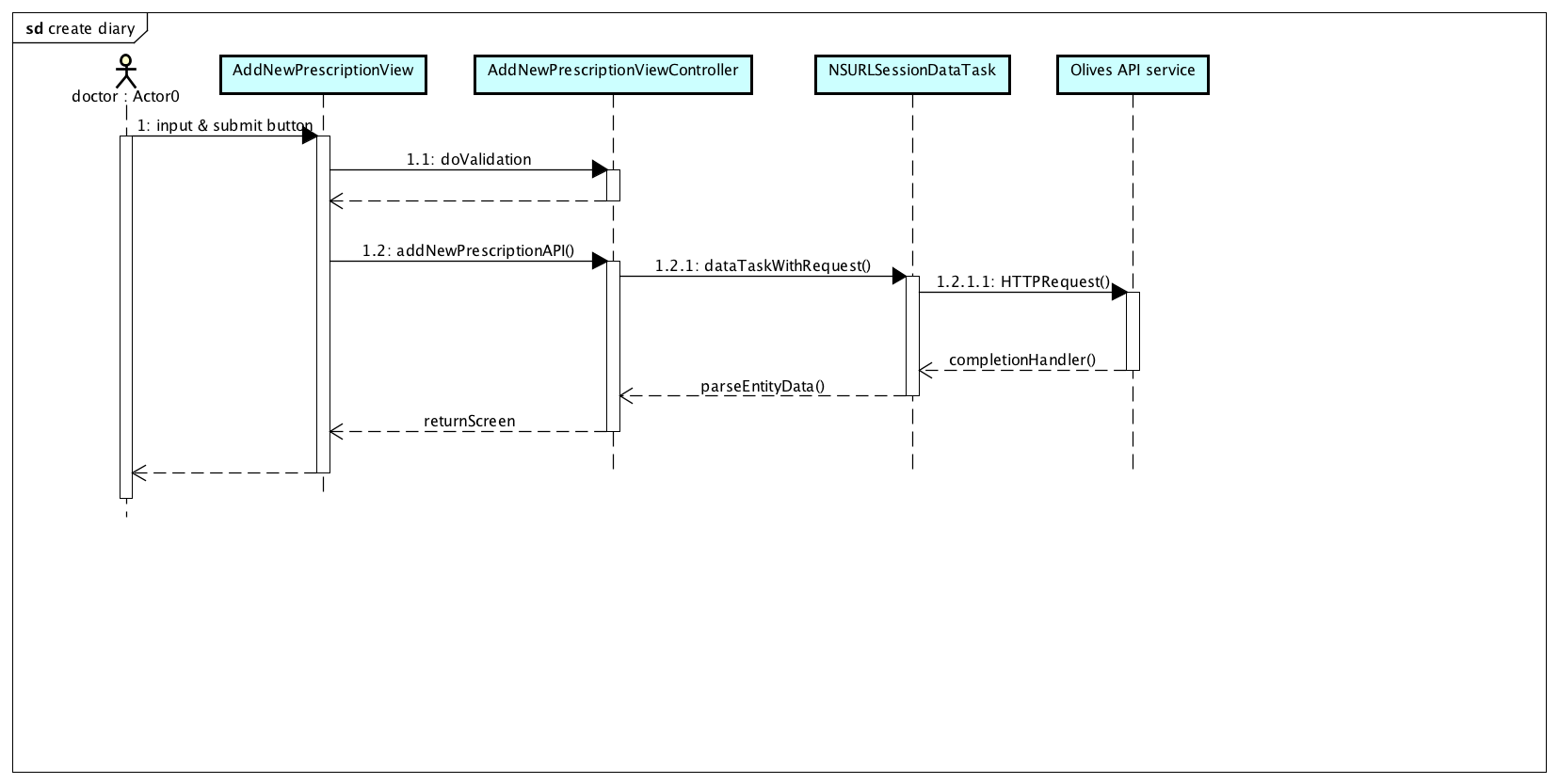
|  |  |
| --- | --- |
| No | Name |
| 1 | Add appointment |
| 2 | Add prescription |
| 3 | Add medical record |
| 4 | Create diary |
|  |  |

Similar use cases with following sequence diagram

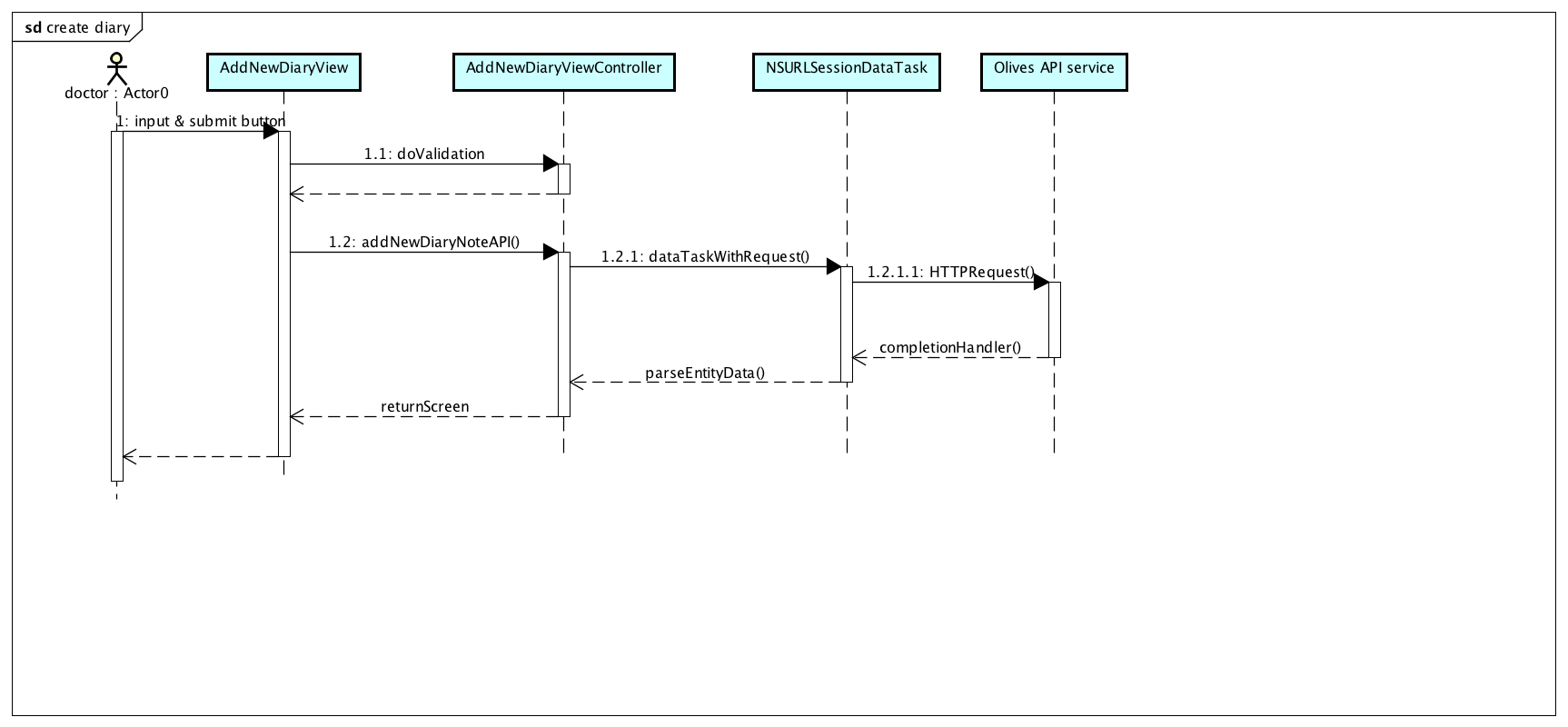


General create sequence diagram

Examples:

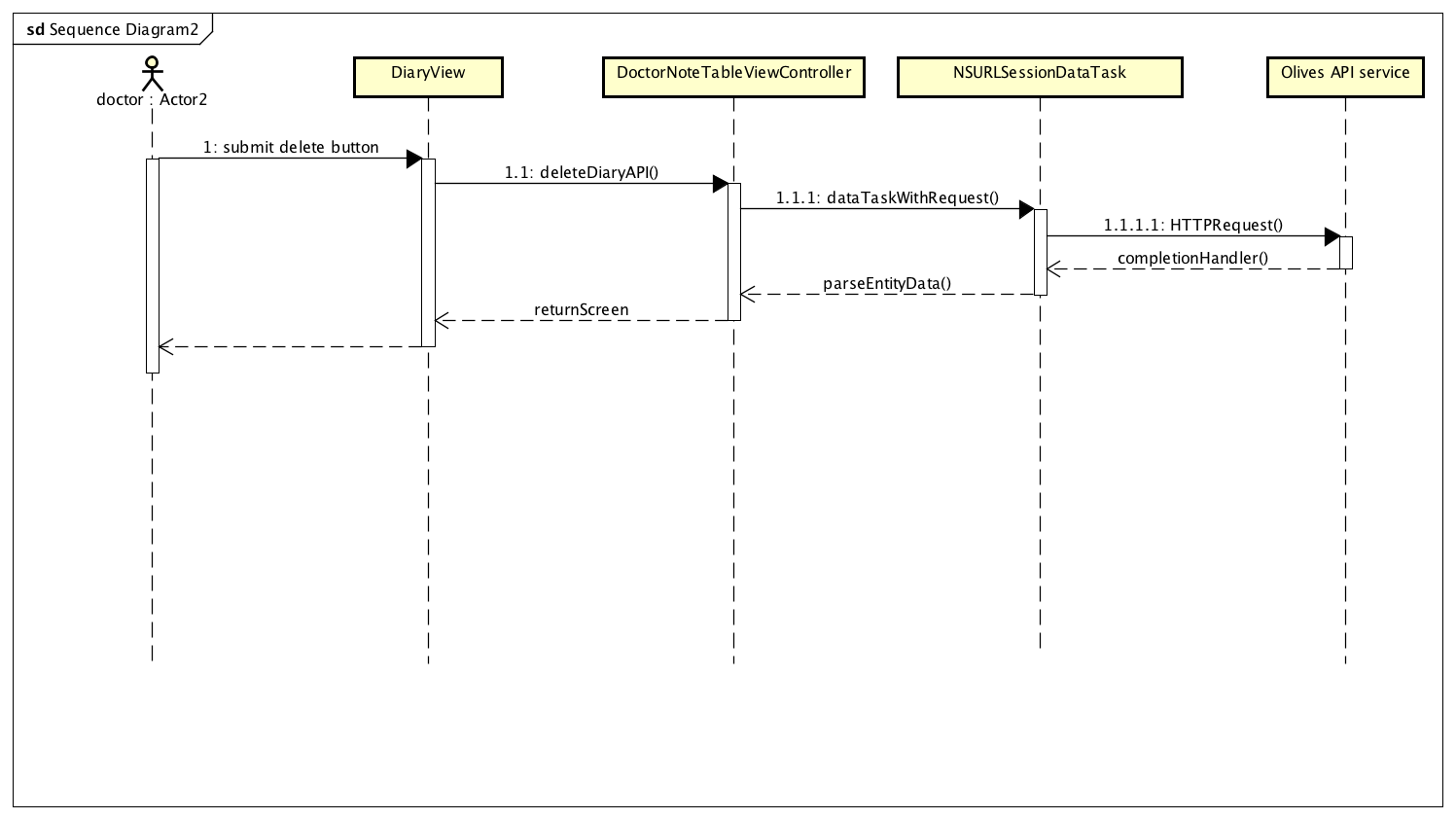


create new prescription diagram



create new diary sequence diagram

6.1.3.6 Delete sequence diagram

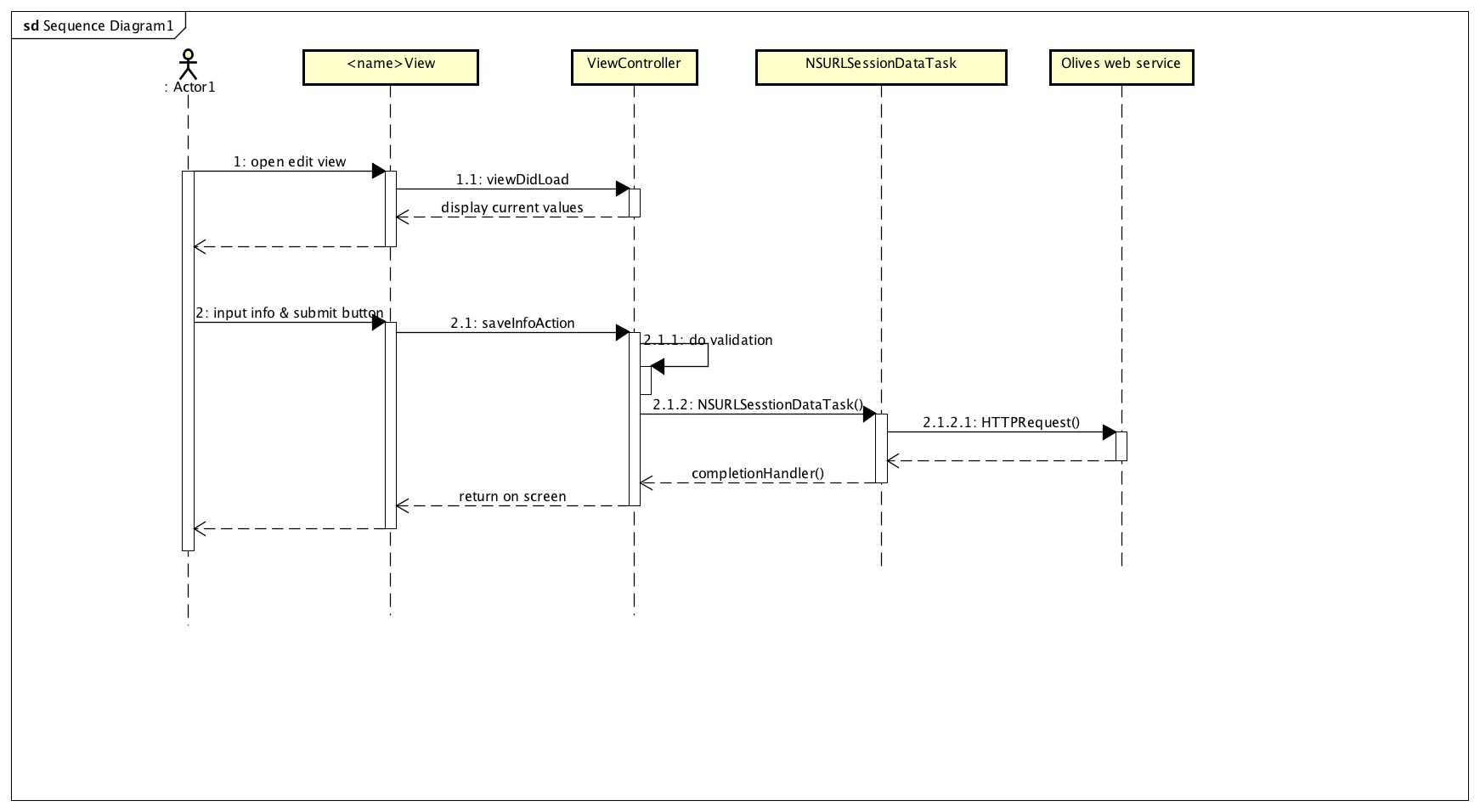


delete diary sequence diagram

6.1.3.7 Edit sequence diagram

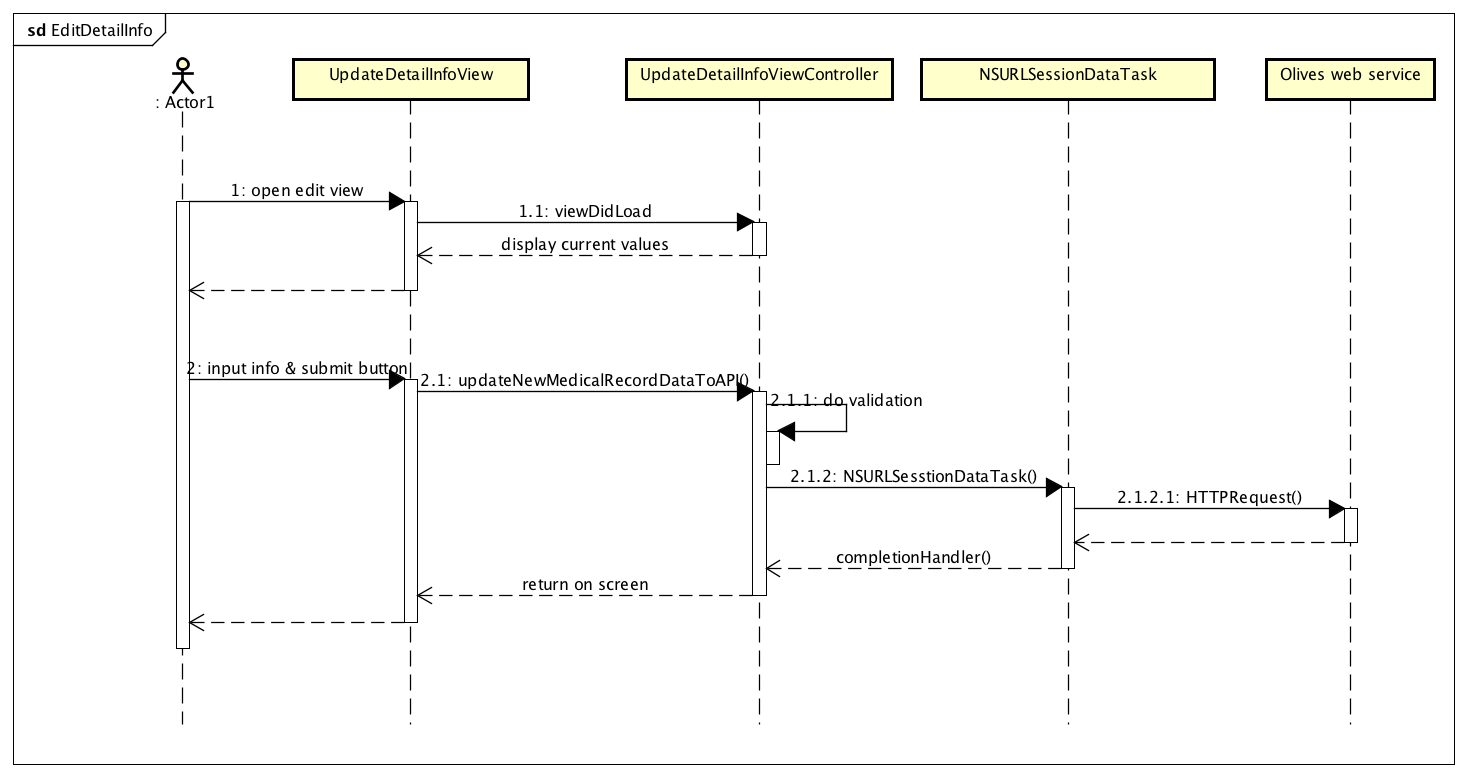
|  |  |
| --- | --- |
| 1 | Edit profiles |
| 2 | Accept patient requests |
| 3 | Cancel patient requests |
| 4 | Edit appointment |
| 5 | Edit prescription |
| 6 | Edit medical record |
| 7 | Edit diary |
|  |  |

Similarly use cases with following sequence diagram

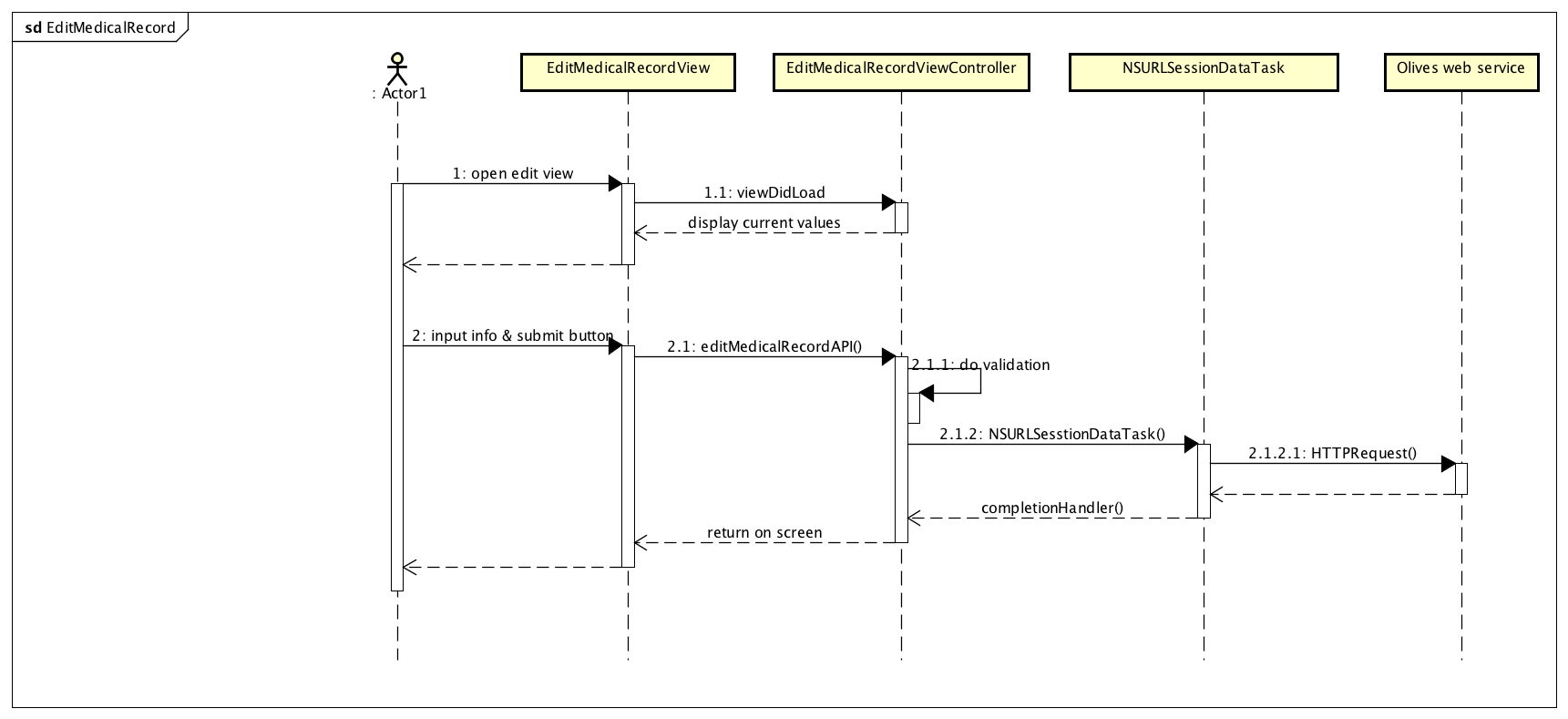


general edit sequence diagram

Examples:



update detail info sequence diagram

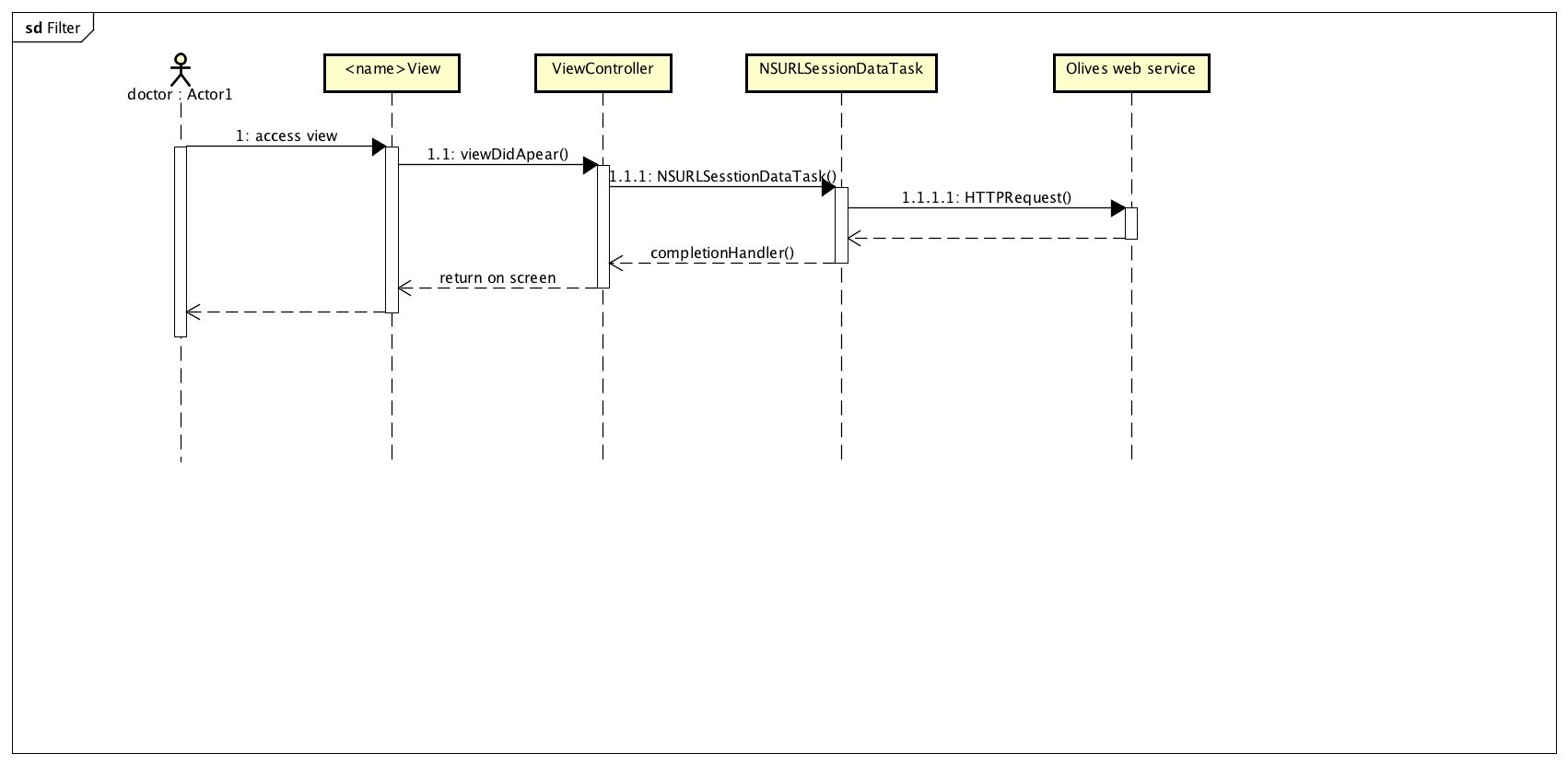


edit medical record sequence diagram

6.1.3.8 Filter sequence diagram

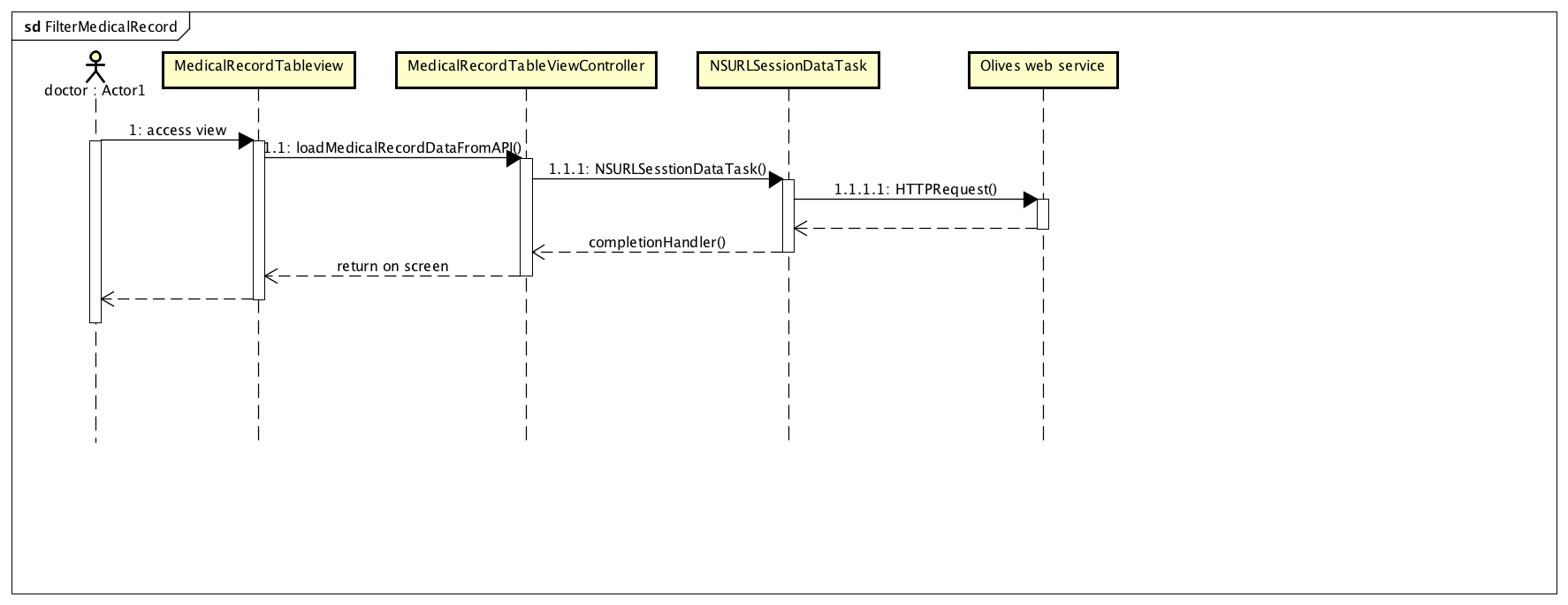
|  |  |
| --- | --- |
| 1 | View current patient |
| 2 | View pending request accept patient |
| 3 | View pending appointment |
| 4 | View current appointment |
| 5 | View patient daily record |
| 6 | View prescription |
| 7 | View medical record |
| 8 | View diary |
| 9 | View patient statistic info |

Similarly use cases with following sequence diagram

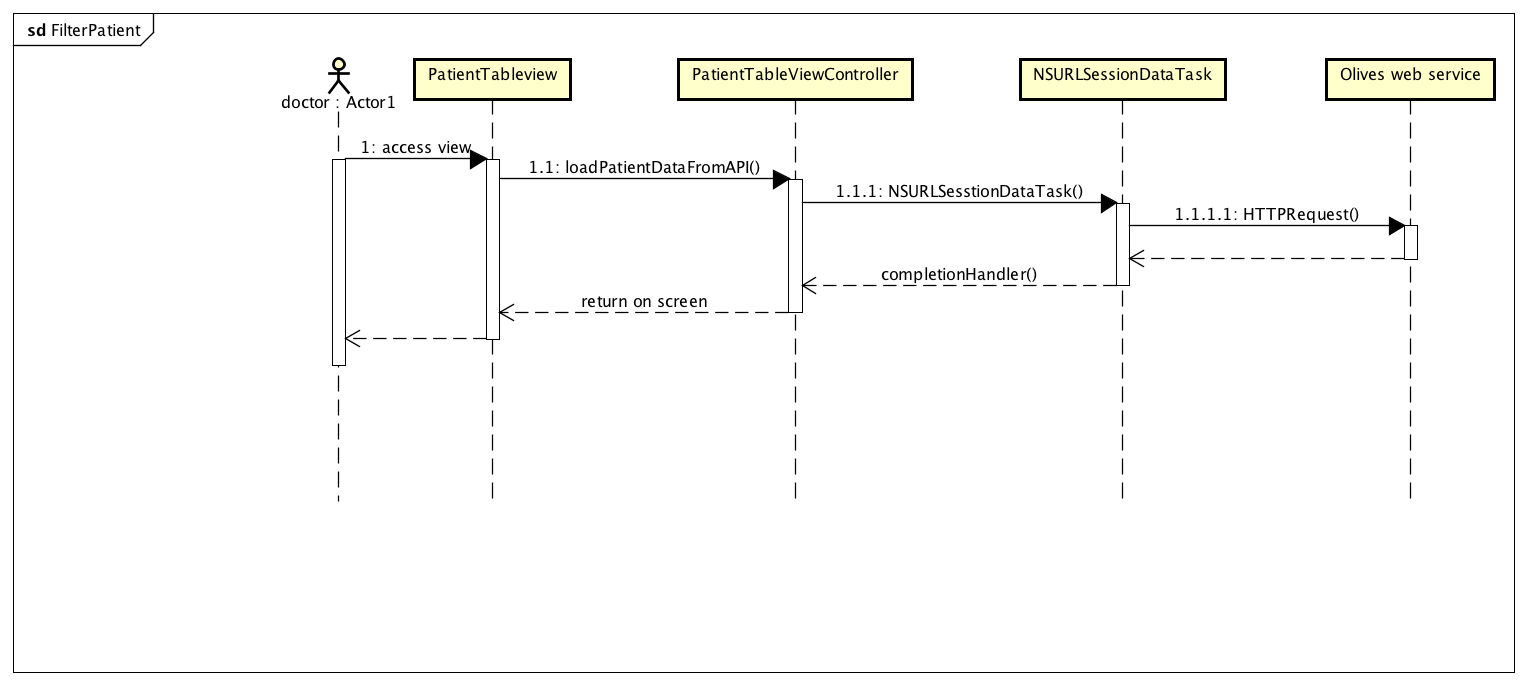


general filter sequence diagram

Examples:



filter medical record sequence diagram



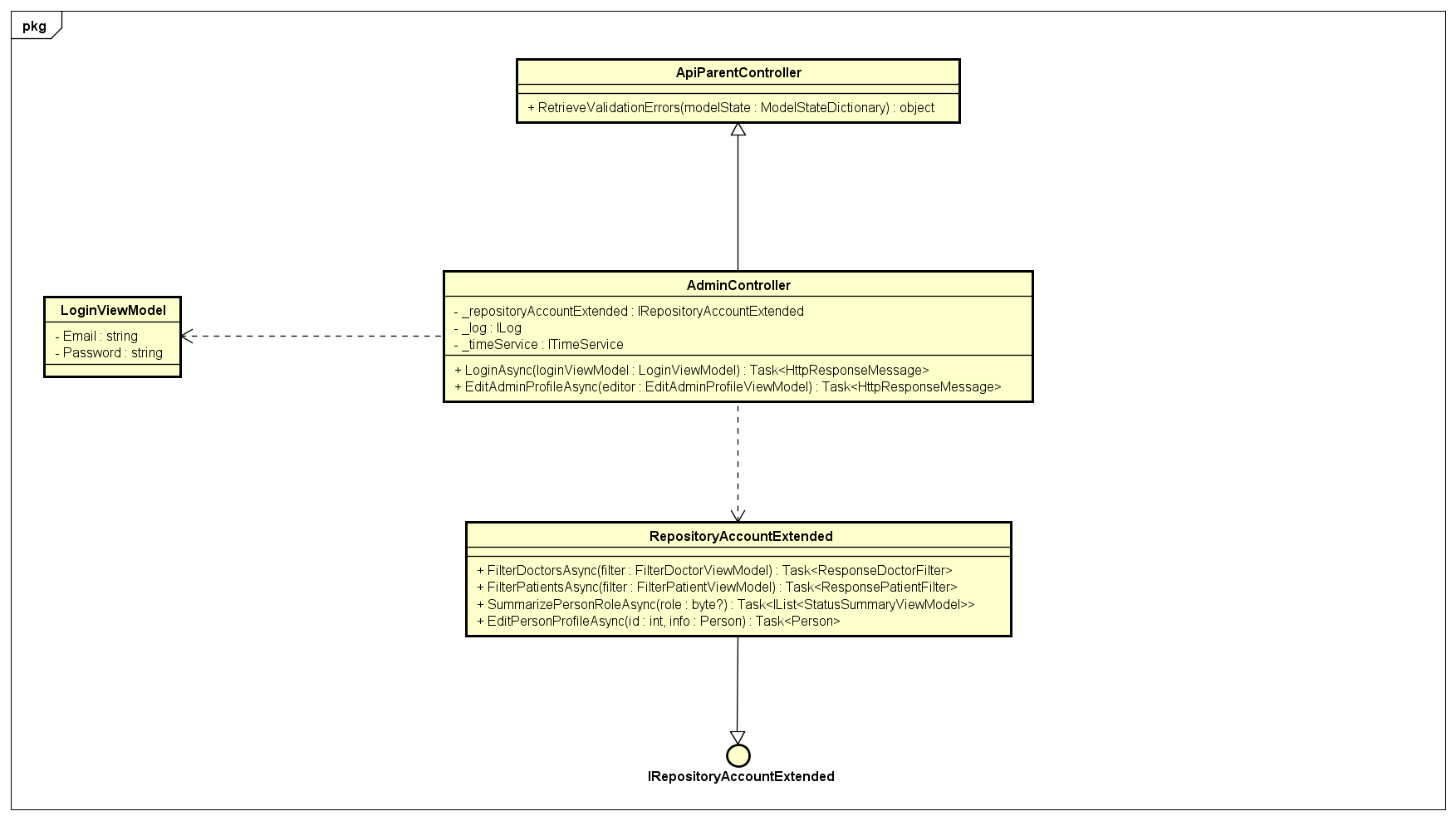
patient filter sequence diagram

### 6.2. Detail design & use case

#### 6.2.1. Olives Administration sub-system

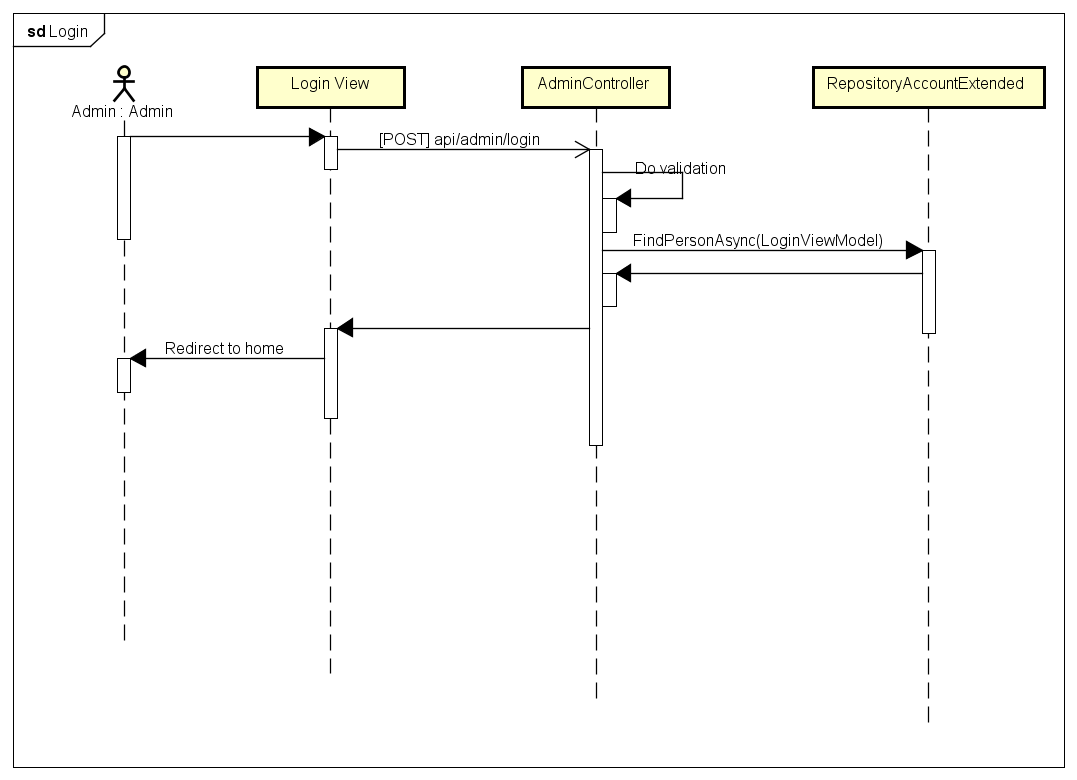
##### 6.2.1.1. Login

###### a. Class diagram



Class diagram of Admin Login.

###### b. Sequence diagram



###### c. Class definition

ApiParentController class

*[Method]RetrieveValidationErrors()*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Take the request parameter model and do the validation and construct a list of validation errors as available which should be seen by clients. | | | |
| Return | **Type** | | | **Description** |
| object | | | Validation error response. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | modelState | ModelStateDictionary | ModelState of MVC/Web API which contains request parameters. |

AdminController class

*Attributes*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_repositoryAccountExtended | var | Instance of interface using to call methods in IRepositoryAccountExtended. |
| 2 | \_log | var | Instance of interface using to call methods in ILog. |
| 3 | \_timeService | var | Instance of interface using to call methods in ITimeService. |

*[Method]LoginAsync()*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Take the parameters come from clients, do validation and log user into system as the information is valid. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Response message which is thrown back to front-end to handle the processed result. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | loginViewModel | LoginViewModel | A view model contains Email and Password which are used for logging into system. |

*[Method]EditAdminProfileAsync()*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Take the parameters come from clients, do validation and edit the profile of admin as the information is valid. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Response message which is thrown back to front-end to handle the processed result. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | editor | EditAdminProfileViewModel | A view model contains the information which is used for changing information in profile. |

RepositoryAccountExtended class

*Attributes*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_dataContext | var | Instance of interface using to call methods in IOliveDataContext. |

*[Method]FilterDoctorAsync()*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Take the request parameters, filter and return a list of condition matched doctors. | | | |
| Return | **Type** | | | **Description** |
| ResponseDoctorFilter | | | Contains a filtered results list and a total record counter. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | FilterDoctorViewModel | Model contains the conditions which are used for filtering. |

*[Method]FilterPatientsAsync()*

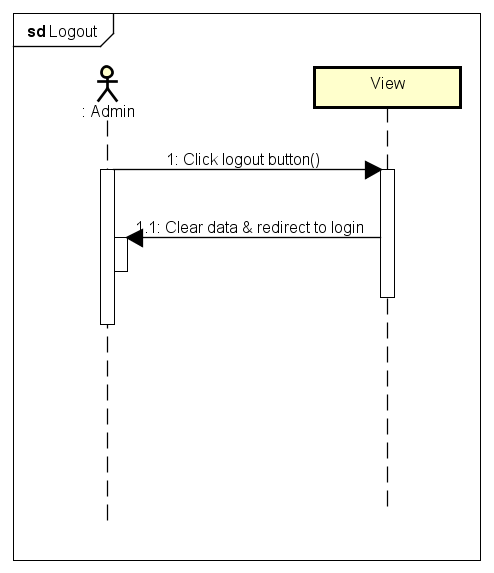
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Take the request parameters, filter and return a list of condition matched patients. | | | |
| Return | **Type** | | | **Description** |
| ResponsePatientFilter | | | Contains a filtered results list and a total record counter. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | FilterPatientViewModel | Model contains the conditions which are used for filtering. |

*[Method]SummarizePersonRoleAsync()*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Take the request parameters, filter and return a list of condition matched people with role statistic. | | | |
| Return | **Type** | | | **Description** |
| IList<StatusSummaryViewModel> | | | Contains a filtered results list and a total record counter. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | role | Byte? | Role which is used for determining the filter. |

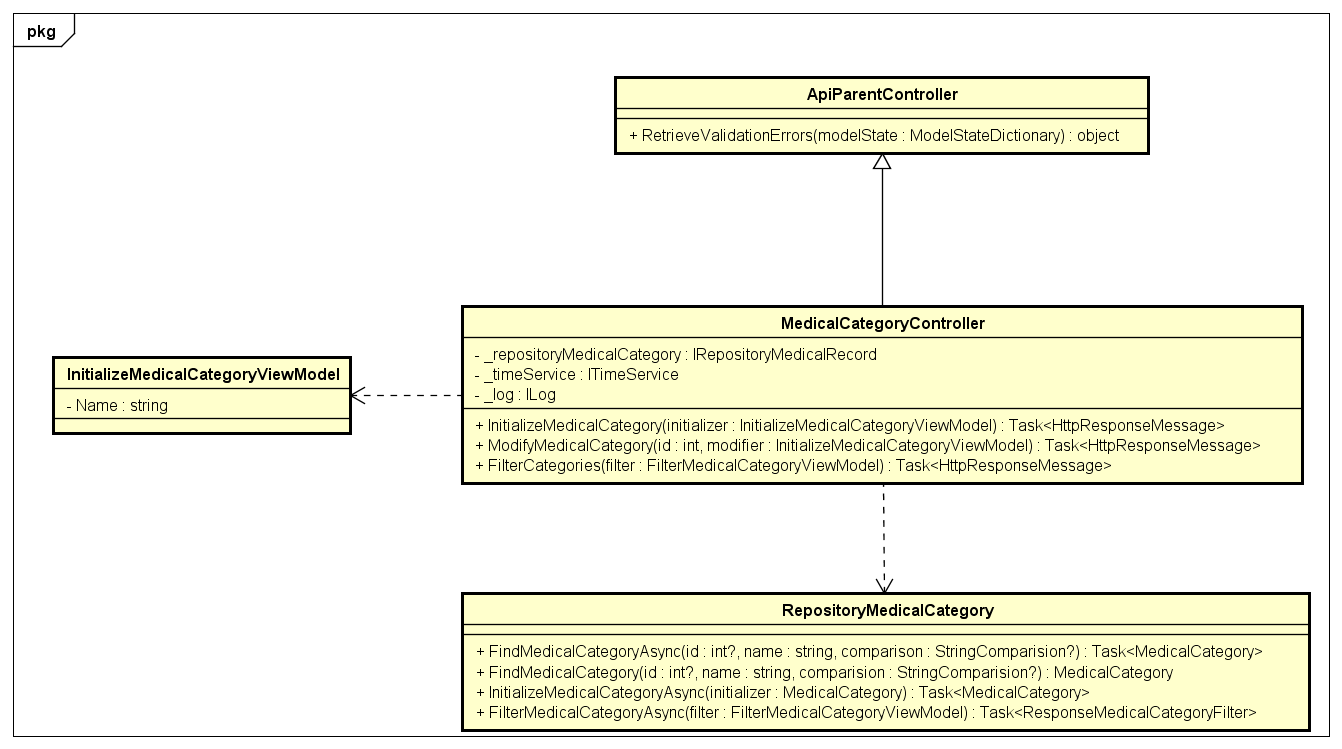
##### 6.2.1.2. Logout

1. Sequence diagram



##### 6.2.1.3. Create medical category

1. Class diagram



1. Class definition

MedicalCategoryController class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_repositoryMedicalCategory | var | Instance of interface using to call methods in IRepositoryMedicalCategory. |
| 2 | \_timeService | var | Instance of interface using to call methods in ITimeService. |
| 3 | \_log | var | Instance of interface using to call methods in IRepositoryMedicalCategory. |

Method: InitializeMedicalCategory()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Take the request parameters, validate and initialize a medical category as the information is valid. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Contains the information which will be thrown back to the clients. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | initializer | InitializeMedicalCategoryViewModel | Model contains the information of medical record which will be used for initializing a medical category. |

Method: ModifyMedicalCategory()

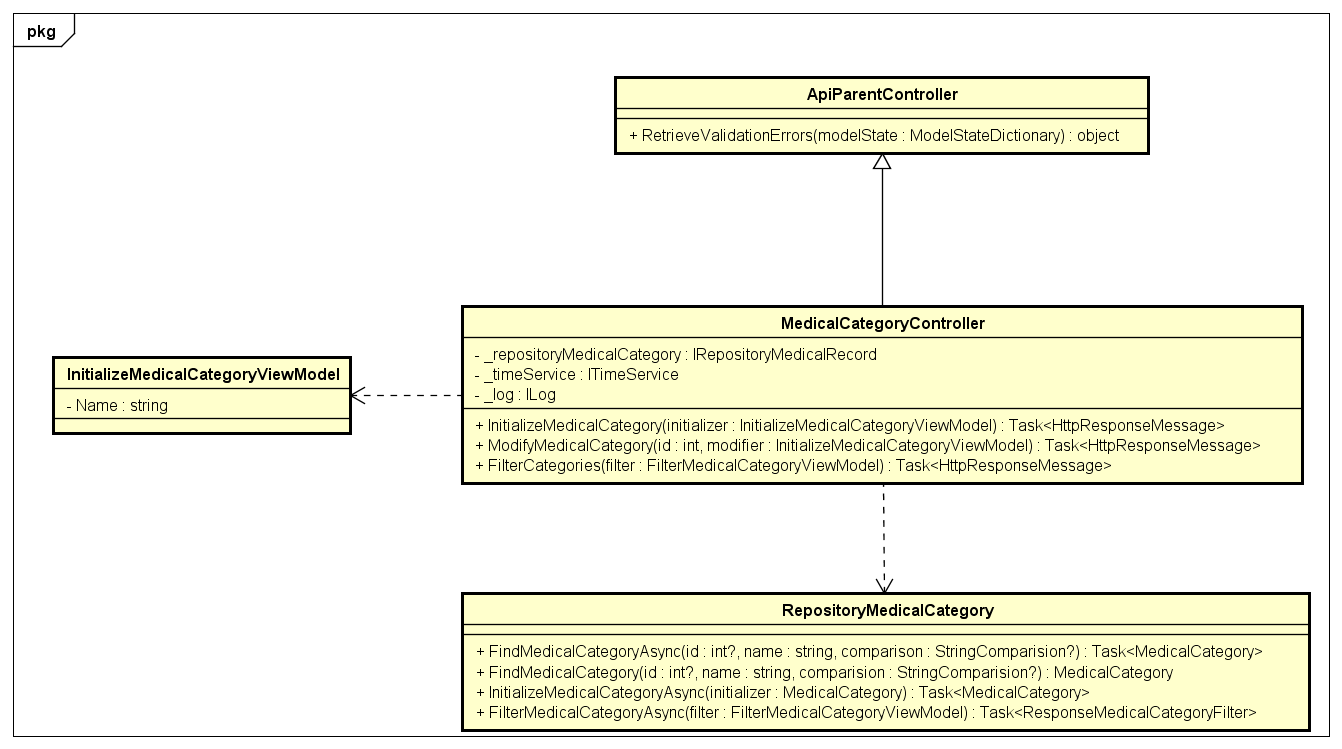
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Take the request parameters, validate and modify a medical category as the information is valid | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Contains the information which will be thrown back to the clients. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of medical category which will be targeted to be edited. |
| 2 | modifier | InitializeMedicalCategoryViewModel | Model contains the information of medical record which will be used for modifying a medical category. |

Method: FilterCategories()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Take the request parameters, validate and filter and return the list of condition matched medical categories. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Contains the information which will be thrown back to the clients. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 2 | filter | FilterMedicalCategoryViewModel | Model contains the information which are used for medical categories filtering. |

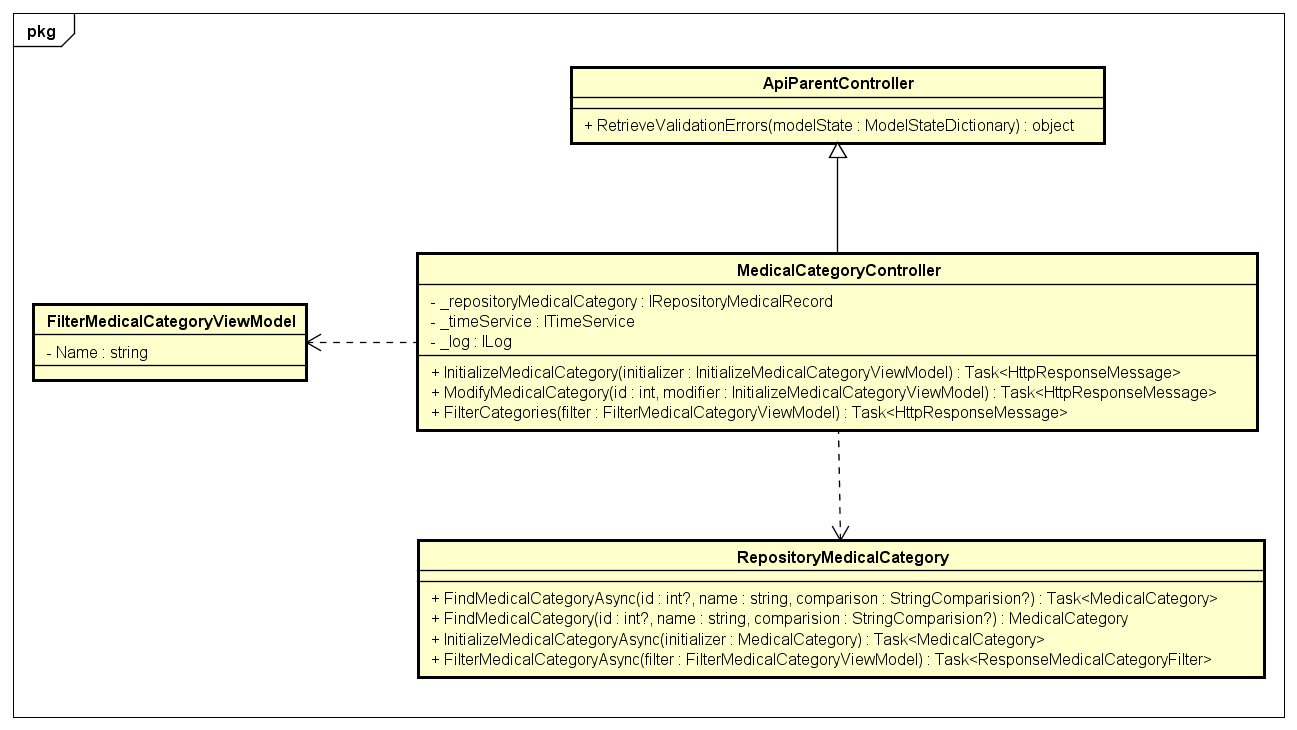
##### 6.2.1.4. Edit medical category

1. Class diagram



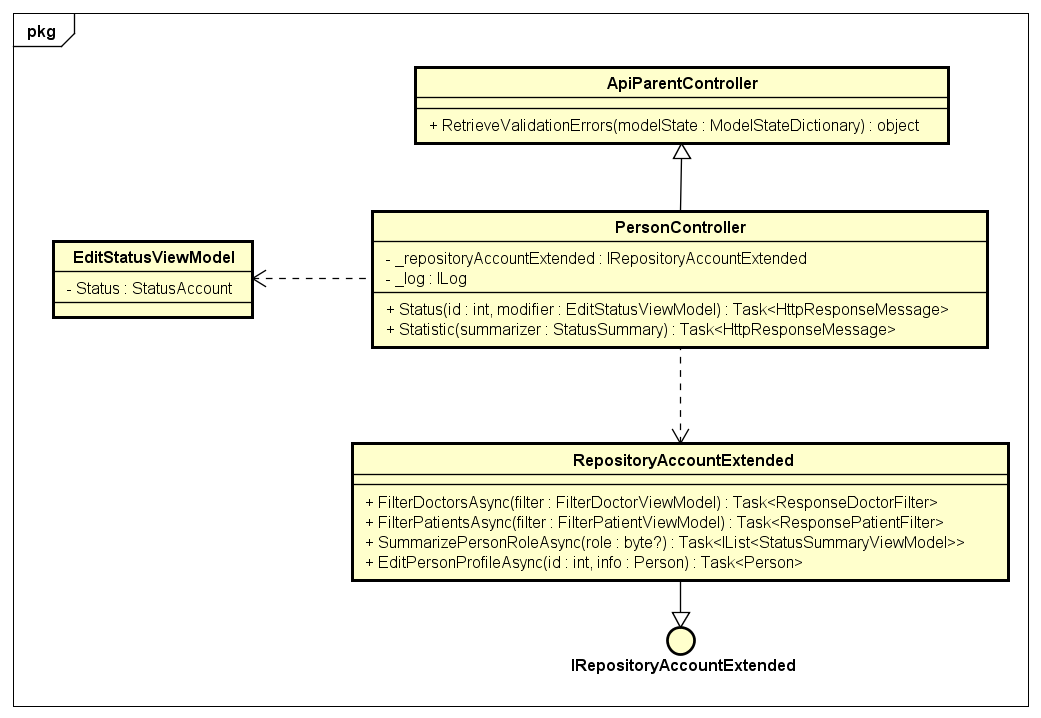
##### 6.2.1.5. Filter medical category

###### a. Class diagram

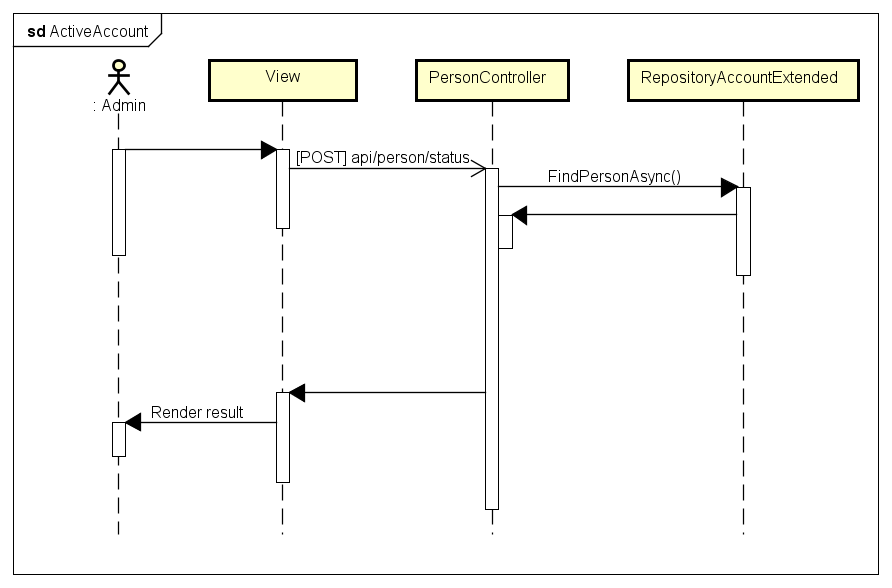


##### 6.2.1.6. Activate account

1. Class diagram

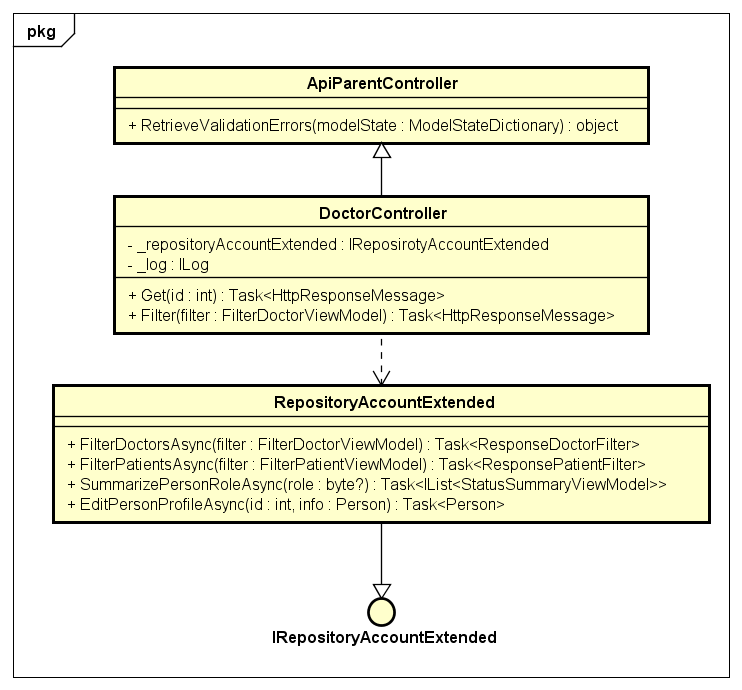


1. Sequence diagram



##### 6.2.1.7. Find doctor

1. Class diagram



1. Class definition

DoctorController class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_repositoryMedicalCategory | var | Instance of interface using to call methods in IRepositoryMedicalCategory. |
| 2 | \_timeService | var | Instance of interface using to call methods in ITimeService. |
| 3 | \_log | var | Instance of interface using to call methods in IRepositoryMedicalCategory. |

Method: Get()

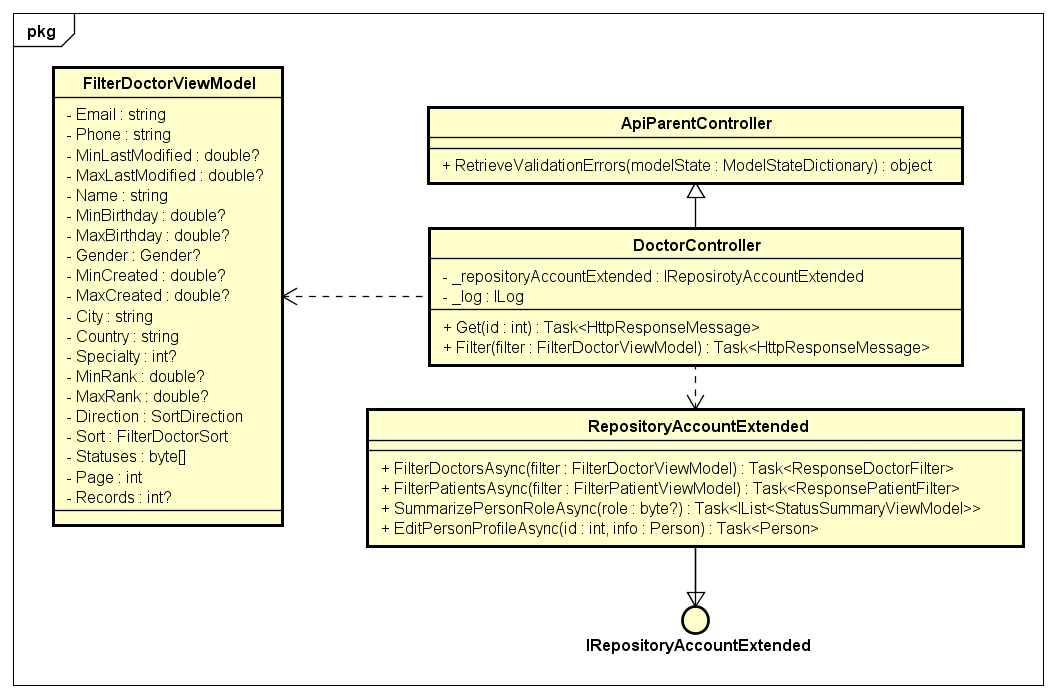
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Take the id sent from client and find the doctor by using id. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Contains the information which will be thrown back to the clients. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of doctor. |

Method: Filter()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Take the id sent from client and find the doctor by using id. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Contains the information which will be thrown back to the clients. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | FilterDoctorViewModel | Model contains the information for filter. |

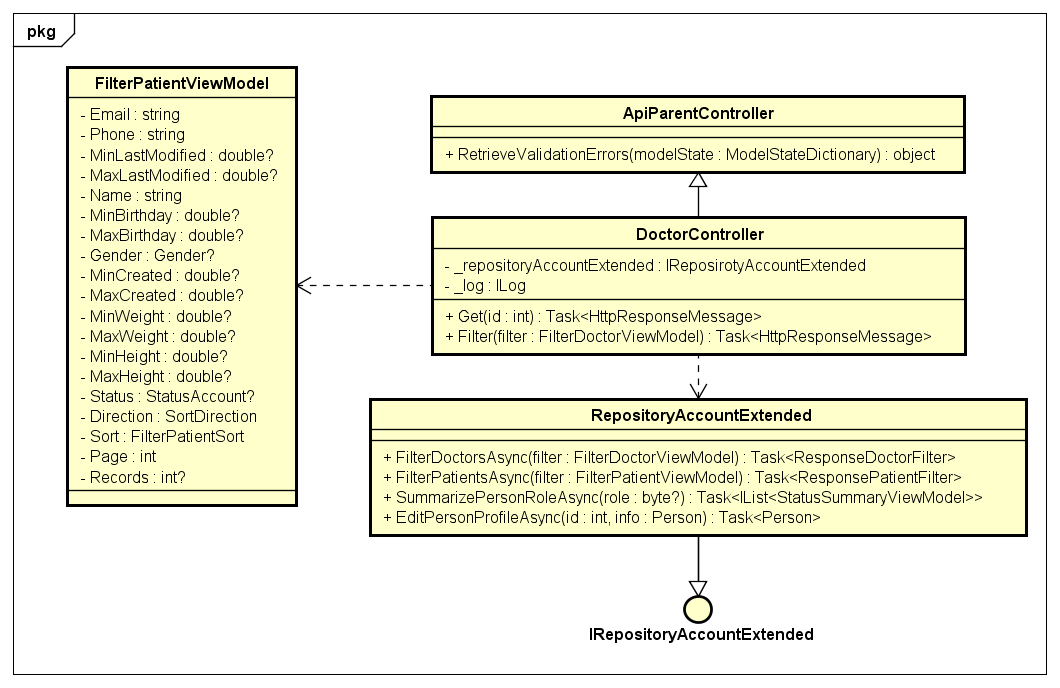
##### 6.2.1.8. Filter doctors

1. Class diagram



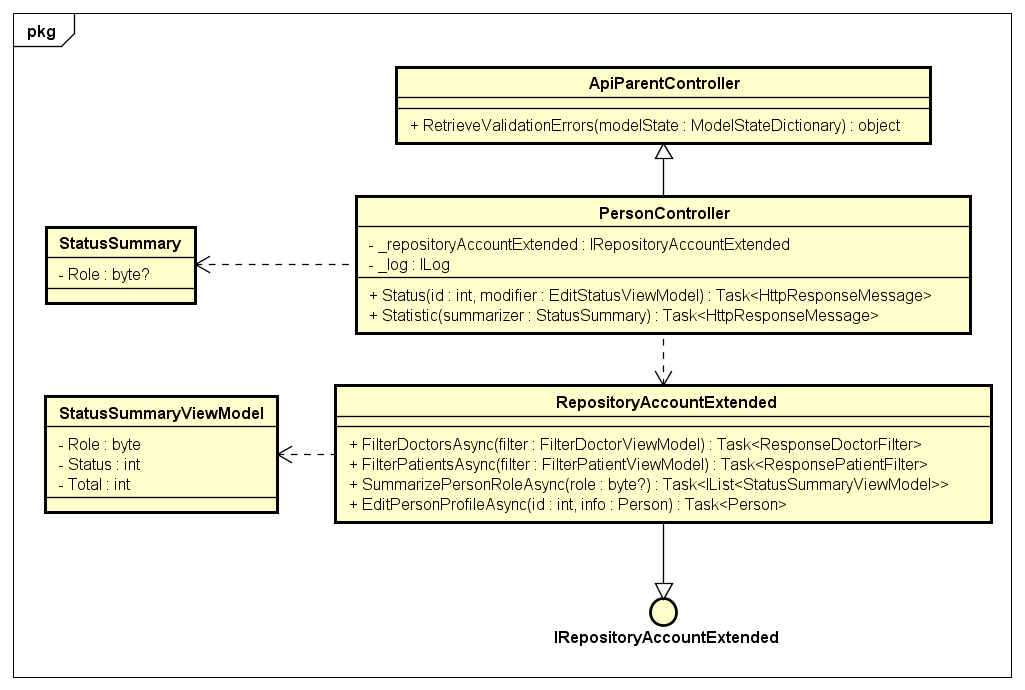
##### 6.2.1.9. Find patients

1. Class diagram

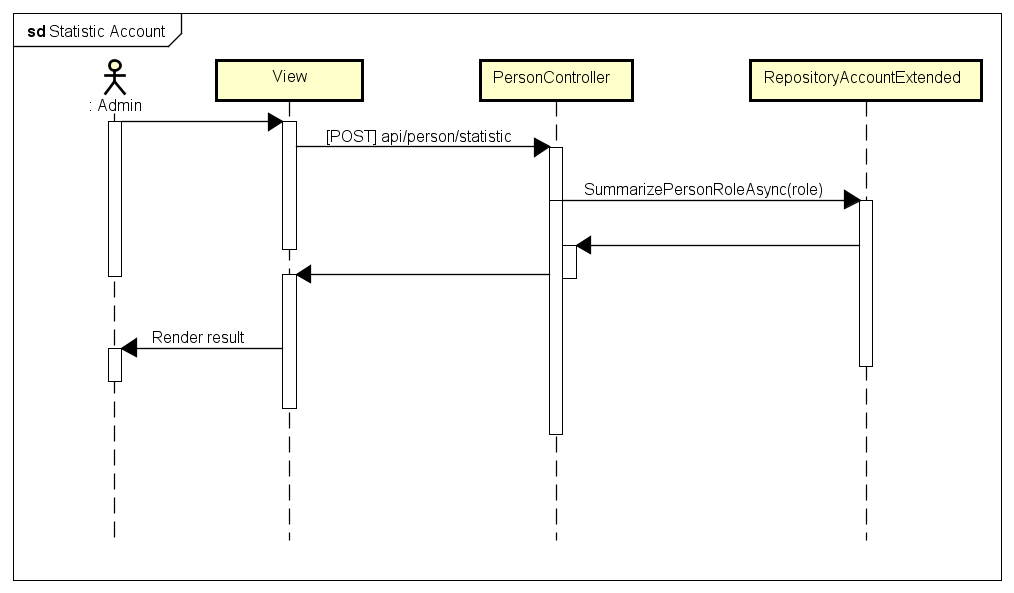


##### 6.2.1.10. Statistic accounts by role

1. Class diagram



1. Sequence diagram



1. Class diagram

PersonController class

Attribute

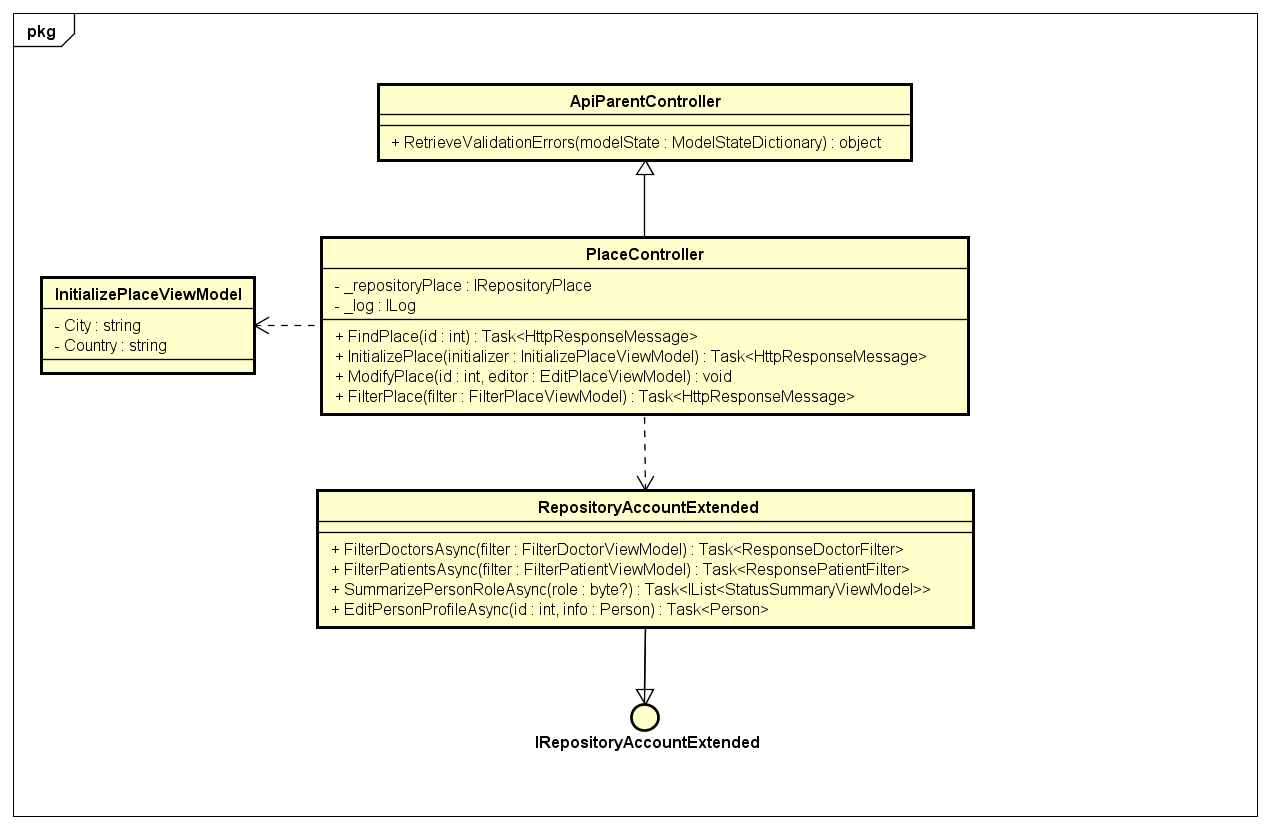
|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_repositoryMedicalCategory | var | Instance of interface using to call methods in IRepositoryMedicalCategory. |
| 2 | \_timeService | var | Instance of interface using to call methods in ITimeService. |
| 3 | \_log | var | Instance of interface using to call methods in IRepositoryMedicalCategory. |

Method: Status()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Take the id sent from client and find the person by using id. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Contains the information which will be thrown back to the clients. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of doctor. |
| 2 | modifier | EditStatusViewModel | Model contains information which is used for modifying person status. |

##### 6.2.1.11. Initialize place

1. Class diagram



1. Class definition

PlaceController class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_repositoryPlace | IRepositoryPlace | Instance of interface using to call methods in IRepositoryPlace. |
| 2 | \_log | ILog | Instance of interface using to call methods in ILog. |

Method: FindPlace()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Take the id sent from client and find the place by using id. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Contains the information which will be thrown back to the clients. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of place. |

Method: InitializePlace()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Take the information requested from client and create a place in database as the informations is available. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Contains the information which will be thrown back to the clients. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | initializer | InitializePlaceViewModel | Model which contains place information for being initialized. |

Method: ModifyPlace()

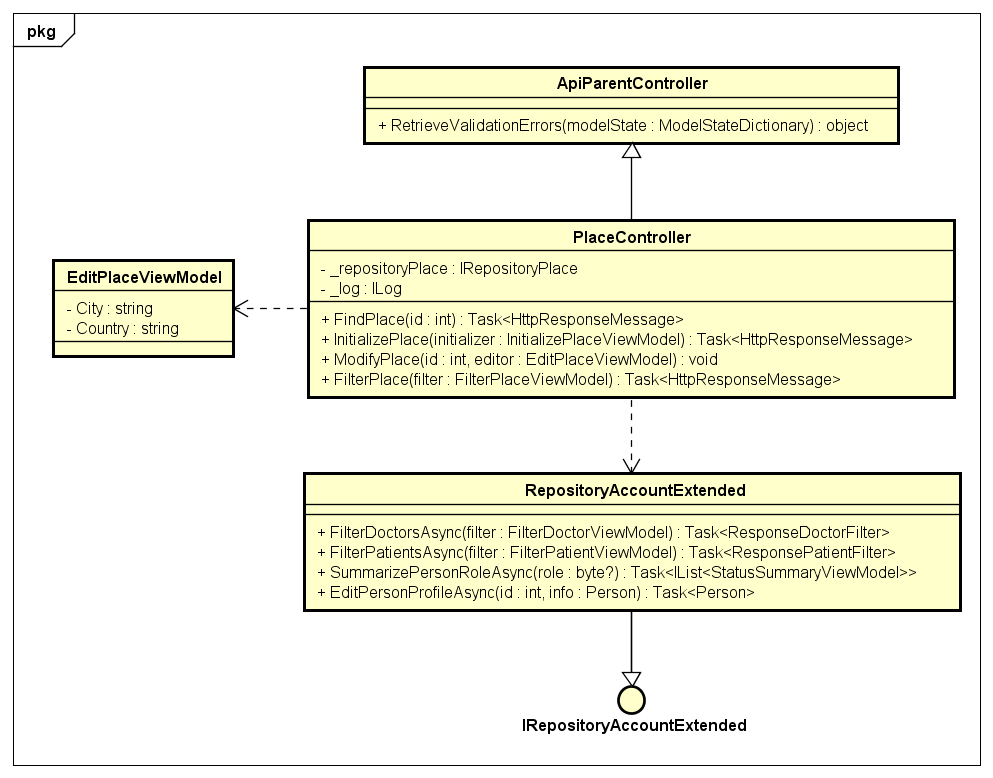
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Take the information requested from client, do validation and modify place information as the parameters are valid. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Contains the information which will be thrown back to the clients. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of place which needs modifying |
| 2 | initializer | EditPlaceViewModel | Collection of information which is used for replacing the old ones. |

Method: FilterPlace()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Take the information requested from client, do validation filter places list as parameters are valid. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Contains the information which will be thrown back to the clients. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 2 | filter | FilterPlaceViewModel | Class which contains conditions to filter places. |

##### 6.2.1.12. Edit place

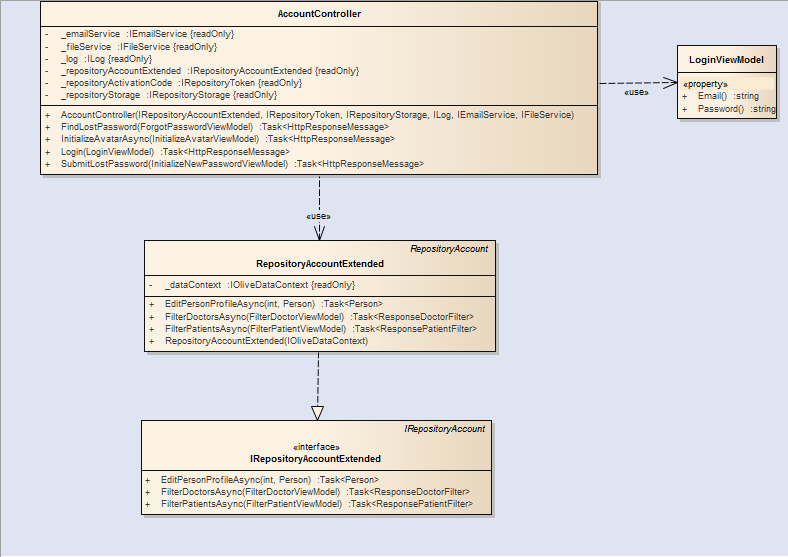
1. Class diagram



#### 6.2.2. Olives sub-system (Back-end)

##### 6.2.1.1. Login

1. Class diagram



1. Class definition

PlaceController class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_dataContext | IOliveDataContext | Instance of interface using to call methods in IOliveDataContext. |

Method: FilterDoctorsAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Find doctors by using specific conditions. | | | |
| Return | **Type** | | | **Description** |
| ResponseDoctorFilter | | | Contains a list of filtered doctors and number which represents total filtered results. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | FilterDoctorViewModel | Model contains conditions which are used for filtering conditions. |

Method: FilterPatientsAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Find patients by using specific conditions. | | | |
| Return | **Type** | | | **Description** |
| ResponsePatientFilter | | | Contains a list of filtered patients and number which represents total filtered results. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | FilterDoctorViewModel | Model contains conditions which are used for filtering conditions. |

Method: EditPersonProfileAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Edit person by using specific information. | | | |
| Return | **Type** | | | **Description** |
| Person | | | Instance which contains edited information. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of person whose profile should be changed. |
| 2 | info | Person | Instance contains information which contain information which is used for information change. |

AccountController class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_repositoryStorage | IRepositoryStorage | Instance of interface using to call methods in IRepositoryStorage. |
| 2 | \_repositoryAccountExtended | IRepositoryAccountExtended | Instance of interface using to call methods in IRepositoryAccountExtended. |
| 3 | \_repositoryActivationCode | IRepositoryToken | Instance of interface using to call methods in IRepositoryToken. |
| 4 | \_log | ILog | Instance of interface using to call methods in ILog. |
| 5 | \_emailService | IEmailService | Instance of interface using to call methods in IEmailService. |
| 6 | \_fileService | IFileService | Instance of interface using to call methods in IFileService. |

Method: InitializeAvatarAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Find doctors by using specific conditions. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Information which will be responded to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | info | ForgotPasswordViewModel | Model contains information which is used for changing password request. |

Method: SubmitLostPassword()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | The function takes parameters requested from clients, do validation and changing the new password. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Information which will be responded to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | initializer | InitializeNewPasswordViewModel | Model contains information of token and password. |

Method: Login()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for taking parameters, doing validation and logging user into system. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Information which will be responded to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | loginViewModel | LoginViewModel | Information contains information of email and password for logging action. |

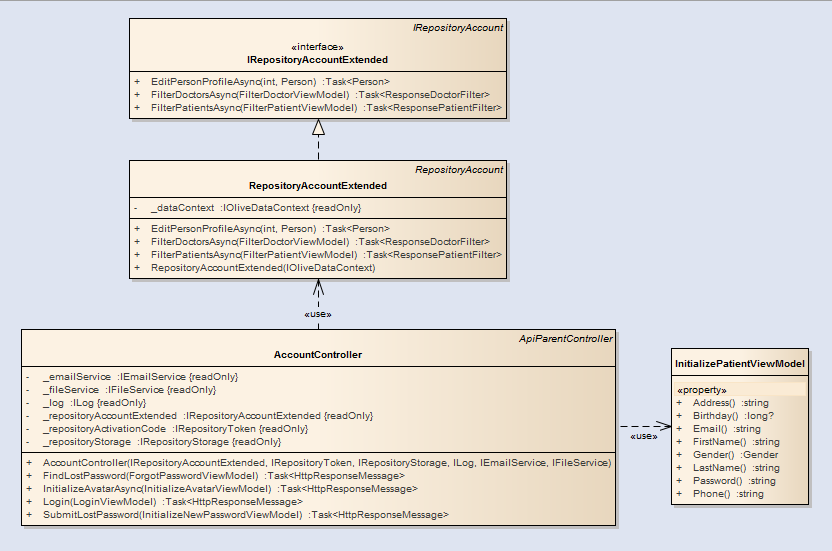
LoginViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Email | string | Information of email. |
| 2 | Password | string | Password of account. |

##### 6.2.1.2. Patient register account

1. Class diagram



1. Class definition

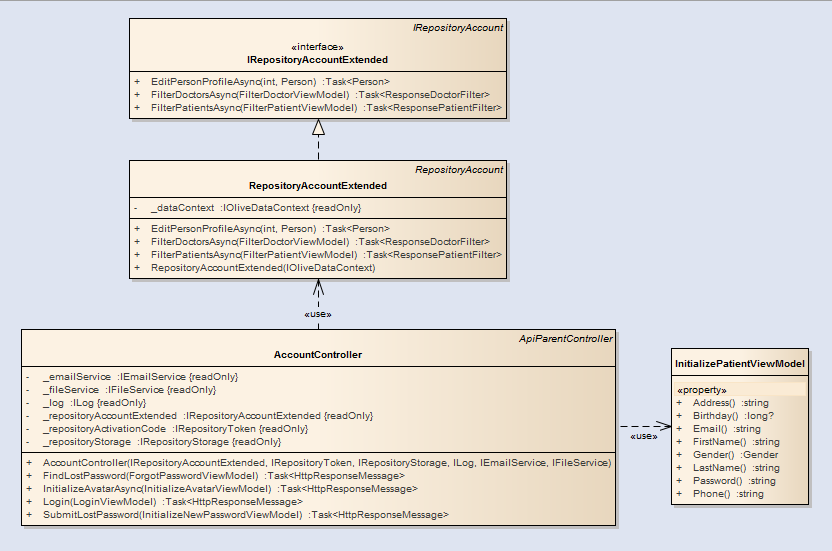
InitializePatientViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | FirstName | string | First name of patient. |
| 2 | LastName | string | Last name of patient. |
| 3 | Birthday | double | Birthday of patient, the format is UTC. |
| 4 | Gender | Gender | Gender of patient, by default gender is: Gender.Male. |
| 5 | Email | string | Email which is used for logging into system. |
| 6 | Password | string | Password of account. |
| 7 | Phone | string | Phone number of patient. |
| 8 | Address | string | Address where patient lives. |

##### 6.2.1.3. Patient register account

1. Class diagram



1. Class definition

InitializePatientViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | FirstName | string | First name of patient. |
| 2 | LastName | string | Last name of patient. |
| 3 | Birthday | double | Birthday of patient, the format is UTC. |
| 4 | Gender | Gender | Gender of patient, by default gender is: Gender.Male. |
| 5 | Email | string | Email which is used for logging into system. |
| 6 | Password | string | Password of account. |
| 7 | Phone | string | Phone number of patient. |
| 8 | Address | string | Address where patient lives. |

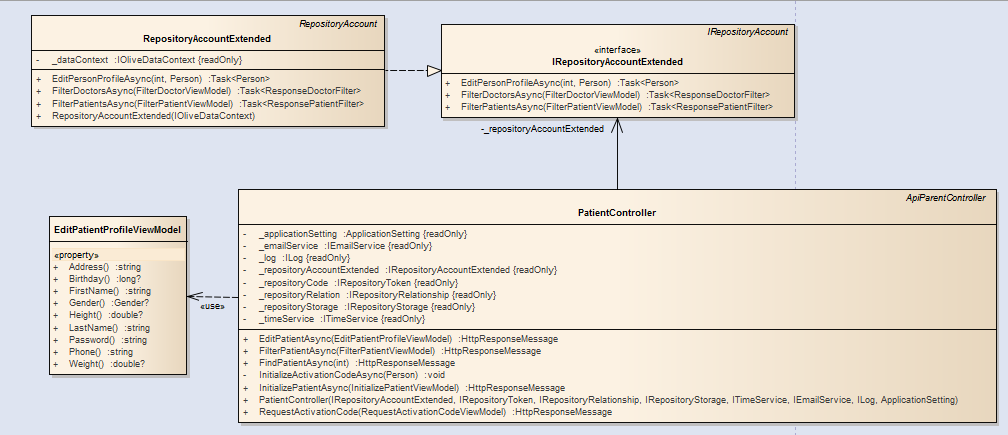
##### 6.2.1.4. Forgot password

1. Sequence diagram

<TODO> Fill fill fill

##### 6.2.1.5. Request change password

1. Class diagram



1. Sequence diagram

<TODO : FILL FILL FILL>

1. Class definition

PatientController class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_repositoryAccountExtended | IRepositoryAccountExtended | Instance which is used for accessing function to IRepositoryAccountExtended. |
| 2 | \_repositoryRelation | IRepositoryRelationship | Instance which is used for accessing function to IRepositoryRelationship. |
| 3 | \_repositoryStorage | IRepositoryStorage | Instance which is used for accessing function to IRepositoryStorage. |
| 4 | \_log | ILog | Instance which is used for accessing function to ILog. |
| 5 | \_applicationSetting | ApplicationSetting | Instance which is used for accessing function to ApplicationSetting. |
| 6 | \_timeService | ITimeService | Instance which is used for accessing function to ITimeService. |
| 7 | \_repositoryCode | IRepositoryToken | Instance which is used for accessing function to IRepositoryToken. |
| 8 | \_emailService | IEmailService | Instance which is used for accessing function to IEmailService. |

Method: FindPatientAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | The function takes parameters requested from client and accessing database to find patient by using id. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Information which will be responded to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of patient. |

Method: InitializePatientAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | The function takes parameters requested from client and initialize a patient object with the given information. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Information which will be responded to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | info | InitializePatientViewModel | Information which contains patient’s information. |

Method: EditPatientAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | The function takes parameters requested from client and edit patient profile. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Information which will be responded to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | editor | EditPatientProfileViewModel | Information which contains patient’s information. |

Method: FilterPatientAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | The function takes parameters requested from client and filter patients list. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Information which will be responded to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | FilterPatientViewModel | Model contains information which is used for filtering patients. |

Method: RequestActivationCode()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function takes information from client to initialize activation code for patient. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Information which will be responded to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | initializer | RequestActivationCodeViewModel | Model contains information which is used for initializing patient activation token. |

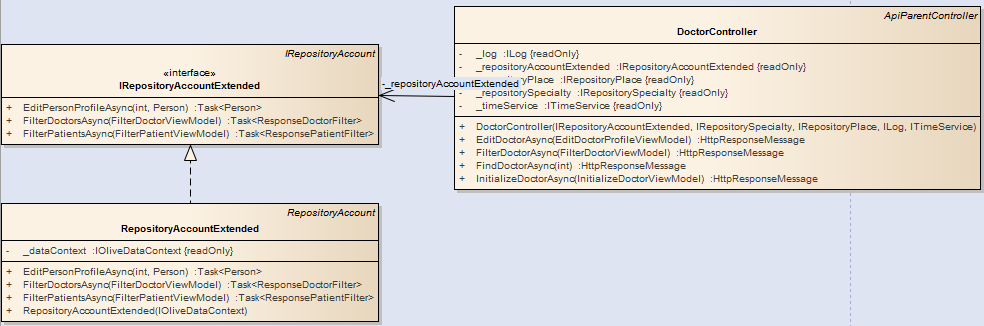
EditPatientViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | FirstName | string | First name of patient. |
| 2 | LastName | string | Last name of patient. |
| 3 | Birthday | double | Birthday of patient, the format is UTC. |
| 4 | Gender | Gender | Gender of patient, by default gender is: Gender.Male. |
| 5 | Email | string | Email which is used for logging into system. |
| 6 | Password | string | Password of account. |
| 7 | Phone | string | Phone number of patient. |
| 8 | Address | string | Address where patient lives. |
| 9 | Weight | double? | Patient weight. |
| 10 | Height | double? | Patient height. |

##### 6.2.1.6. Find doctor

1. Class diagram



1. Class definition

DoctorController class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_repositoryAccountExtended | IRepositoryAccountExtended | Instance which is used for accessing function to IRepositoryAccountExtended. |
| 2 | \_repositorySpecialty | IRepositoryRelationship | Instance which is used for accessing function to IRepositorySpecialty. |
| 3 | \_repositoryPlace | IRepositoryStorage | Instance which is used for accessing function to IRepositoryStorage. |
| 4 | \_log | ILog | Instance which is used for accessing function to ILog. |
| 5 | \_timeService | ITimeService | Instance which is used for accessing function to ITimeService. |

Method: FindDoctorAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | The function takes parameters requested from client and accessing database to find doctor by using id. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Information which will be responded to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of doctor. |

Method: InitializeDoctorAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | The function takes parameters requested from client, do validation and initialize a new patient into database. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Information which will be responded to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | initializer | InitializeDoctorViewModel | Information of doctor. |

Method: EditDoctorAsync()

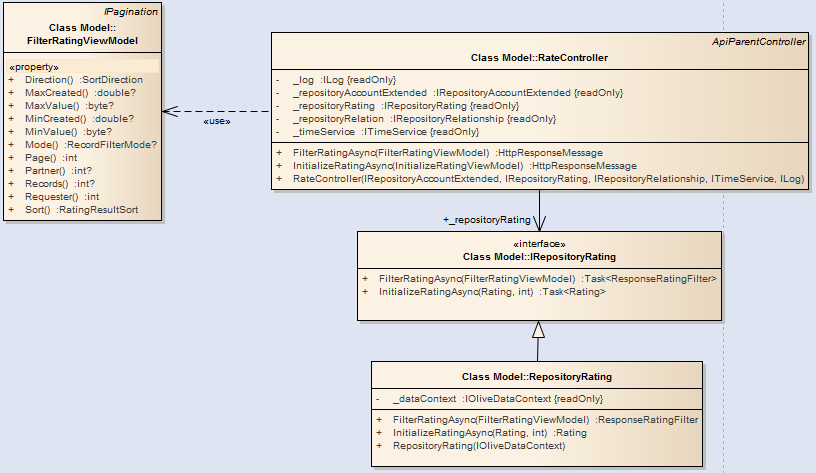
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | The function takes parameters requested from client, do validation and modify a new patient into database. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Information which will be responded to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | editor | EditDoctorProfileViewModel | Information of doctor. |

Method: FilterDoctorAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | The function takes parameters requested from client, do validation and filter the list of doctors from database. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Information which will be responded to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | FilterDoctorViewModel | Conditions which are used for filtering. |

##### 6.2.1.7. Filter doctor rate

1. Class diagram



1. Class definition

DoctorController class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_repositoryAccountExtended | IRepositoryAccountExtended | Instance which is used for accessing function to IRepositoryAccountExtended. |
| 2 | \_repositoryRating | IRepositoryRating | Instance which is used for accessing function to IRepositoryRating. |
| 3 | \_repositoryRelation | IRepositoryRelationship | Instance which is used for accessing function to IRepositoryRelationship. |
| 4 | \_log | ILog | Instance which is used for accessing function to ILog. |
| 5 | \_timeService | ITimeService | Instance which is used for accessing function to ITimeService. |

Method: InitializeRatingAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | The function takes parameters requested from client and initializing a doctor rating into database. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Information which will be responded to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | initializer | InitializeRatingViewModel | Information of rating. |

Method: FilterRatingAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | The function takes parameters requested from client and filtering doctor ratings. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Information which will be responded to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | FilterRatingViewModel | Model contains information which are used for filtering ratings. |

RepositoryRating class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_repositoryAccountExtended | IRepositoryAccountExtended | Instance which is used for accessing function to IRepositoryAccountExtended. |
| 2 | \_repositoryRating | IRepositoryRating | Instance which is used for accessing function to IRepositoryRating. |
| 3 | \_repositoryRelation | IRepositoryRelationship | Instance which is used for accessing function to IRepositoryRelationship. |
| 4 | \_log | ILog | Instance which is used for accessing function to ILog. |
| 5 | \_timeService | ITimeService | Instance which is used for accessing function to ITimeService. |

Method: InitializeRatingAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for initializing rating to database. | | | |
| Return | **Type** | | | **Description** |
| Rating | | | Information of rating which has been created. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | rated | int | Id of doctor who is rated. |
| 2 | rating | Rating | Rating information. |

Method: FilterRatingAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for initializing rating to database. | | | |
| Return | **Type** | | | **Description** |
| Rating | | | Information of rating which has been created. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | FilterRatingViewModel | Conditions which are used for filtering rating. |

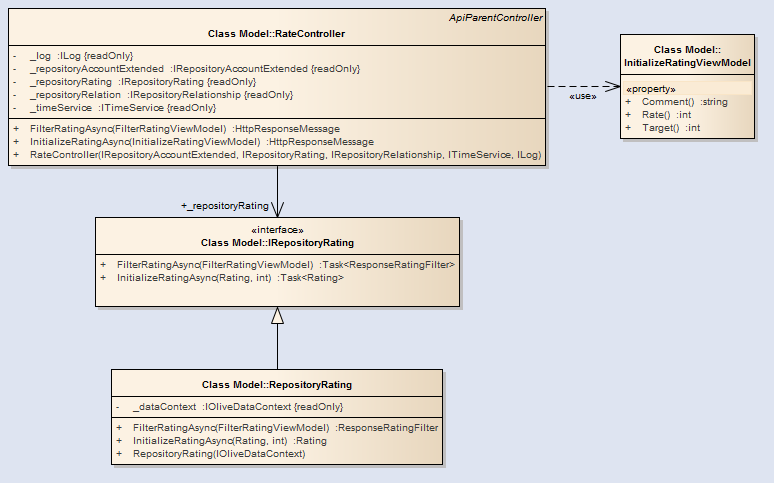
RepositoryRating class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Requester | Person | Person who created filter request. |
| 2 | Partner | int | Id of partner. |
| 3 | Mode | RecordFilterMode? | Whether all ratings should be taken or not. |
| 4 | MinCreated | double? | Time after which rating is created. |
| 5 | MaxCreated | double? | Time before which rating had been created. |
| 6 | MinValue | double? | Minimum rating value. |
| 7 | MaxValue | double? | Maximum rating value. |
| 8 | Sort | RatingResultSort | Which property should be used for sorting purpose. |
| 9 | Direction | SortDirection | Whether property should be sorted ascendingly or decendingly. |
| 10 | Page | IRepositoryRelationship | Index of page result. |
| 11 | Records | IRepositoryRelationship | Number of records per page. |

##### 6.2.1.8. Initialize doctor rating

1. Class diagram



1. Class definition

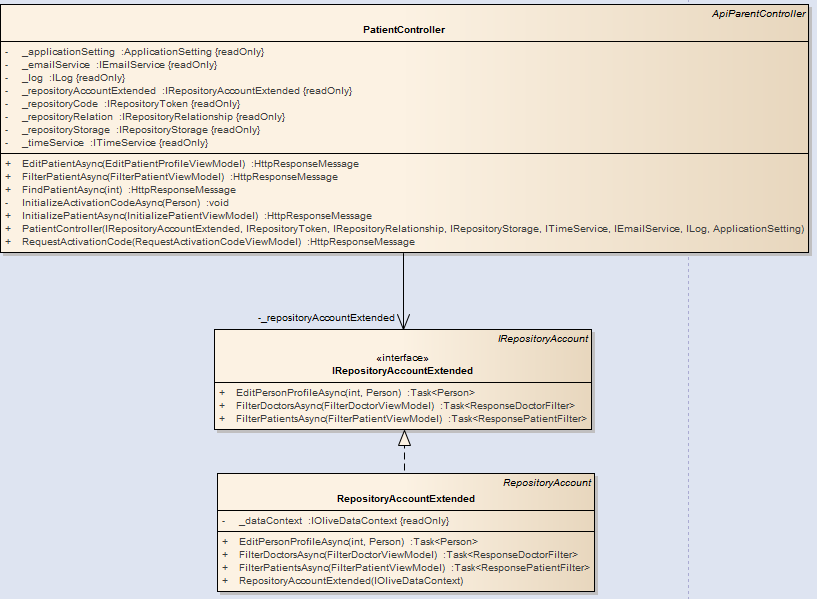
InitializeRatingViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Comment | string | Comment of rating. |
| 2 | Rate | int | Rating point. |
| 3 | Target | int | Doctor who is the target of rating. |

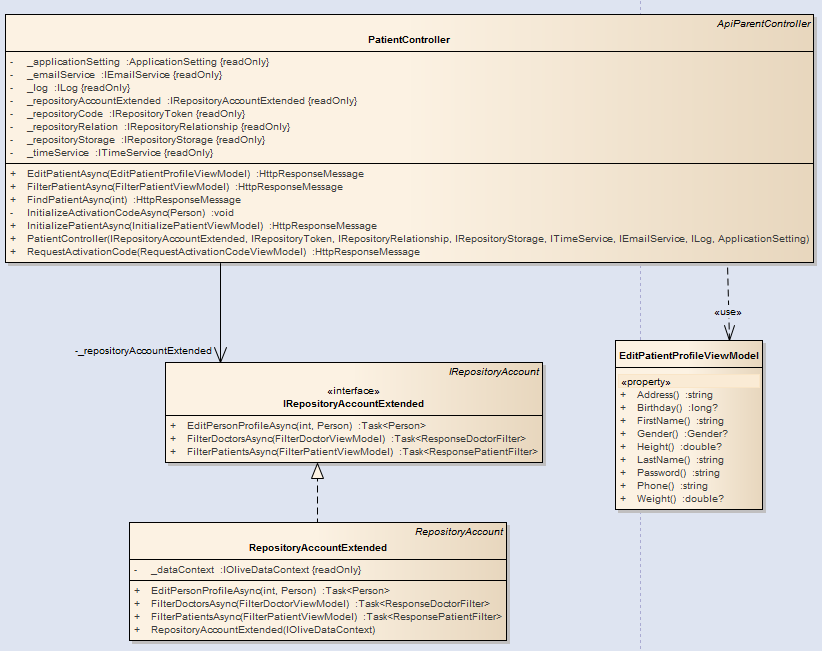
##### 6.2.1.9. Patient finds personal profile

1. Class diagram



##### 6.2.1.10. Patient edit profile

1. Class diagram



1. Class definition

EditPatientProfileViewModel class

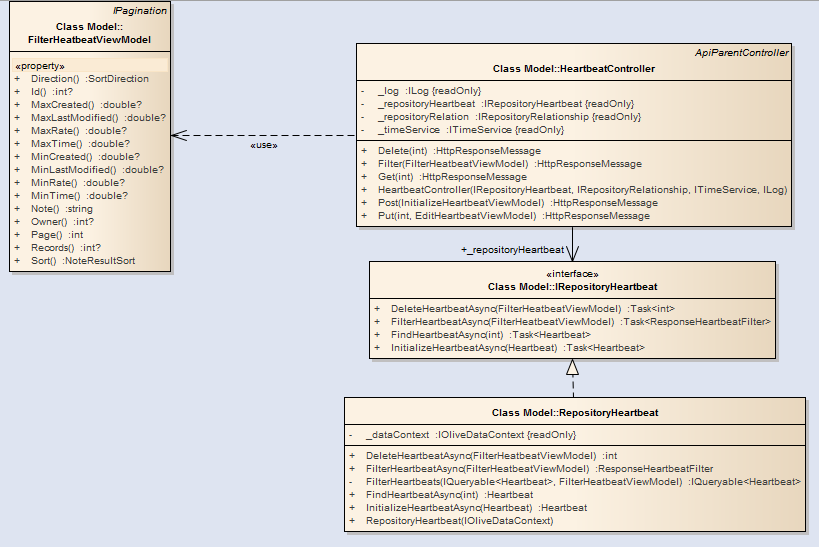
Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Address | string | Address of patient. |
| 2 | Birthday | double? | Birthday of patient. |
| 3 | FirstName | IRepositoryRelationship | First name of patient. |
| 4 | Gender | ILog | Gender of patient. |
| 5 | Height | ITimeService | Height of patient. |
| 6 | LastName | ITimeService | Last name of patient. |
| 7 | Password | ITimeService | Password of patient account. |
| 8 | Phone | ITimeService | Phone which patient uses. |
| 9 | Weight | ITimeService | Weight of patient. |

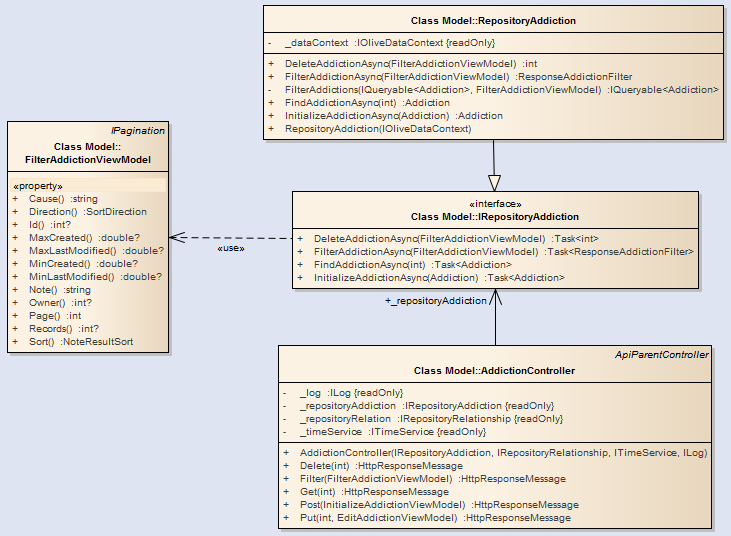
##### 6.2.1.11. Filter daily record

* Dialy records can belong to: Allergy, heartbeat, addiction, blood pressure, blood sugar.
* The data flow is : View -> <name>Controller -> IRepository<name> -> Repository<Name>.
* Name is one of the following: Heartbeat, Allergy, Addiction, BloodPressure, BloodSugar.
* They all have same diagrams and sequence diagrams.

1. Class diagram



Class diagram of Heartbeat view.



Class diagram of Addiction view.

1. Class definition

<name>Controller class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_repository<name> | IRepository<name> | Instance which is used for accessing function to IRepository<name>. |
| 2 | \_repositoryRelation | IRepositoryRelationship | Instance which is used for accessing function to IRepositoryRelationship. |
| 3 | \_log | ILog | Instance which is used for accessing function to ILog. |
| 4 | \_timeService | ITimeService | Instance which is used for accessing function to ITimeService. |

Method: Get()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for searching <name> note from database. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Data which will be sent to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of **<name>** note. |

Method: Post()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for initializing <name> note to database. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Data which will be sent to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | info | Initialize**<name>**ViewModel | Information of **<name>** note which will be added into database. |

Method: Put()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for searching for a <name> note and updating its information. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Data which will be sent to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of **<name>** note whose information will be updated. |
| 2 | modifier | Initialize**<name>**ViewModel | Information of **<name>** note which will be updated into database. |

Method: Delete()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for searching for a <name> note and deleting its information. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Data which will be sent to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of **<name>** note whose information will be deleted. |

Method: Filter()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for filtering <name> notes by using specific conditions. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Data which will be sent to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | Filter**<name>**ViewModel | Conditions which are used for filtering **<name>** notes. |

Repository<name> class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_dataContext | IOliveDataContext | Instance which is used for accessing function to IOliveDataContext. |

Method: Initialize<name>Async()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for searching <name> note from database. | | | |
| Return | **Type** | | | **Description** |
| <name> | | | Information of <name> note which has been initialized. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | info | <name> | <name> note information. |

Method: Find<name>Async ()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for searching heartbeat note from database. | | | |
| Return | **Type** | | | **Description** |
| <name> | | | Information of <name> which is found in database. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of <name> note. |

Method: Delete<name>Async ()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for searching <name> note from database. | | | |
| Return | **Type** | | | **Description** |
| int | | | Number of records which have been deleted. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | Filter<name>ViewModel | Conditions which are used for finding <name> notes. |

Method: Filter<name>Async ()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for filtering <name> notes from database. | | | |
| Return | **Type** | | | **Description** |
| Response<name>Filter | | | Model contains list of filtered <name> notes and total number of filtered result. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | Filter<name>ViewModel | Conditions which are used for finding <name> notes. |

FilterHeatbeatViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Id | int? | Id of heartbeat note. |
| 3 | Owner | int? | Id of note owner. |
| 4 | MinRate | double? | Minimum heart rate. |
| 5 | MaxRate | double? | Maximum heart rate. |
| 6 | Note | string | Note of heartbeat measurement. |
| 7 | MinTime | double? | Time which note is about. |
| 8 | MaxTime | double? | Time which note is about. |
| 9 | MinCreated | double? | Time after which note was submitted to server. |
| 10 | MaxCreated | double? | Time before which note had been submitted to server. |
| 11 | MinLastModified | double? | Time after which note was lastly modified. |
| 12 | MaxLastModified | double? | Time before whih note had been lastly modified. |
| 13 | Sort | NoteResultSort | Property which will be used for sorting. |
| 14 | Direction | SortDirection | Whether records are sorted ascendingly or decendingly. |
| 15 | Page | int | Index of result page. |
| 16 | Records | int | Number of record per page. |

FilterAddictionViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Id | int? | Id of note. |
| 3 | Owner | int? | Id of note owner. |
| 4 | Cause | string | Which cause patient be allergic with. |
| 5 | Note | string | Note information. |
| 6 | MinCreated | double? | Time after which note was submitted to server. |
| 7 | MaxCreated | double? | Time before which note had been submitted to server. |
| 8 | MinLastModified | double? | Time after which note was lastly modified. |
| 9 | MaxLastModified | double? | Time before whih note had been lastly modified. |
| 10 | Sort | NoteResultSort | Property which will be used for sorting. |
| 11 | Direction | SortDirection | Whether records are sorted ascendingly or decendingly. |
| 12 | Page | int | Index of result page. |
| 13 | Records | int | Number of record per page. |

FilterAllergyViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Id | int? | Id of note. |
| 3 | Owner | int? | Id of note owner. |
| 4 | Name | string | Allergy name. |
| 5 | Cause | string | Which cause patient be addictive with. |
| 6 | Note | string | Note information. |
| 7 | MinCreated | double? | Time after which note was submitted to server. |
| 8 | MaxCreated | double? | Time before which note had been submitted to server. |
| 9 | MinLastModified | double? | Time after which note was lastly modified. |
| 10 | MaxLastModified | double? | Time before whih note had been lastly modified. |
| 11 | Sort | NoteResultSort | Property which will be used for sorting. |
| 12 | Direction | SortDirection | Whether records are sorted ascendingly or decendingly. |
| 13 | Page | int | Index of result page. |
| 14 | Records | int | Number of record per page. |

FilterBloodPressureViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Id | int? | Id of note. |
| 3 | Owner | int? | Id of note owner. |
| 4 | MinDiastolic | string | Value which diastolic must be greater than. |
| 5 | MaxDiastolic | string | Value which diastolic must be lower than. |
| 6 | MinSystolic | string | Value which systolic must be greater than. |
| 7 | MaxSystolic | string | Value which systolic must be greater than |
| 8 | Note | string | Note information. |
| 9 | MinCreated | double? | Time after which note was submitted to server. |
| 10 | MaxCreated | double? | Time before which note had been submitted to server. |
| 11 | MinLastModified | double? | Time after which note was lastly modified. |
| 12 | MaxLastModified | double? | Time before whih note had been lastly modified. |
| 13 | Sort | NoteResultSort | Property which will be used for sorting. |
| 14 | Direction | SortDirection | Whether records are sorted ascendingly or decendingly. |
| 15 | Page | int | Index of result page. |
| 16 | Records | int | Number of record per page. |

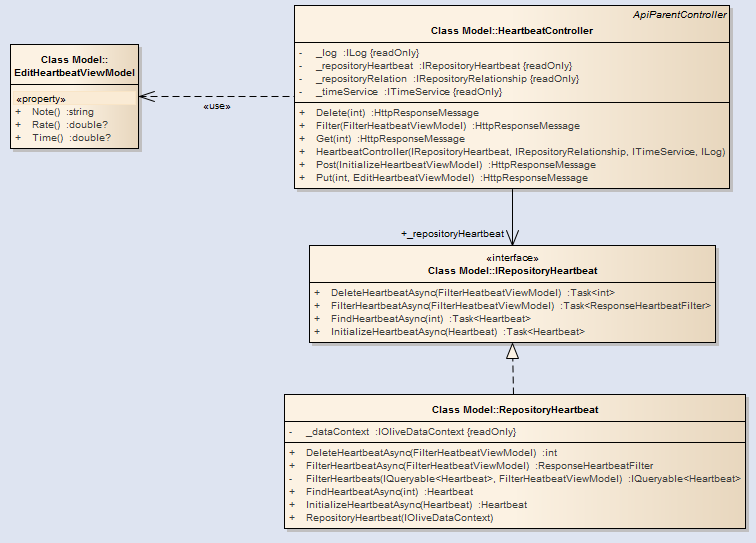
FilterBloodSugarViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Id | int? | Id of note. |
| 2 | Owner | int? | Id of note owner. |
| 3 | Note | string | Note information. |
| 4 | MinValue | double? | Minimum value of note. |
| 5 | MaxValue | double? | Maximum value of note. |
| 6 | MinCreated | double? | Time after which note was submitted to server. |
| 7 | MaxCreated | double? | Time before which note had been submitted to server. |
| 8 | MinLastModified | double? | Time after which note was lastly modified. |
| 9 | MaxLastModified | double? | Time before whih note had been lastly modified. |
| 10 | Sort | NoteResultSort | Property which will be used for sorting. |
| 11 | Direction | SortDirection | Whether records are sorted ascendingly or decendingly. |
| 12 | Page | int | Index of result page. |
| 13 | Records | int | Number of record per page. |

##### 6.2.1.12. Edit daily record

1. Class diagram



1. Class definition

EditHeartbeatViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Note | string | Note of heartbeat measurement. |
| 2 | Rate | double? | Rate of note. |
| 3 | Time | double? | Time which note is about. |

EditAddictionViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Cause | string | Cause of addiction. |
| 2 | Note | double? | Note of addiction. |

EditAllergyViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Name | string | Name of allergy. |
| 2 | Cause | double? | Cause of allergy. |
| 3 | Note | double? | Note of allergy. |

EditBloodPressureViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Diastolic | string | Blood minimum pressure. |
| 2 | Systolic | double? | Blood maximum pressure. |
| 3 | Time | double? | Time of blood pressure is measurement was done. |
| 4 | Note | string | Note of blood pressure measurement. |

EditBloodSugarViewModel class

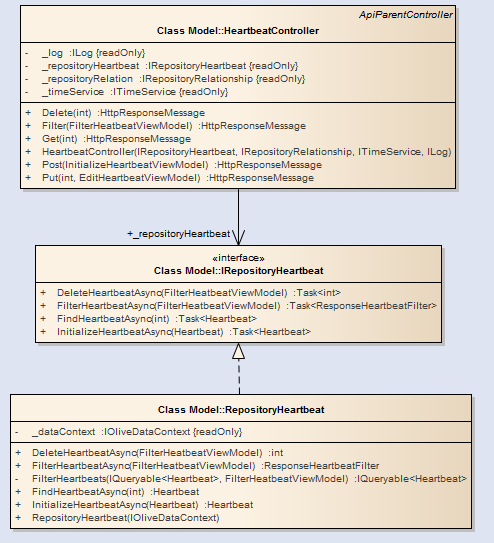
|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Value | string | Blood sugar level. |
| 2 | Time | double? | Time when the measurement was done. |
| 3 | Note | double? | Note of measurement. |

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Diastolic | string | Blood minimum pressure. |
| 2 | Systolic | double? | Blood maximum pressure. |
| 3 | Time | double? | Time of blood pressure is measurement was done. |
| 3 | Note | string | Note of blood pressure measurement. |

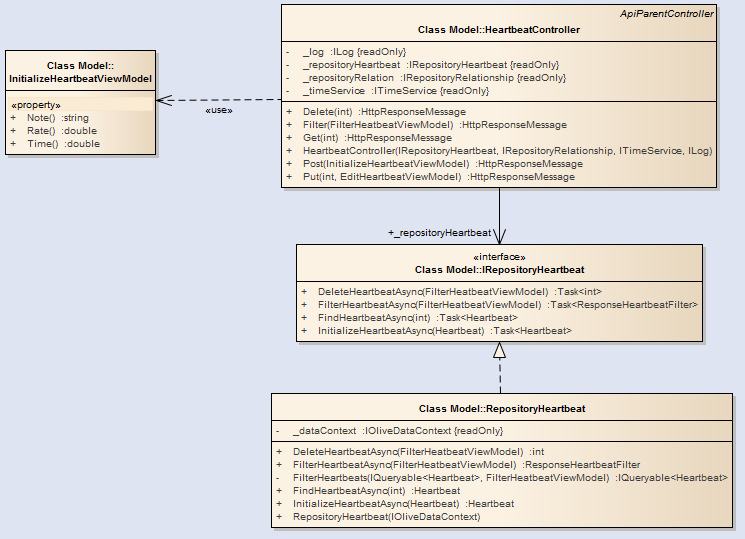
##### 6.2.1.13. Delete daily record

1. Class diagram



##### 6.2.1.14. Initialize daily record

1. Class diagram



1. Class definition

InitializeHeartbeatViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Note | string | Note of heartbeat measurement. |
| 2 | Rate | double | Rate of note. |
| 3 | Time | double | Time which note is about. |

InitializeHeartbeatViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Note | string | Note of heartbeat measurement. |
| 2 | Rate | double | Rate of note. |
| 3 | Time | double | Time which note is about. |

InitializeAddictionViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Cause | string | Note of heartbeat measurement. |
| 2 | Note | double | Rate of note. |

InitializeAllergyViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Name | string | Name of allergy. |
| 2 | Cause | string | Cause of allergy. |
| 3 | Note | string | Note of allergy. |

InitializeBloodPressureViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Diastolic | int | Blood minimum pressure. |
| 2 | Systolic | int | Blood maximum pressure. |
| 3 | Time | double | Time when the measurement is done. |
| 4 | Note | string | Note of measurement. |

InitializeBloodPressureViewModel class

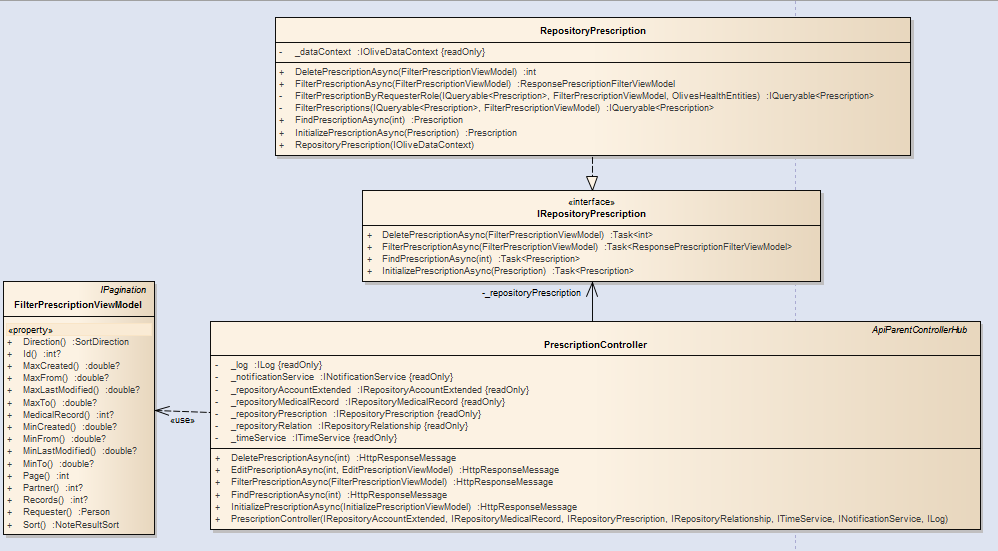
Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Value | int | Blood pressure value. |
| 2 | Time | int | Time when measurement is done. |
| 3 | Note | double | Note of measurement. |

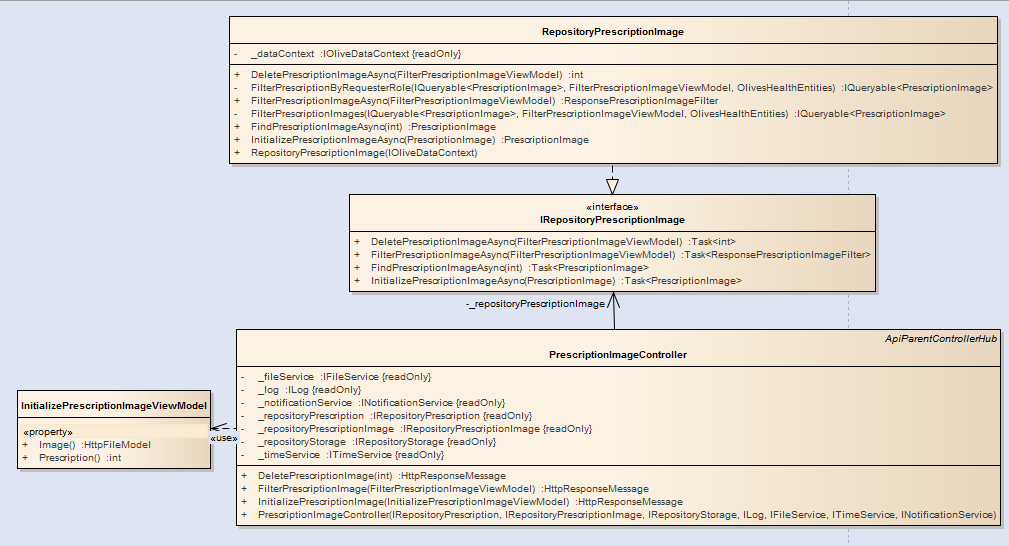
##### 6.2.1.15. View Prescription

* Prescription contains : Prescription and Prescription image.
* They all following the naming convention: <name>Controller, <name>ViewModel, Repository<name>, IRepository<name>

1. Class diagram



Class diagram of Prescription view.



Class diagram of Prescription image view.

1. Class definition

<name>Controller class

Method: Filter<name>Async ()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for filtering <name> notes from database. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Model contains information which will be thrown to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | Filter**<name>**ViewModel | Conditions which are used for finding <name> notes. |

Method: Initialize<name>Async ()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for initializing <name> notes from database. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Model contains information which will be thrown to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | initializer | Initialize**<name>**ViewModel | Model contains information which will be used for initializing. |

Method: Edit<name>Async ()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for modifying <name> notes from database. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Model contains information which will be thrown to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of record which will be modified. |
| 2 | info | Edit**<name>**ViewModel | Model contains information for updating record. |

Method: Delete<name>Async ()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for modifying <name> notes from database. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Model contains information which will be thrown to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of record which will be deleted. |

Repository<name> class

Method: Filter<name>Async ()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for filtering <name> notes from database. | | | |
| Return | **Type** | | | **Description** |
| Response<name>FilterViewModel | | | Model contains the list of filtered result and number which represents total filtered records. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | Filter**<name>**ViewModel | Conditions which are used for finding <name> notes. |

Method: Find<name>Async ()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for filtering <name> notes from database. | | | |
| Return | **Type** | | | **Description** |
| <name> | | | Instance which have been searched from database. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Index of result which will be used for searching in database. |

Method: Delete<name>Async ()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for deleting <name> notes from database by using specific conditions. | | | |
| Return | **Type** | | | **Description** |
| int | | | Number of deleted records. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | Filter<name>ViewModel | Model contains conditions which are used for filtering results which needed deleting. |

Method: Initialize<name>Async ()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | This function is for initializing/modifying <name> notes into/from database. | | | |
| Return | **Type** | | | **Description** |
| <name> | | | Instance which has been created/modified. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | initializer | <name> | Model contain information which will be updated/initialized. |

InitializePrescriptionViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | MedicalRecord | int | Id of medical record. |
| 2 | Name | string | Name of prescription. |
| 3 | From | double | When the prescription should be available. |
| 4 | To | double | When the prescription should be expired. |
| 4 | Note | string | Note of prescription. |
| 5 | Medicines | Dictionary<string, MedicineInfoViewModel> | List of medicines which are included in prescription. |

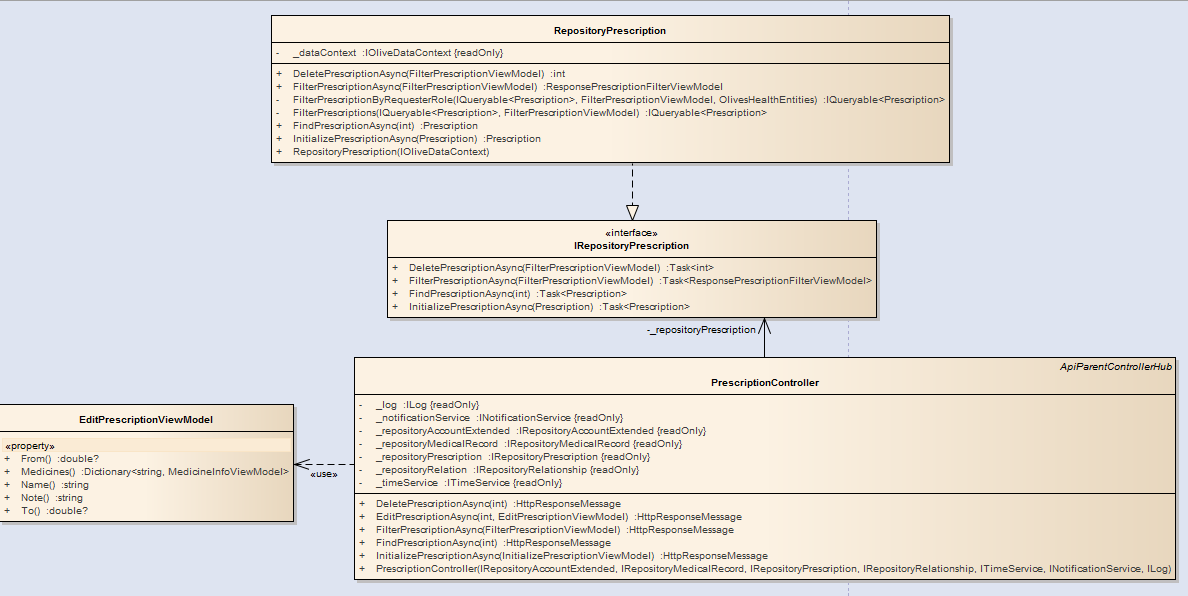
InitializePrescriptionImageViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Prescription | int | Prescription id. |
| 2 | Image | string | Byte stream of file submitted to server. |

##### 6.2.1.16. Edit Prescription

1. Class diagram



1. Class definition

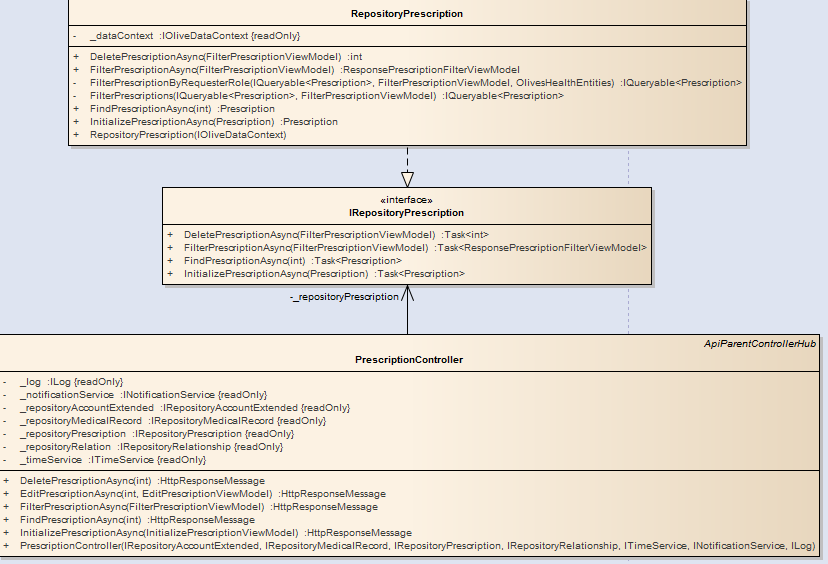
EditPrescriptionViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Name | int | Name of prescription. |
| 2 | From | string | When the prescription should be available. |
| 3 | To | double | When the prescription should be expired. |
| 4 | Dictionary<string, MedicineInfoViewModel> | double | Information of medicines inside a prescription. |
| 5 | Note | string | Note of prescription. |

##### 6.2.1.17. Delete Prescription

1. Class diagram



##### 6.2.1.18. View medical record

* Medical records can be: **Medical record**, **Medical images**, **Medical note**, **Experiment note**.
* The data flow is: View -> <name>Controller -> IRepository<name> -> Repository<Name>.
* Name is one of the following: **MedicalRecord**, **MedicalImage**, **MedicalNote**, **ExperimentNote**.
* They all have same diagrams and sequence diagrams.

1. Class diagram

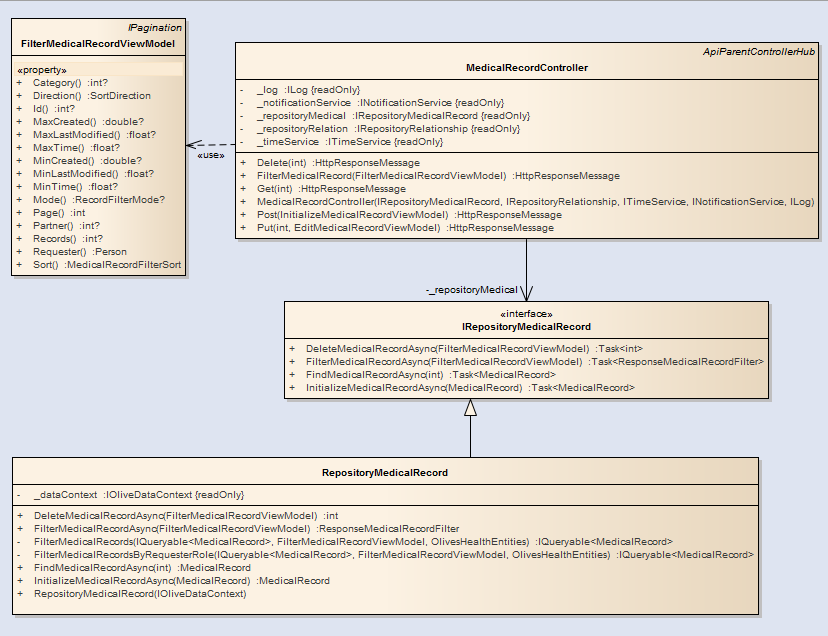


Diagram of Medical Record view.

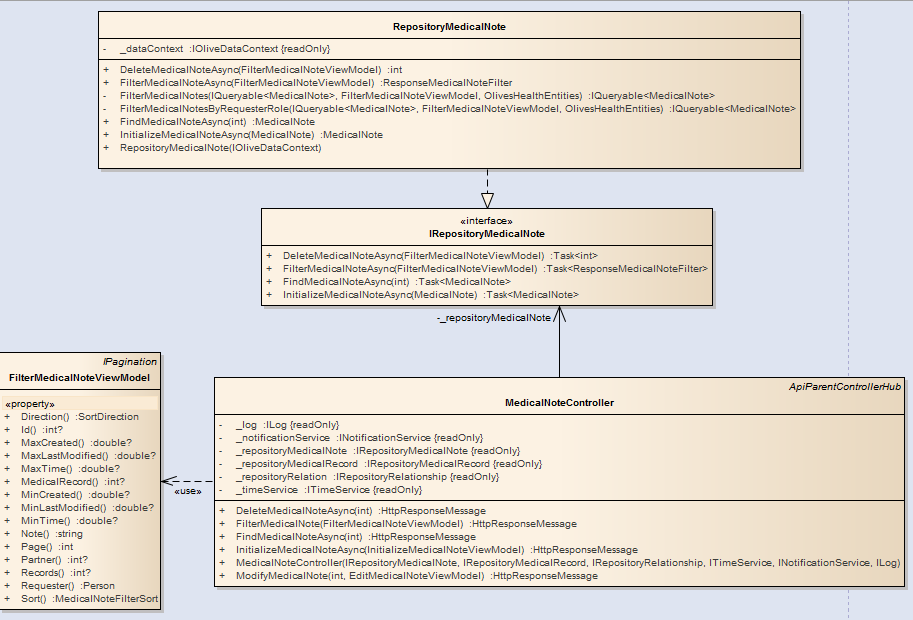


Diagram of Medical Note view.

1. Class definition

FilterMedicalNoteViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Id | int | Id of medical note. |
| 2 | MedicalRecord | int | Medical record id which note belongs to. |
| 3 | Requester | Person | Requester who sent the filter request. |
| 4 | Partner | int? | Index of person who takes part in prescription. |
| 5 | Note | string | Note of medical note. |
| 6 | MinTime | double? | Time which note is about. |
| 7 | MaxTime | double? | Time which note is about. |
| 8 | MinCreated | double? | Time after which note was lastly created. |
| 9 | MaxCreated | double? | Time before which note had been created. |
| 10 | MinLastModified | double? | Time after which note was lastly modified. |
| 11 | MaxLastModified | double? | Time before which note had been modified. |
| 12 | Sort | MedicalNoteFilterSort | Which property should be sorted. |
| 13 | Direction | SortDirection | Whether records should be sorted ascendingly or decendingly. |
| 14 | Page | string | Index of result page. |
| 15 | Records | string | Results per page. |

FilterMedicalRecordViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Id | int | Id of medical record. |
| 2 | Requester | Person | Requester who sent filter request. |
| 3 | Partner | int? | Who takes part in the medical record |
| 4 | Mode | RecordFilterMode? | Base on role of requester to filter medical records. |
| 5 | MinTime | double? | Time which the medical record is about. |
| 6 | MaxTime | double? | Time which the medical record is about. |
| 7 | Category | int | Index of category which medical record belongs to. |
| 8 | MinCreated | double? | Time after which medical record was created. |
| 9 | MaxCreated | double? | Time before which medical record had been created. |
| 10 | MinLastModified | double? | Time after which medical record was lastly modified. |
| 11 | MaxLastModified | double? | Time before which medical record had been modified. |
| 12 | Sort | MedicalRecordFilterSort? | Which property should be used for sorting. |
| 12 | Direction | SortDirection | Whether records should be sorted ascendingly or decendingly. |
| 12 | Page | int | Index of result page. |
| 12 | Records | int | Records number per page. |

FilterMedicalImageViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Id | int | Id of medical image. |
| 2 | Requester | Person | Requester who sent filter request. |
| 3 | Partner | int? | Who takes part in the medical record |
| 4 | MedicalRecord | int | Index of medical record. |
| 4 | Mode | RecordFilterMode? | Base on role of requester to filter medical records. |
| 8 | MinCreated | double? | Time after which medical record was created. |
| 9 | MaxCreated | double? | Time before which medical record had been created. |
| 10 | Page | int | Index of result page. |
| 11 | Records | int | Records number per page. |

FilterExperimentNoteViewModel class

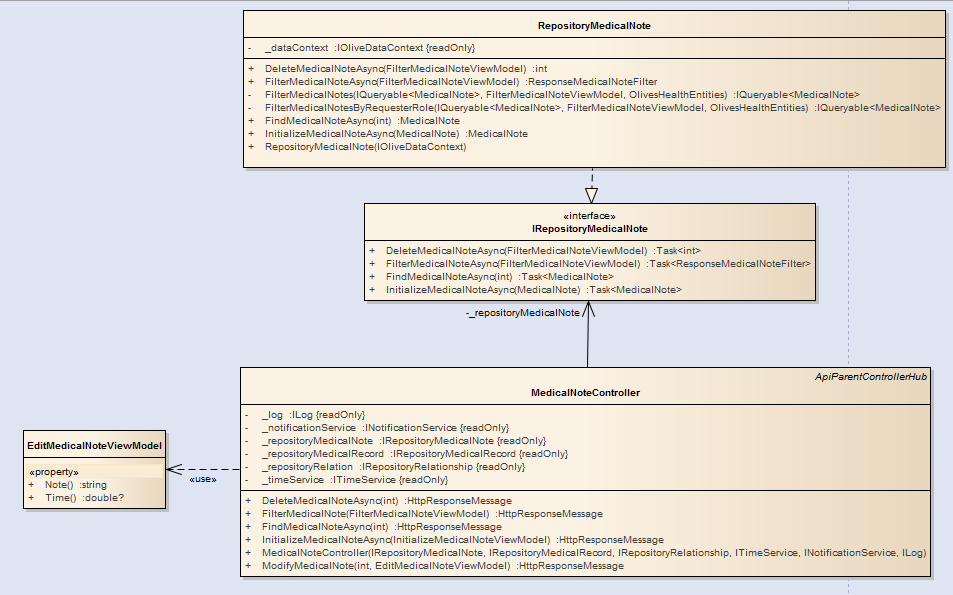
Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Id | int? | Index of experiment note. |
| 4 | MedicalRecord | int | Index of medical record. |
| 2 | Requester | Person | Requester who sent filter request. |
| 4 | Name | string | Name of experiment note. |
| 8 | MinCreated | double? | Time after which medical record was created. |
| 9 | MaxCreated | double? | Time before which medical record had been created. |
| 8 | MinLastModified | double? | Time after which medical record was lastly modified. |
| 9 | MaxLastModified | double? | Time before which medical record had been modified. |
| 10 | Page | int | Index of result page. |
| 11 | Records | int | Records number per page. |

##### 6.2.1.19. Edit medical record

* The data flow is: View -> <name>Controller -> IRepository<name> -> Repository<Name>.
* Name is one of the following: **MedicalRecord**, **MedicalImage**, **MedicalNote**, **ExperimentNote**.
* They all have same diagrams and sequence diagrams.

1. Class diagram



1. Class definition

EditExperimentNote class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Name | int? | Index of experiment note. |
| 2 | Time | double? | Time which note is about. |
| 3 | Infos | Dictionary<string, double> | Information of experiments. |

EditMedicalNoteViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Note | string | Note of medical. |
| 2 | Time | double? | Time which note is about. |

EditMedicalRecordViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Category | string | Note of medical. |
| 2 | Name | string | Name of medical record. |
| 3 | Infos | Dictionary<string, string> | Information of medical state. |
| 4 | Time | double? | Time which note is about. |

##### 6.2.1.20. Add medical record

InitializeMedicalImageViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | MedicalRecord | int | Index of medical record which contains image. |
| 2 | File | HttpFileModel | Image file. |

InitializeMedicalNoteViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | MedicalRecord | int | Index of medical record which contains image. |
| 2 | Note | string | Note of medical. |
| 3 | Time | double | Time when note is about. |

InitializeMedicalRecordViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Owner | int | Medical record owner. |
| 2 | Creator | string | Creator of medical record. |
| 3 | Name | string | Name of medical record. |
| 4 | Category | int | Category which medical record belongs to. |
| 5 | Infos | Dictionary<string, string> | Time when note is about. |
| 6 | Time | double | Time when note is about. |

##### 6.2.1.21. Delete medical record

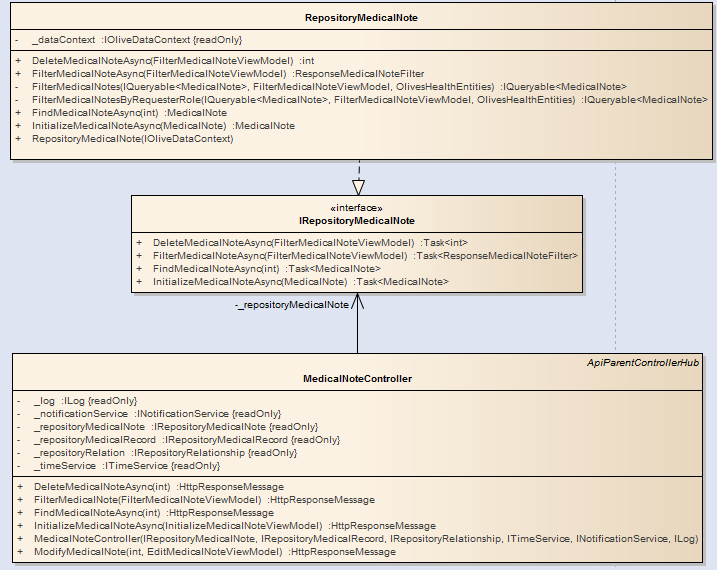


Diagram of Medical note delete.

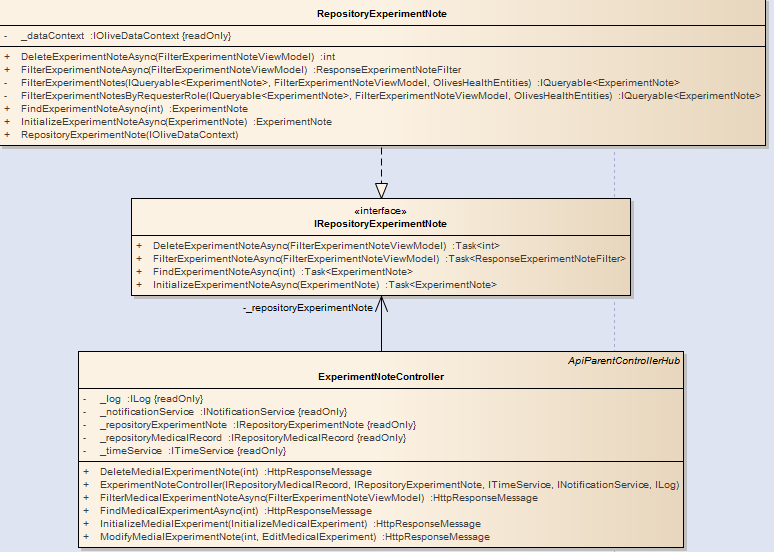
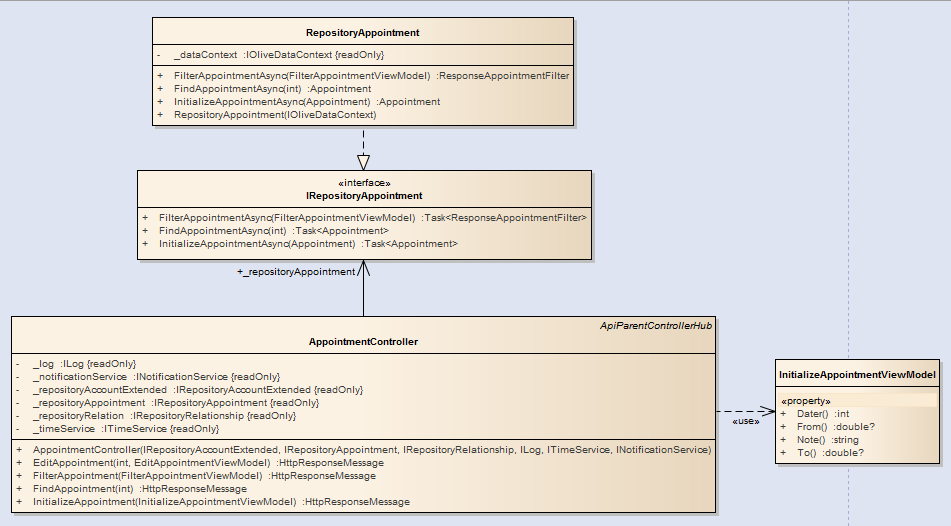


Diagram of Experiment note delete

##### 6.2.1.22. Create appointment

1. Class diagram



1. Class definition

AppointmentController class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_repositoryAccountExtended | IRepositoryAccountExtended | Instance which is used for accessing function to IRepositoryAccountExtended. |
| 2 | \_repositoryAppointment | IRepositoryAppointment | Instance which is used for accessing function to IRepositoryAppointment. |
| 3 | \_notificationService | INotificationService | Instance which is used for accessing function to INotificationService. |
| 4 | \_timeService | ITimeService | Instance which is used for accessing function to ITimeService. |
| 5 | \_log | ILog | Instance which is used for accessing function to ILog. |

Method: FindAppointmentAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Find appointment by using id and throw result back to client-side. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Data which will be sent to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of appointment. |

Method: InitializeAppointmentAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Initialize an appointment into database throw result back to client-side. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Data which will be sent to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | initializer | InitializeAppointmentViewModel | Information of appointment. |

Method: EditAppointmentAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Modify an appointment in database throw result back to client-side. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Data which will be sent to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of appointment which needs changing. |
| 2 | info | EditAppointmentViewModel | Information of appointment which will be used for being updated. |

Method: FilterAppointmentAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Filter appointments from database throw result back to client-side. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Data which will be sent to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | FilterAppointmentViewModel | Conditions which are used for filtering appointments from database. |

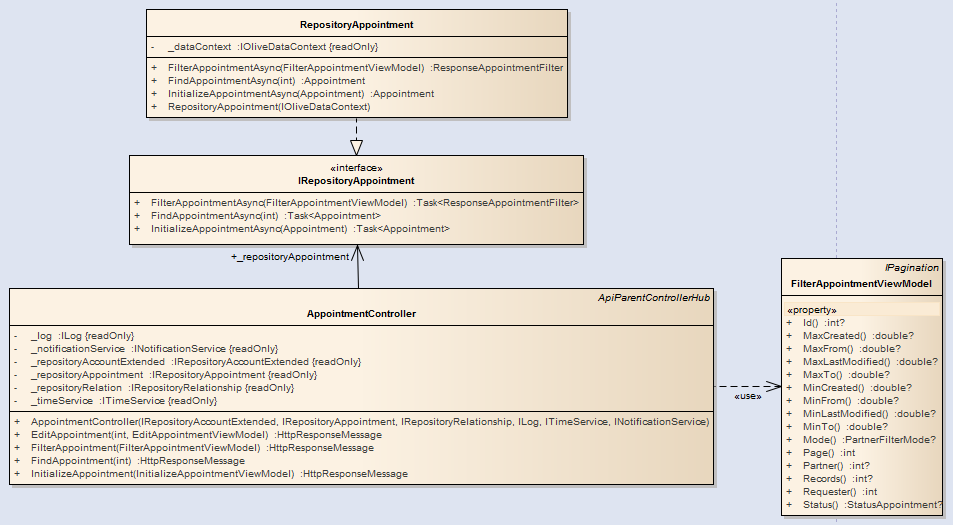
AppointmentController class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Dater | int | Person who is the target of appointment. |
| 2 | From | double? | Time when the appointment should be started. |
| 3 | To | double? | Time when the appointment should be ended. |
| 4 | Note | string | Note of appointment. |

##### 6.2.1.23. View appointment

1. Class diagram



1. Class definition

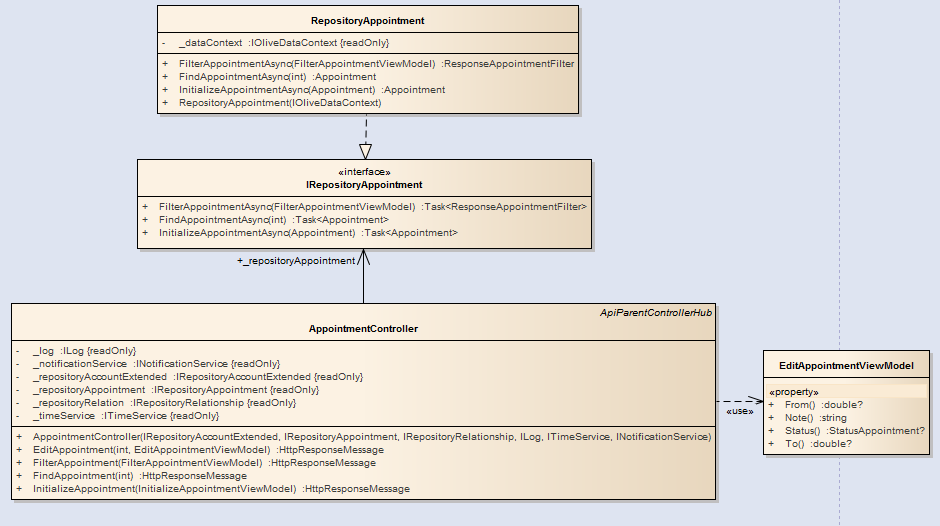
AppointmentController class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Mode | int | Person who is the target of appointment. |
| 2 | Partner | int? | Time when the appointment should be started. |
| 3 | Requester | Person | Time when the appointment should be ended. |
| 4 | Id | int | Id of appointment. |
| 5 | MinCreated | double? | Time after which appointment was created. |
| 6 | MaxCreated | double? | Time before which appointment had been created. |
| 7 | MinFrom | double? | Time after which appointment should be started. |
| 8 | MaxFrom | double? | Time after which appointment should be started. |
| 9 | MinTo | double? | Time after which appointment should be expired. |
| 10 | MaxTo | double? | Time before which appointment should be expired. |
| 11 | Status | StatusAppointment? | Status of appointment. |
| 12 | MinLastModified | double? | Time after which appointment was lastly modified. |
| 13 | MaxLastModified | double? | Time before which appointment had been modified. |
| 14 | Page | int | Index of result page. |
| 15 | Records | int? | Number of records per page. |

##### 6.2.1.24. Accept appointment

1. Class diagram



1. Class definition

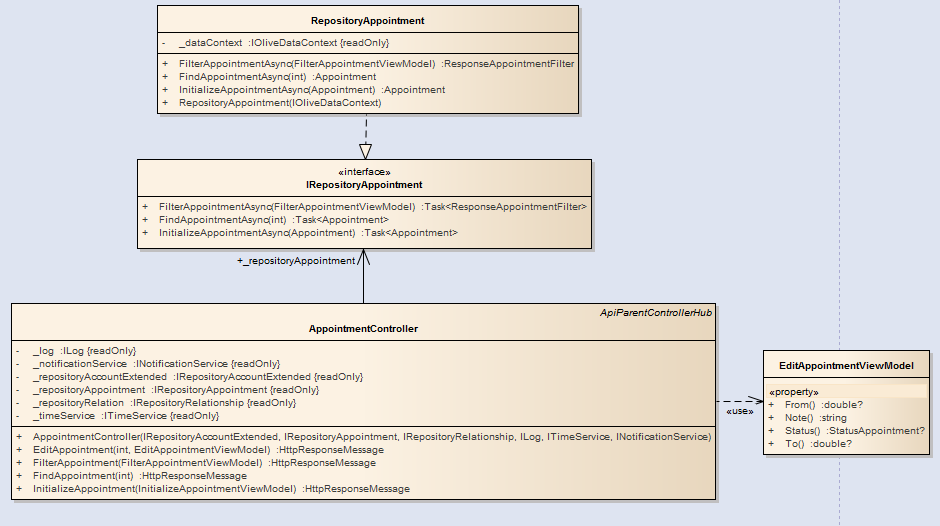
EditAppointmentViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | From | double? | Person who is the target of appointment. |
| 2 | Note | string | Time when the appointment should be started. |
| 3 | Status | StatusAppointment? | Status of appointment. |
| 4 | To | double? | Time after which appointment will be expired. |

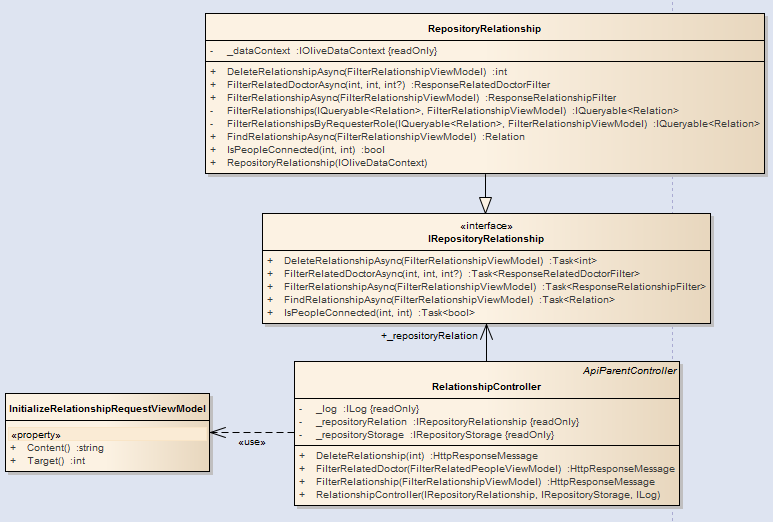
##### 6.2.1.25. Cancel appointment

1. Class diagram



##### 6.2.1.26. Connect/disconnect to/from doctor

1. Class diagram



1. Class definition

RepositoryRelationship class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_log | ILog | Instance which is used for accessing function to ILog. |
| 2 | \_repositoryRelation | IRepositoryRelationship | Instance which is used for accessing function to IRepositoryRelationship. |
| 3 | \_repositoryStorage | IRepositoryStorage | Instance which is used for accessing function to IRepositoryStorage. |

Method: DeleteRelationship()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Delete relationship by using conditions. | | | |
| Return | **Type** | | | **Description** |
| int | | | Number of records which have been deleted. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of appointment. |

Method: FilterRalatedDoctorAsync()

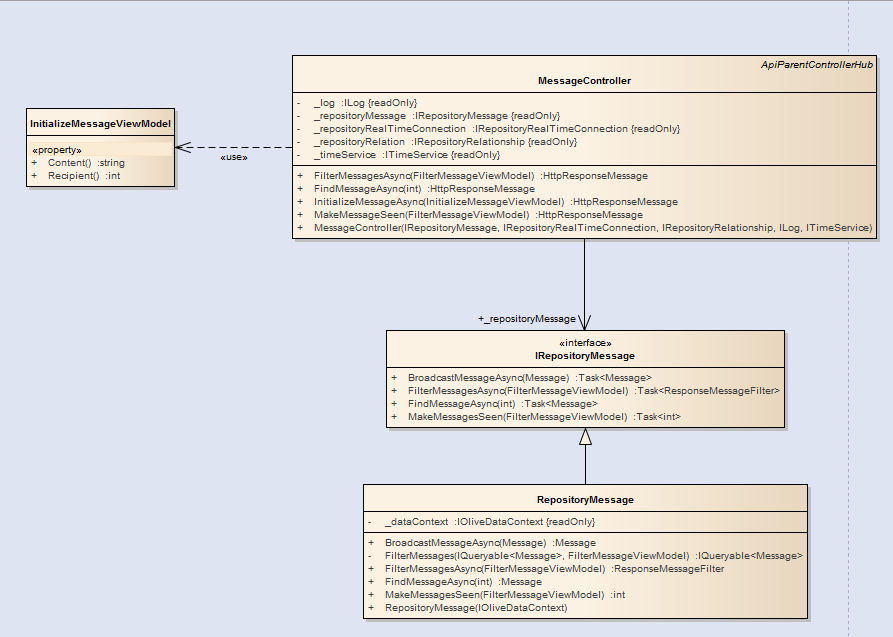
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Filter doctor who has relationship with the requester by using conditions. | | | |
| Return | **Type** | | | **Description** |
| ResponseRelatedDoctorFilter | | | Model which contains list of filtered result and number represents the total condition matched result. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | requester | int | Id of person who sent filter request. |
| 2 | page | int | Id of requester who sent filter request. |
| 3 | record | int | Id of requester who sent filter request. |

Method: IsPeopleConnected()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Check whether 2 people are connected or not. | | | |
| Return | **Type** | | | **Description** |
| bool | | | True if people are connected and vice versa. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | firstPerson | int | Id of first person. |
| 2 | secondPerson | int | Id of second person. |

##### 6.2.1.27. Chat

1. Class diagram



RepositoryRelationship class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_repositoryRelation | IRepositoryRelationship | Instance which is used for accessing function to IRepositoryRelationship. |
| 2 | \_repositoryRealTimeConnection | IRepositoryRealTimeConnection | Instance which is used for accessing function to IRepositoryRealTimeConnection. |
| 3 | \_repositoryMessage | IRepositoryMessage | Instance which is used for accessing function to IRepositoryMessage. |
| 4 | \_timeService | ITimeService | Instance which is used for accessing function to ITimeService. |
| 5 | \_log | ILog | Instance which is used for accessing function to ILog. |

Method: FindMessageAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Find message by using. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Instance of result which will be thrown back to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | id | int | Id of message. |

Method: InitializeMessageAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Initialize a message into database and broadcast it to recipient. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | . |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | initializer | InitializeMessageViewModel | Instance of message. |

Method: InitializeMessageAsync()

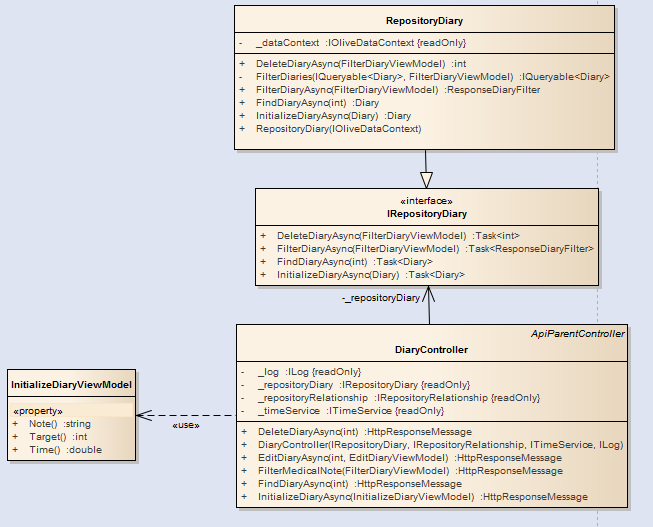
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Initialize a message into database and broadcast it to recipient. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | . |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | initializer | InitializeMessageViewModel | Instance of message. |

Method: MakeMessageSeen()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Mark the message as having been seen. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Instance contains result which will be thrown back to client. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | initializer | FilterMessageViewModel | Instance which contains condition for filtering messages. |

##### 6.2.1.28. View Diary

1. Class diagram



1. Class definition

RepositoryDiary class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | \_repositoryRelation | IRepositoryRelationship | Instance which is used for accessing function to IRepositoryRelationship. |
| 2 | \_repositoryRealTimeConnection | IRepositoryRealTimeConnection | Instance which is used for accessing function to IRepositoryRealTimeConnection. |
| 3 | \_repositoryMessage | IRepositoryMessage | Instance which is used for accessing function to IRepositoryMessage. |
| 4 | \_timeService | ITimeService | Instance which is used for accessing function to ITimeService. |

Method: DeleteDiaryAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Delete diary by using specific conditions asynchronously. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Instance contains response which will be thrown back to client-side. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | FilterDiaryViewModel | Instance which contains condition for filtering diaries. |

Method: FilterDiaryAsync()

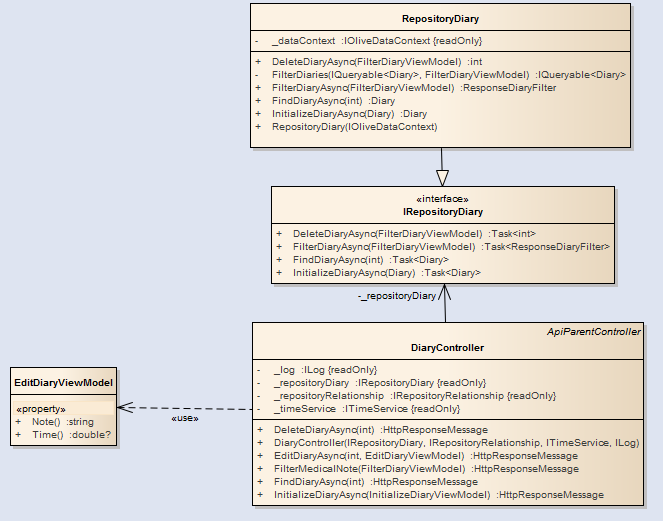
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Filter diaries by using specific conditions. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Instance contains response which will be thrown back to client-side. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | FilterDiaryViewModel | Instance which contains condition for filtering diaries. |

Method: InitializeDiaryAsync()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose | Filter diaries by using specific conditions. | | | |
| Return | **Type** | | | **Description** |
| HttpResponseMessage | | | Instance contains response which will be thrown back to client-side. |
| Parameter | **No** | **Name** | **Type** | **Description** |
| 1 | filter | InitializeDiaryViewModel | Model contains information of diary. |

##### 6.2.1.29. Edit Diary

1. Class diagram



1. Class definition

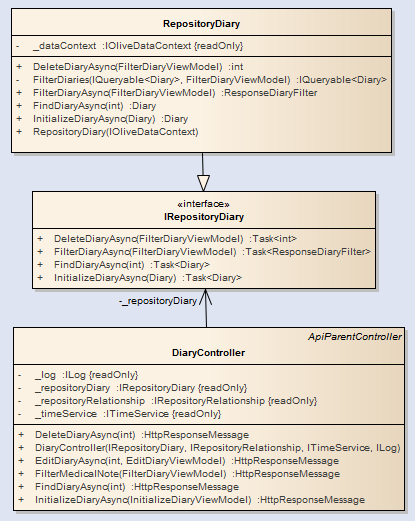
EditDiaryViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Time | double | Time of diary. |
| 2 | Note | Note | Note of diary. |

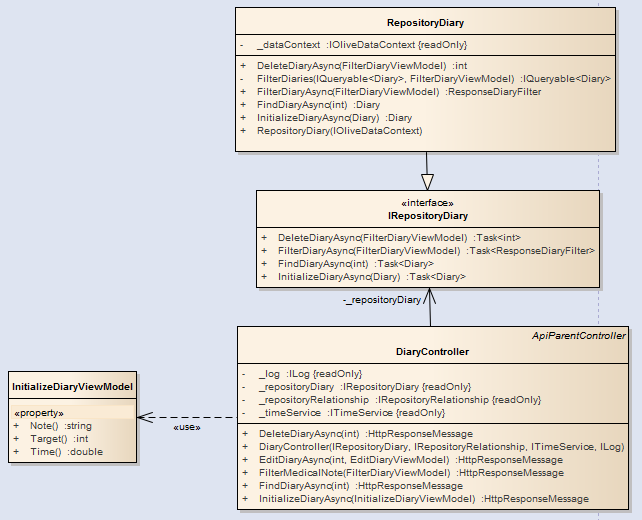
##### 6.2.1.30. Delete Diary

1. Class diagram



##### 6.2.1.30. Initialize a diary

1. Class diagram



1. Class definition

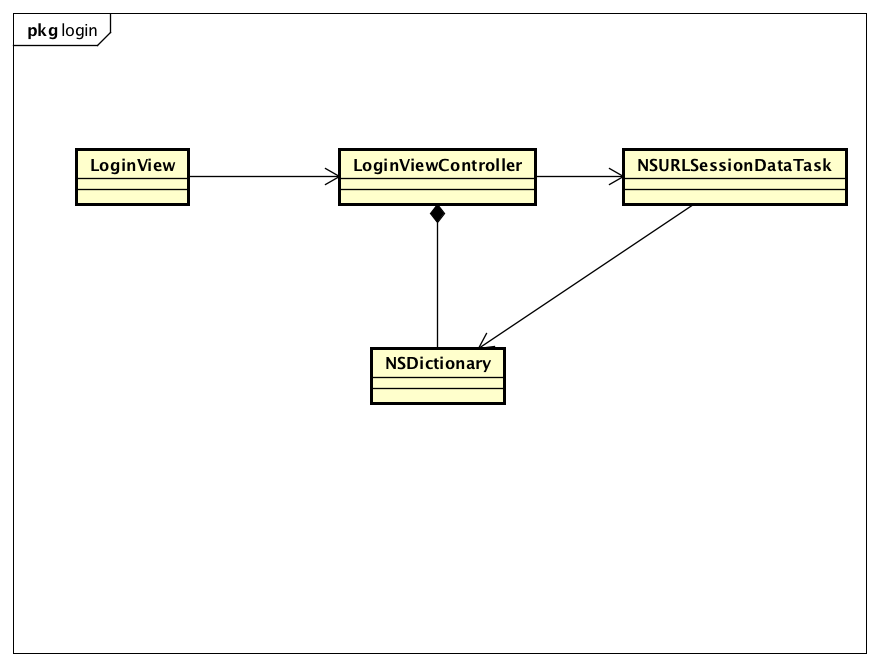
InitializeDiaryViewModel class

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Type | Description |
| 1 | Target | int | Id of patient who diary is about. |
| 2 | Time | double | Time of diary. |
| 3 | Note | Note | Note of diary. |

**6.2.4 Olives doctor app class diagram**

6.2.4.1.Login class diagram



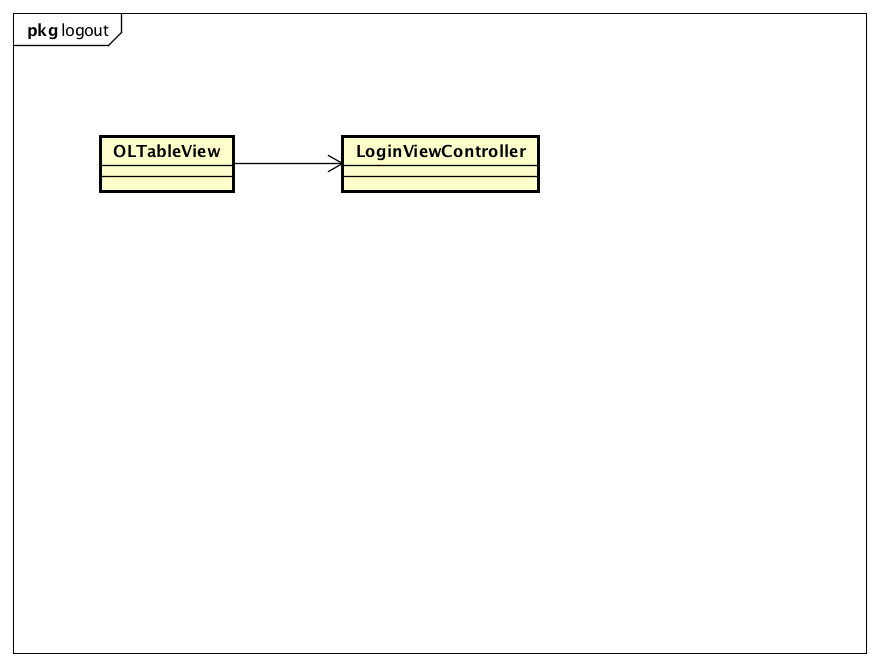
Attribute

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | loginButton | UIButton | Content outlet button value |
| 2 | forgotPasswordButton | UIButton | Content outlet button value |
| 3 | emailTextField | UITextField | Content input email from user |
| 4 | passwordTextField | UITextField | Content input password from user |
| 5 | responseJSONData | NSDictionary | Content return value from server |

Method

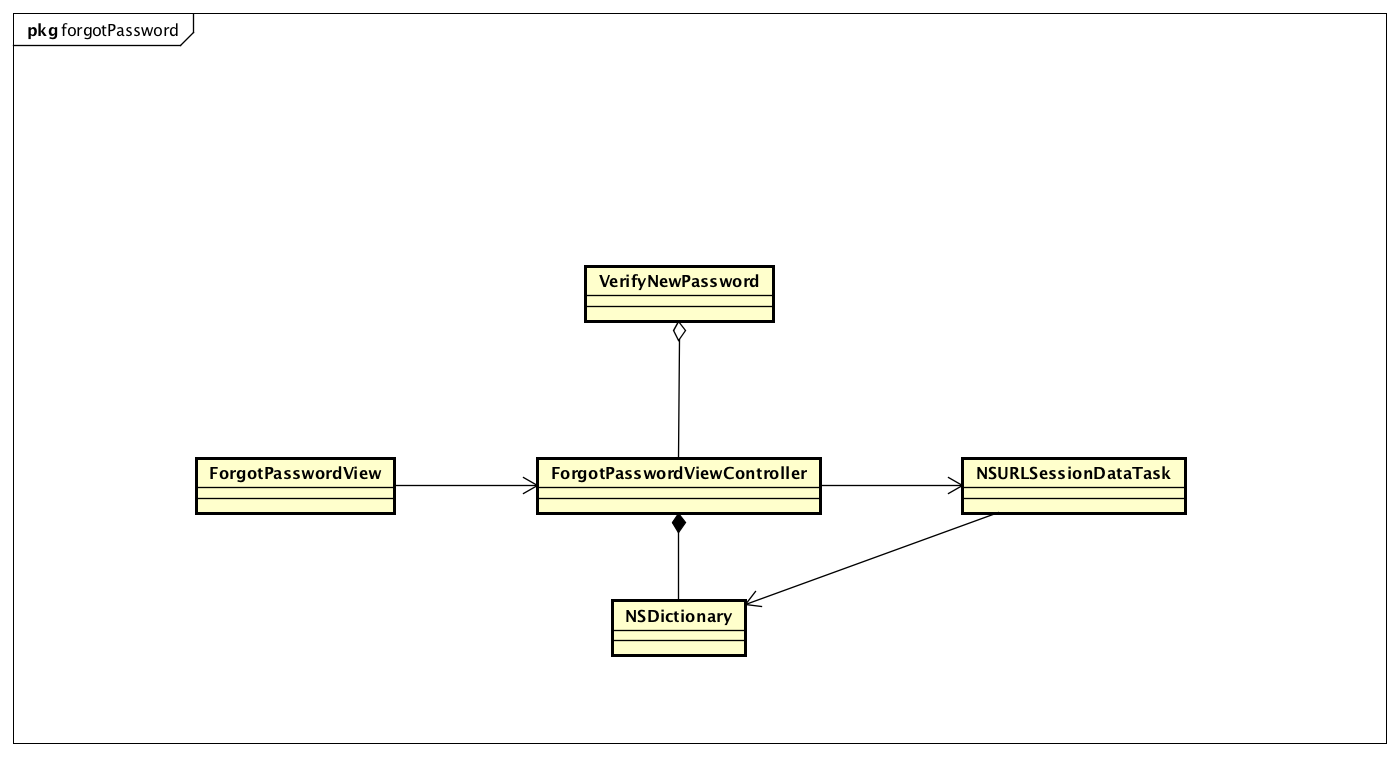
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | loginAPI |  | Void | Post request to server and handle return response from API |
|  |  |  |  |  |

6.2.4.2.logout class diagram



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | logout |  | Void | Move user to login screen |

6.2.4.3.Forgot password



Attribute of ForgotPasswordViewController class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | emailTextField | UITextField | Content input email from user |
| 2 | confirmEmailButton | UIButton | Content outlet of confirm button |

Method of ForgotPasswordViewController class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | confirmEmailButton | Sender | (IBAction) | Receive input from user and request to server |

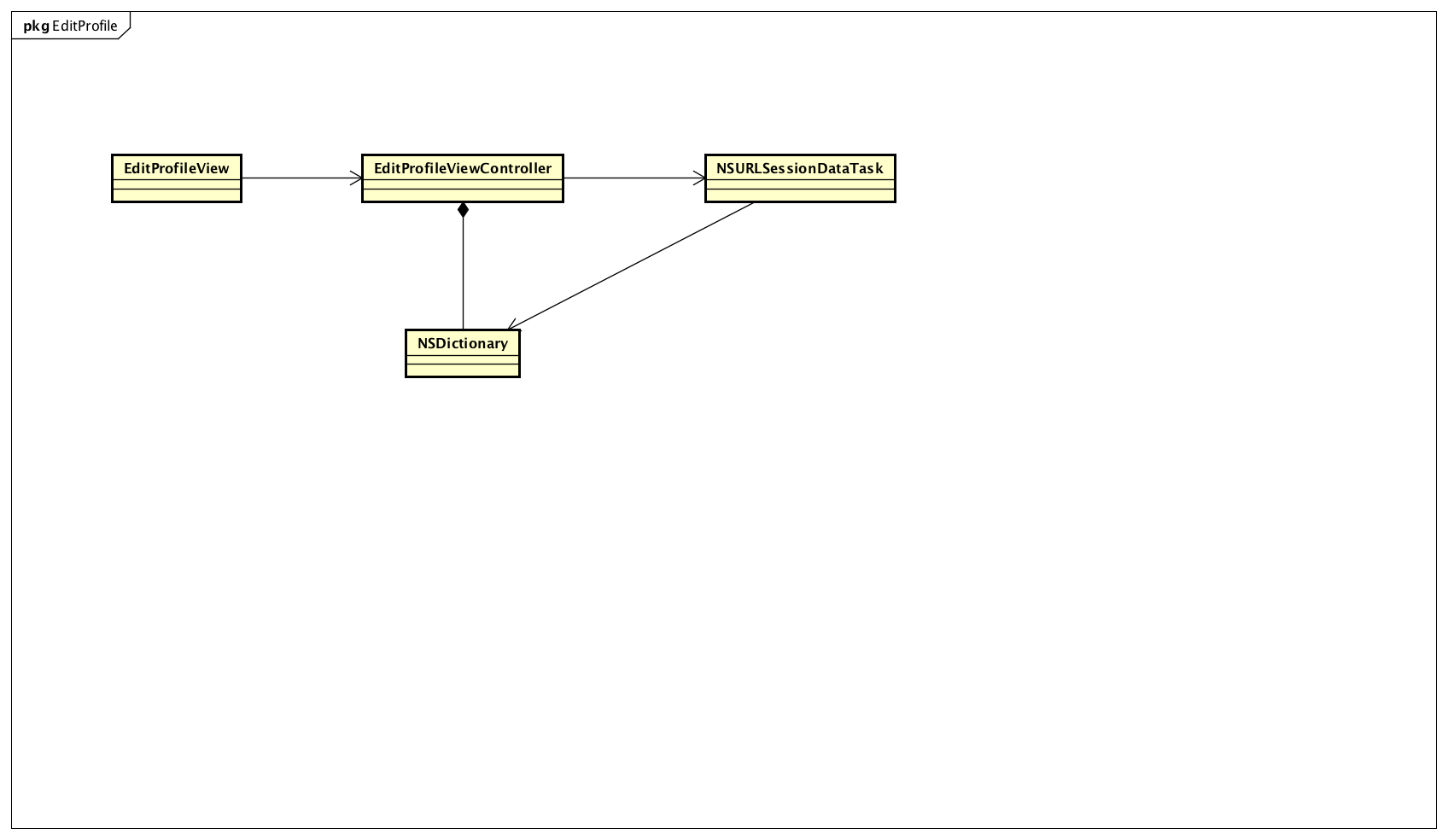
Attribute of VerifyNewPassword class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | verifyCodeTextField | UITextField | Content verify code from user |
| 2 | oldPasswordTextField | UITextField | Content old password from user |
| 3 | newPasswordTestField | UITextField | Content new password from user |
| 4 | changePasswordButton | UIButton | Content outlet of change password button |

Method of VerifyNewPassword class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | changePasswordButton | Sender | (IBAction) | Post request change password to server |

6.2.4.4.Edit profile



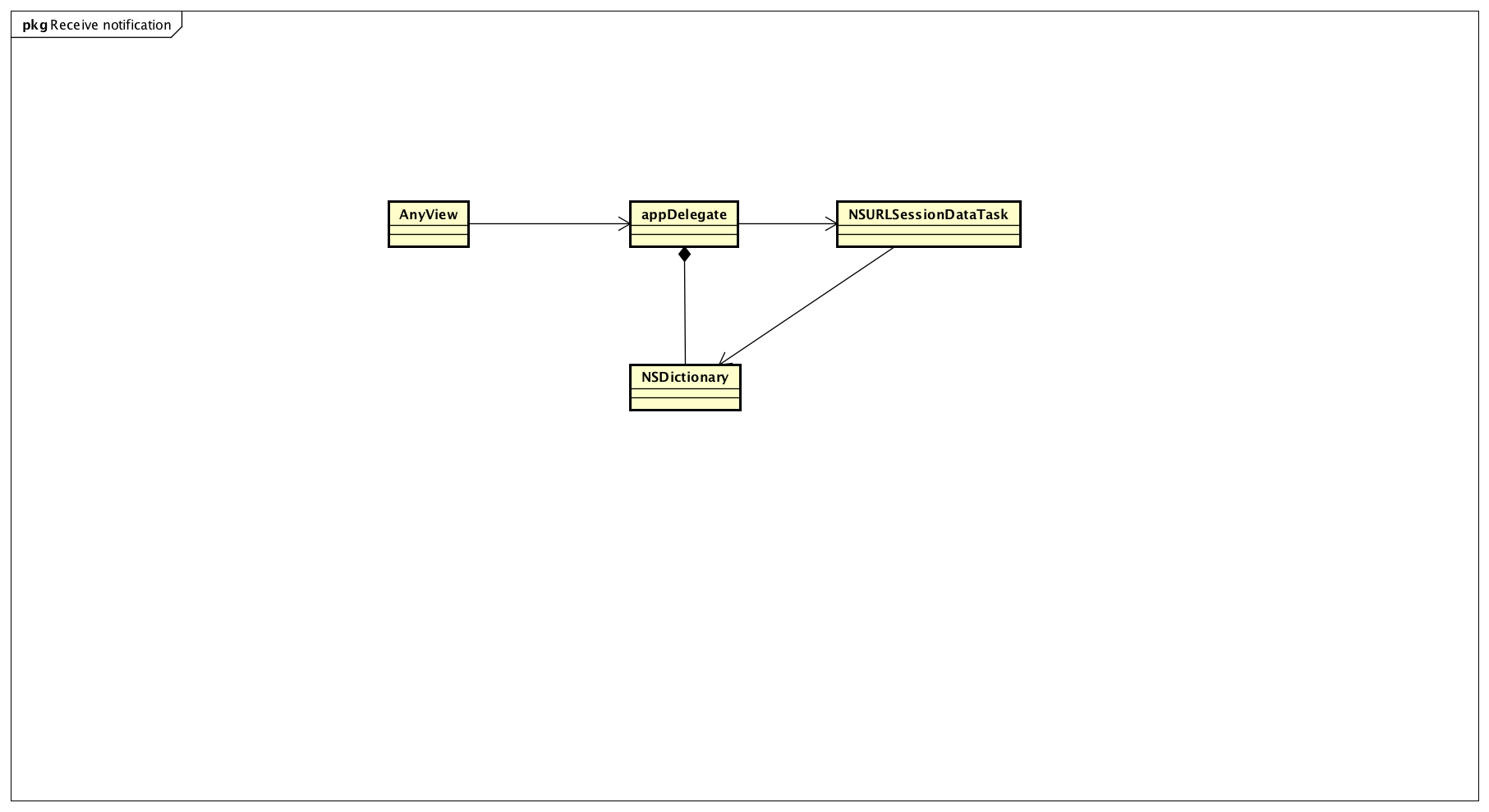
Attribute of EditProfile class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | saveButton | UIButton | Content outlet of button |
| 2 | birthdayDatetime | UIDatePicker | Content birthday from user |
| 3 | firstNameTextField | UITextField | Content first name of user |
| 4 | lastNameTextField | UITextField | Content last name of user |
| 5 | phoneTextField | UITextField | Content phone of user |
| 6 | addressTestField | UITextField | Content address of user |
| 7 | genderSegment | UISegmentControl | Content gender info of user |
| 8 | avatar | UIImageView | Content of image of user |
|  |  |  |  |

Method of EditProfile class

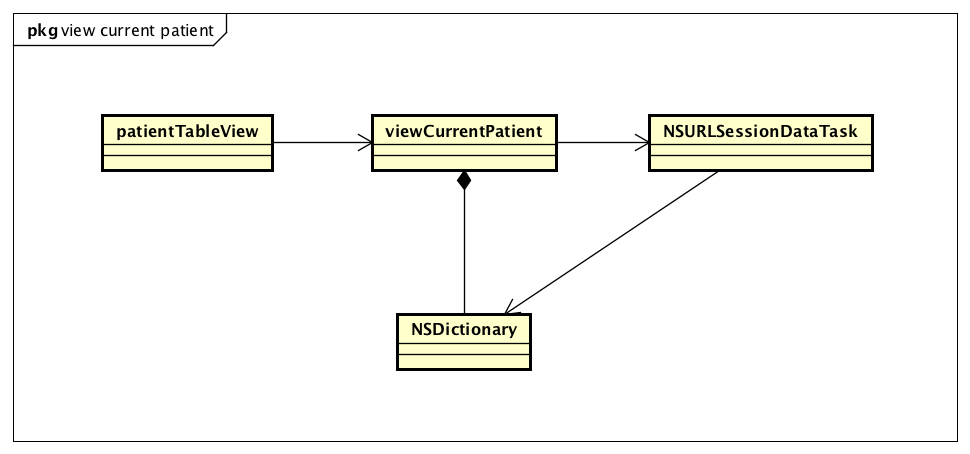
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | updateProfile | Sender | (IBAction) | Update patient profile |

6.2.4.5.Receive Notification



Method of appDelegate class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | Application:didFinishLaunchingWithOption | NSDictionary | BOOL | Configure app during force ground time |

6.2.4.6.View current patient

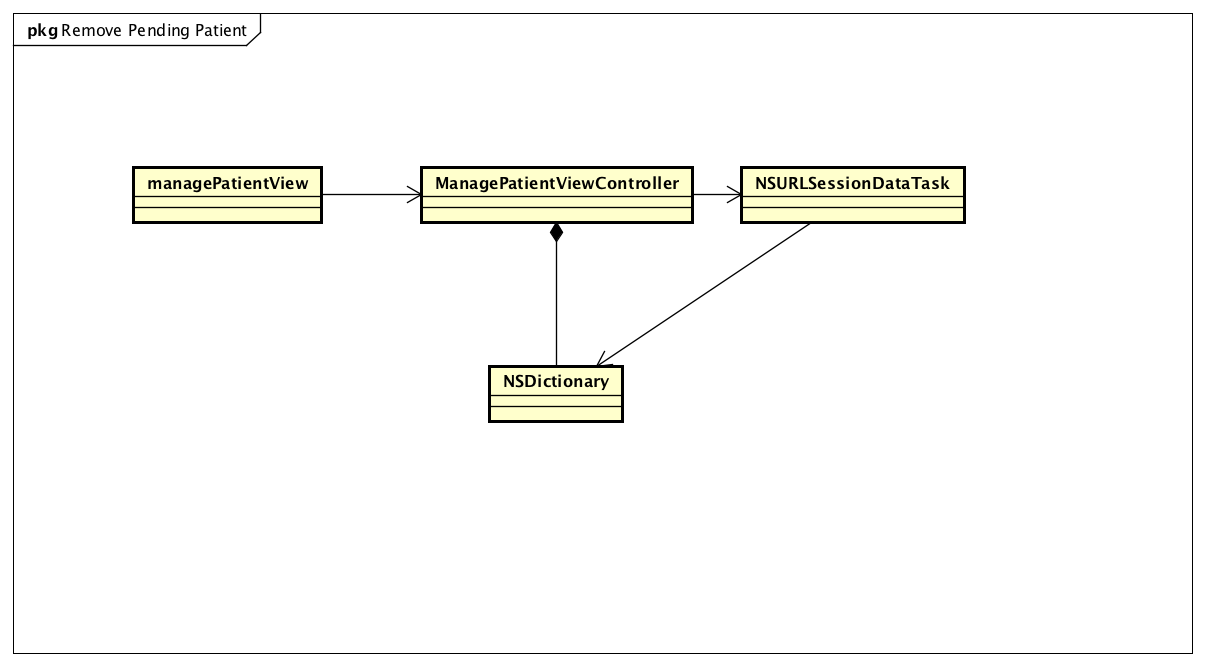
Attribute of viewCurrentPatient class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | responseJSONData | NSDictionary | Content return value from server |
| 2 | patientArray | NSArray | Content array of patient |

Method of viewCurrentPatient class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | loadPatientDataFromAPI |  | Void | Send request to API and handle response data. |

6.2.4.7.Manage patient <remove,accept,cancel>



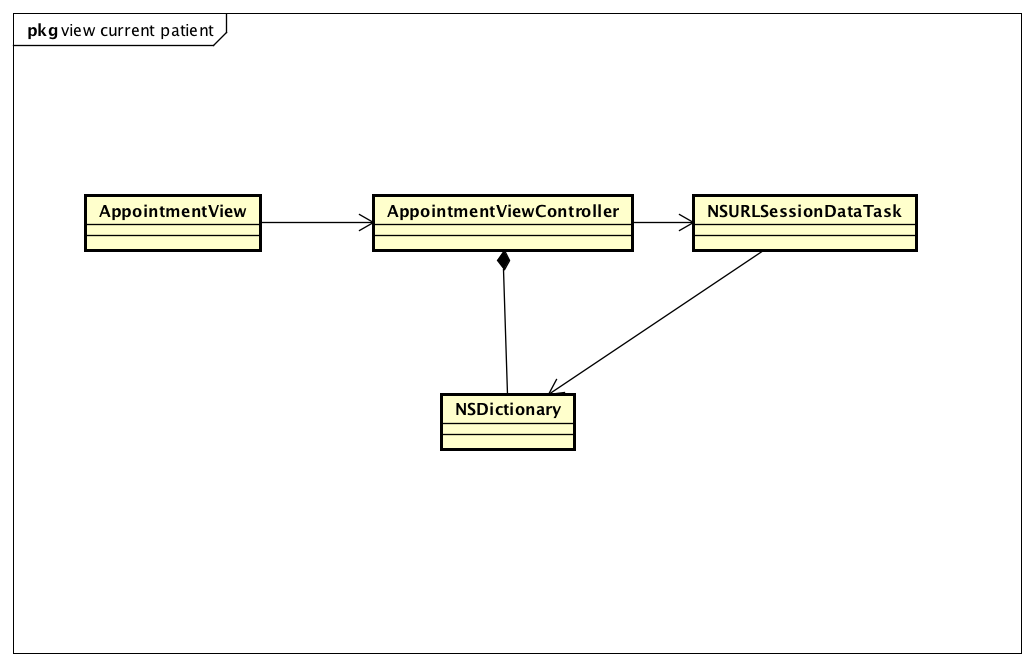
Attribute of ManagePatientViewController class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | responseJSONData | NSDictionary | Content return value from server |
| 2 | segmentController | UISegmentController | Change status of pending position |
| 3 |  |  |  |

Method of ManagePatientViewController class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | removePatientAPIWithID | patientID:NSString | Void | Remove patient in relationship |
| 2 | cancelRequestAPIWithID | patientID:NSString | Void | Cancel patient request |
| 3 | acceptRequestAPIWithID | patientID:NSString | Void | Accept request from patient |
| 4 | loadCurrentPatientDataFromAPi |  | Void | Receive current patients in relationship |
| 5 | loadPendingPatientDataFromAPI |  | Void | Receive pending patient in relationship |

6.2.4.8.View current appointments



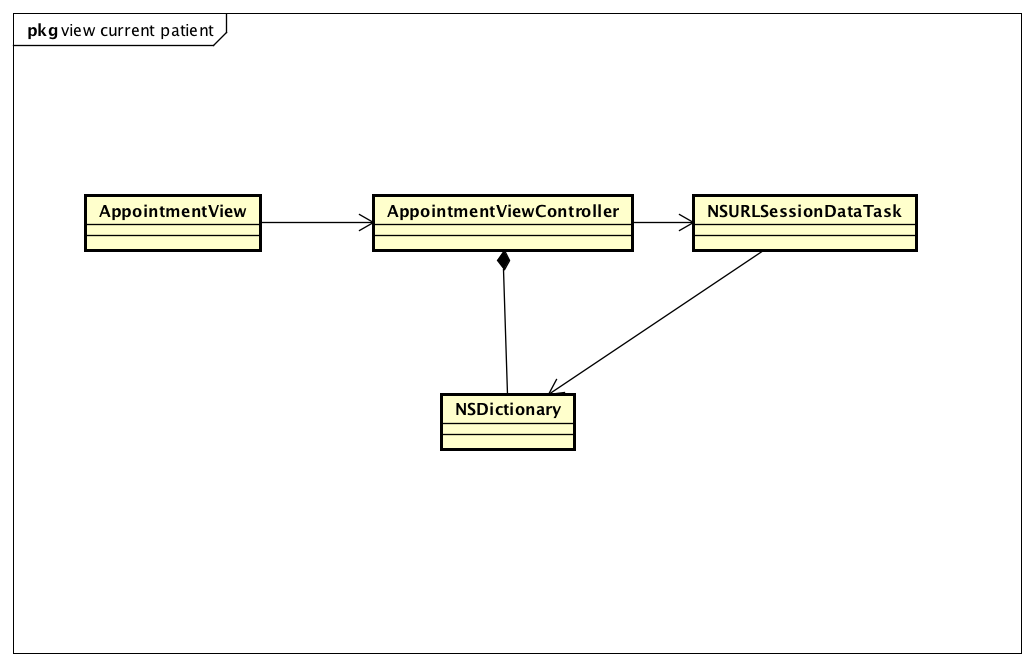
Attribute of AppointmentViewController class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | responseJSONData | NSDictionary | Content current appointments return from server |
| 2 | calendarView | VRGCalendarView | Display calendar |

Method of ManagePatientViewController class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | loadAppointmentDataFromAPIFrom: and : | minDate:NSString  maxDate:NSString | NSDictionary | Load current appointments data for each month |
| 2 | viewDidLoad |  | Void | Setting view when finish loading |
| 3 | viewDidApear |  | Void | Setting view whenever show up again |
| 4 | calendarView: switchedToMonth: targetHeight:animated | calendarView:VGRCAlendar  month:int  targetHeight:float  animated:BOOL | Void | Handle user action when switched to month |
| 5 | calendarView:dateSelected | calendarView:VGRCAlendar  date:NSdate | Void | Handle user action when selected date |

6.2.4.9.View pending appointments



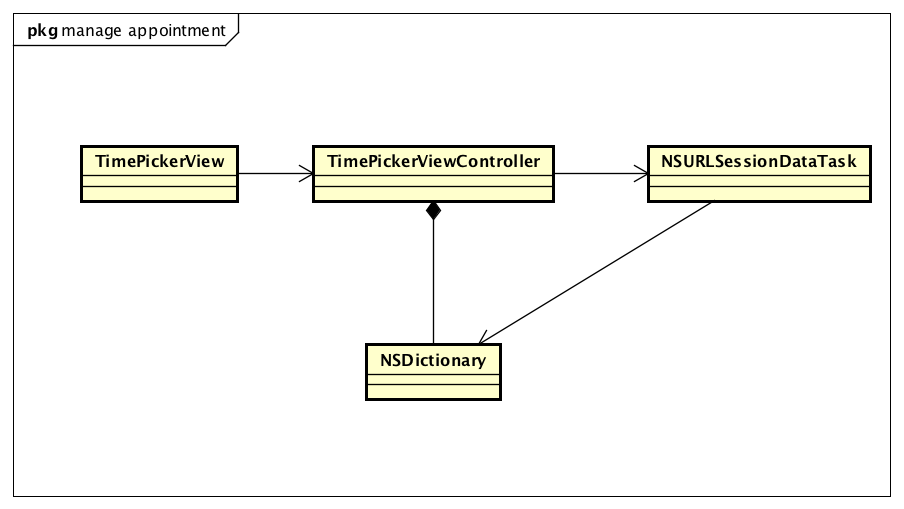
Attribute of AppointmentViewController class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | responseJSONDataForPendingList | NSDictionary | Content pending appointments return from server |
| 2 | calendarView | VRGCalendarView | Display calendar |

Method of ManagePatientViewController class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | loadAppointmentDataWithStatus: | Status:NSinteger | NSDictionary | Load pending appointments data |
| 2 | viewDidLoad |  | Void | Setting view when finish loading |
| 3 | viewDidApear |  | Void | Setting view whenever show up again |
| 4 | calendarView: switchedToMonth: targetHeight:animated | calendarView:VGRCAlendar  month:int  targetHeight:float  animated:BOOL | Void | Handle user action when switched to month |
| 5 | calendarView:dateSelected | calendarView:VGRCAlendar  date:NSdate | Void | Handle user action when selected date |

6.2.4.10.Manage appointment <cancel/accept/edit>



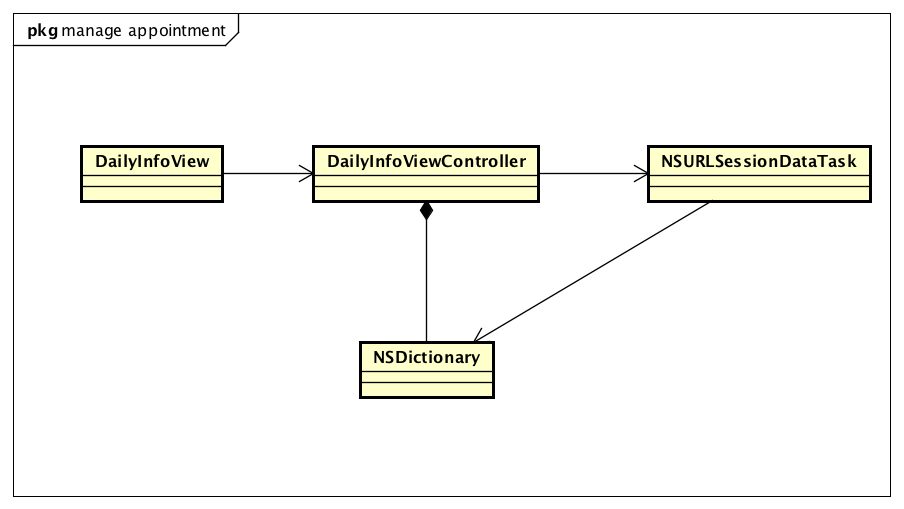
Attribute of TimePickerViewController class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | dateTimePickerFrom | UIDatePicker | Content from time |
| 2 | dateTimePickerTo | UIDatePicker | Content to time |
| 3 | noteLabel | UITextView | Content user note |
| 4 | sendButton | UIButton | Content outlet of send button |
| 5 | choosingPatient | UIView | Choosing patient view |
| 6 | cancelButton | UIButton | Content outlet of cancel button |
| 7 | editOrAcceptButton | UIButton | Edit or accept appointment button |
| 8 | makerNoteTextView | UITextView | Initial note from appointment maker |

Method of TimePickerViewController class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | sendRequestChangeAppointmentStatusToAPI: | Status:NSinteger | NSDictionary | Change status appointment |
| 2 | viewDidLoad |  | Void | Setting view when finish loading |
| 3 | viewDidApear |  | Void | Setting view whenever show up again |
| 4 | sendRequestEditAppointmentToAPI |  | Void | Edit appoinment |
| 5 | sendRequestCreateAppointmentToAPI |  | Void | Create appointment |

6.2.4.11.View Patient daily record

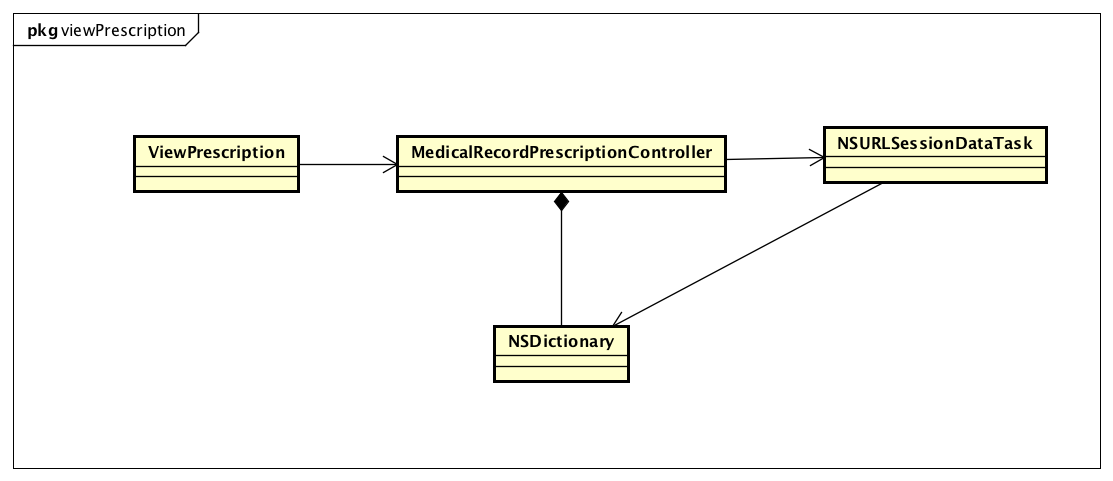


Attribute of DailyInfoViewController class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | responseJSONData | NSDictionary | Content daily statistic info return from server |
| 2 | lineChartView | LineChartView | Display patient daily statistic info in chart |
| 3 | tableView | UITableView | Display patient daily statistic info in data |

Method of DailyInfoViewController class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | loadDataFromAPIFromDate:toDate: | minTime:double  maxTime:double | Void | Load daily info from server in specific interval time |
| 2 | configModeForDisplay | totalDays:NSInteger | Void | Config data to display in choosen mode |
| 3 | displayDataToChartViewWithTotalsDays: fromStartDate: | total:NSinteger  startDate:NSDate | Void | Display data to chart view |

6.2.4.12.View prescription

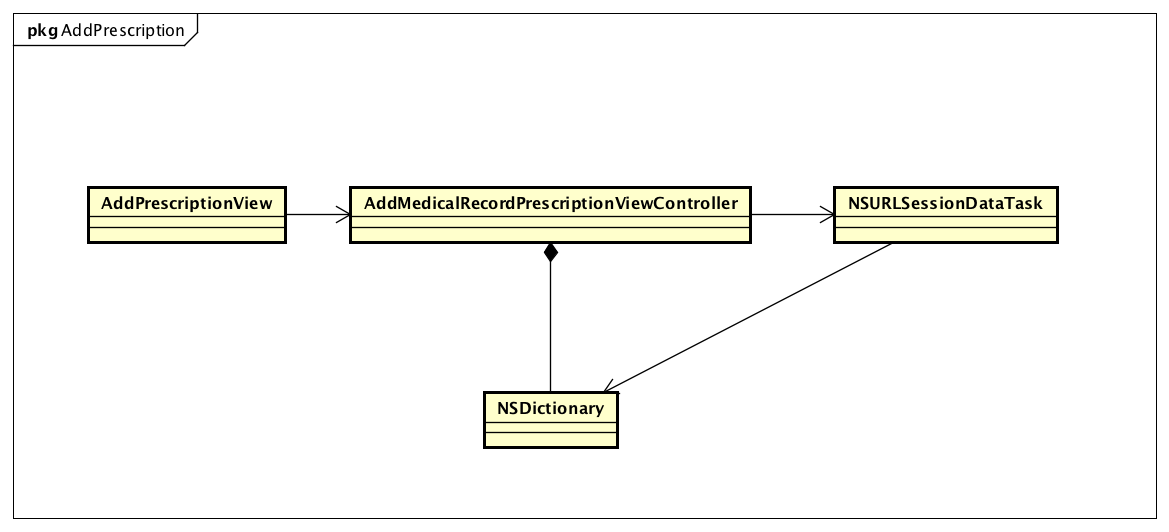
Attribute of MedicalRecordPrescription class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | responseJSONData | NSDictionary | Content return data from server |
| 2 | prescriptionArray | NSArray | Content prescription in array |

Method of MedicalRecordPrescription class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | loadPrescriptionDataFromAPI: |  | Void | Load prescription data from API |

6.2.4.13.Manage Prescription <add/update>



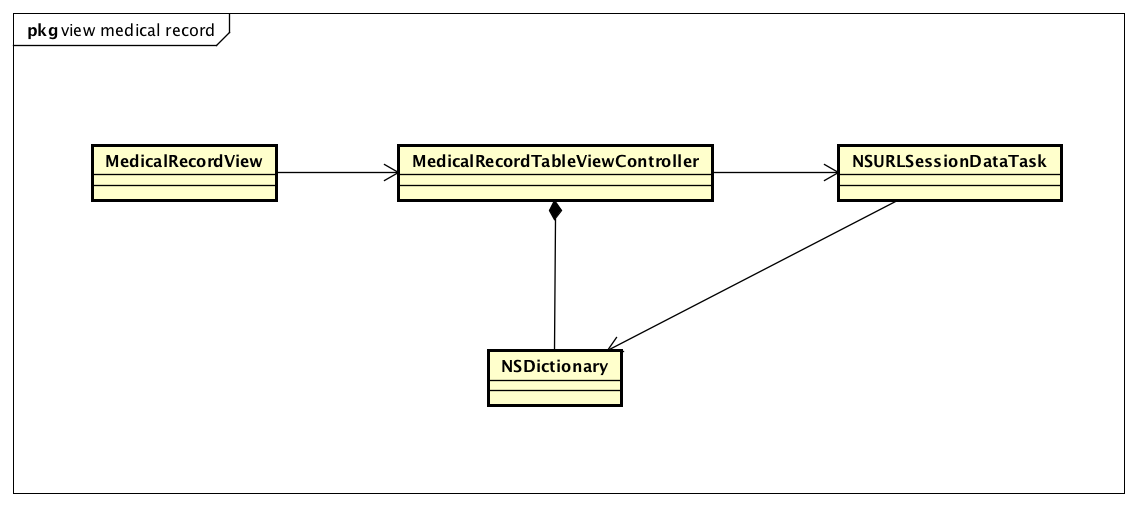
Attribute of AddMedicalRecordPrescriptionViewController class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | responseJSONData | NSDictionary | Content return data from server |
| 2 | prescriptionName | UITextView | Content prescription name |
| 3 | noteTextField | UITextView | Content prescription note |
| 4 | fromDateTimePicker | UIDateTimePicker | Content from time |
| 5 | toDateTimePicker | UIDateTimePicker | Content to time |

Method of AddMedicalRecordPrescriptionViewController class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | addNewPrescriptionToAPI: |  | Void | Add prescription to server |
| 2 | editPrescriptionToAPI |  | Void | Edit prescription to server |

6.2.4.14.View medical record



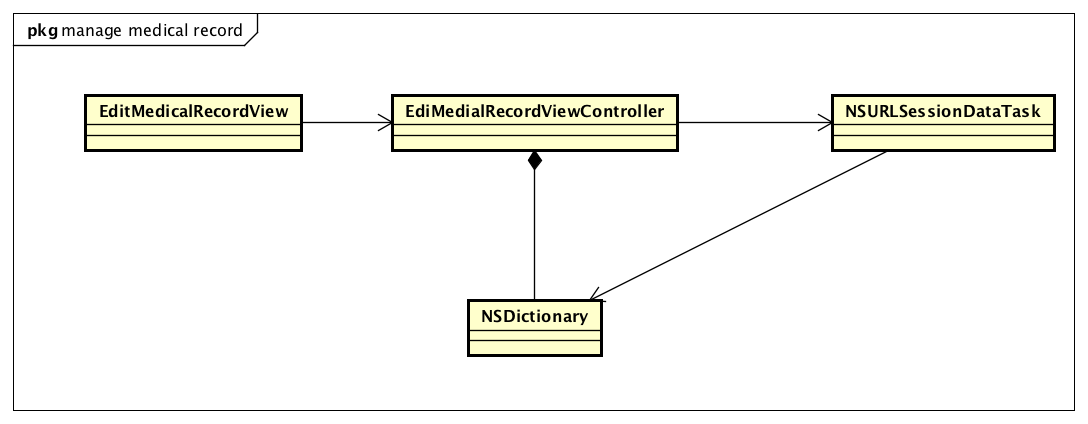
Attribute of MedicalRecordTableViewController class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | responseJSONData | NSDictionary | Content return data from server |
| 2 | medicalCategoryArray | NSArray | Content category in array |
| 3 | medicalRecordArray | NSArray | Content medical record in array |

Method of MedicalRecordTableViewController class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | loadMedicalRecordDataFromAPI |  | Void | Load medical record data from server |

6.2.4.15.Edit medical record



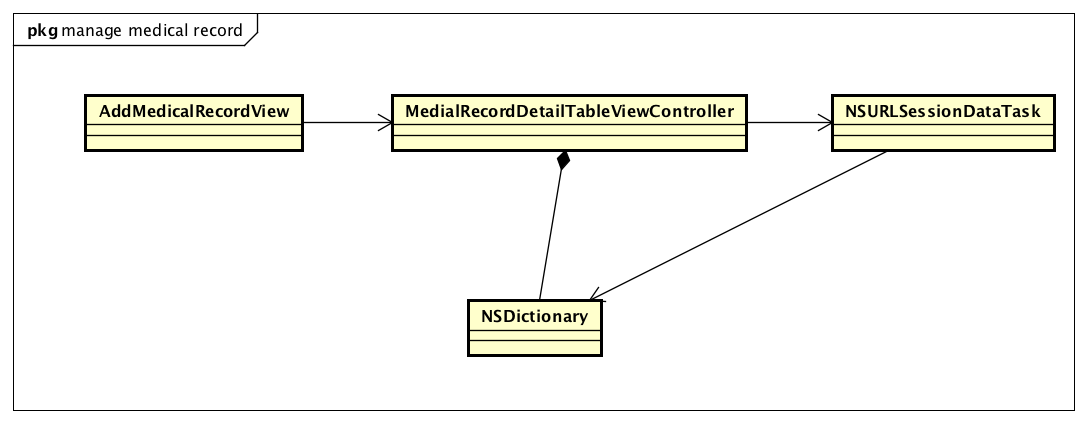
Attribute of EditMedicalRecordViewController class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | responseJSONData | NSDictionary | Content return data from server |
| 2 | nameTextField | UITextField | Content name of medical record |
| 3 | dateTimePicker | UIDateTimePicker | Content time of medical record |
| 4 | saveButton | UIButton | Content outlet of save button |

Method of EditMedicalRecordViewController class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | editMedicalRecordAPI |  | Void | Send request edit medical record to server |
|  |  |  |  |  |

6.2.4.16.Create medical record



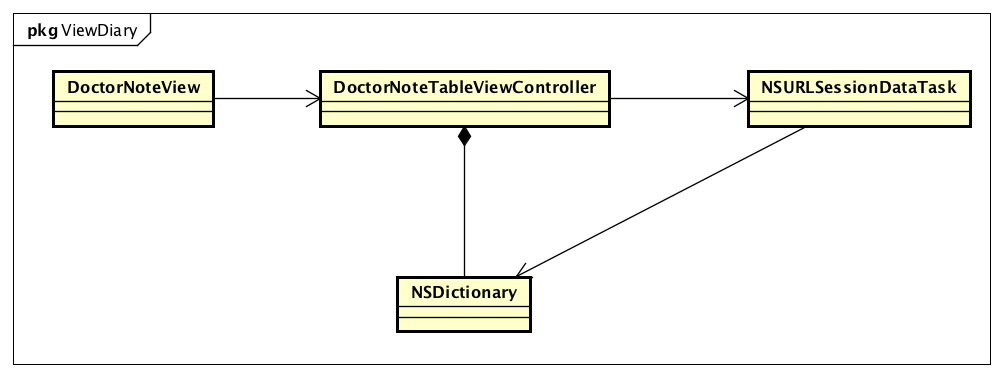
Attribute of MedicalRecordDetailTableViewController class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | responseJSONData | NSDictionary | Content return data from server |
| 2 | nameTextField | UITextField | Content name of medical record |
| 3 | dateTimePicker | UIDateTimePicker | Content time of medical record |
| 4 | saveButton | UIButton | Content outlet of save button |

Method of MedicalRecordDetailTableViewController class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | createMedicalRecordDataToAPI |  | Void | Send request create medical record to server |
|  |  |  |  |  |

6.2.4.17.View diary



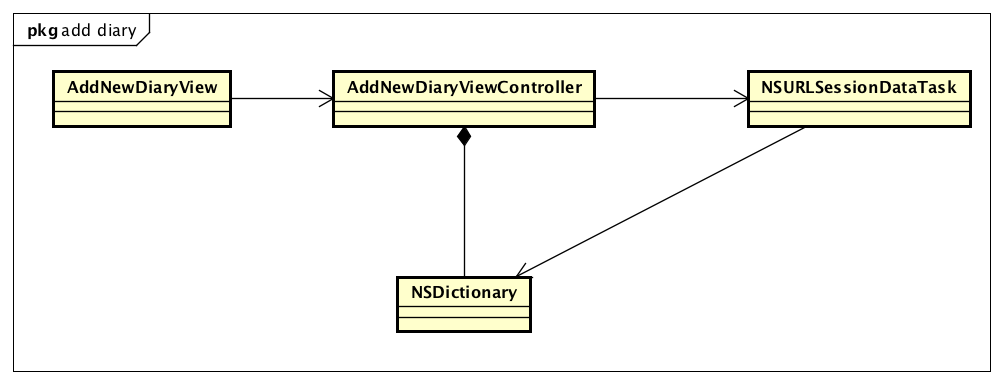
Attribute of DoctorNoteTableViewController class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | responseJSONData | NSDictionary | Content return data from server |
| 2 | diaryArray | NSArray | Content array of diary |

Method of DoctorNoteTableViewController class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | loadDiaryFromAPI |  | Void | Return data from server |

6.2.4.18.Add diary



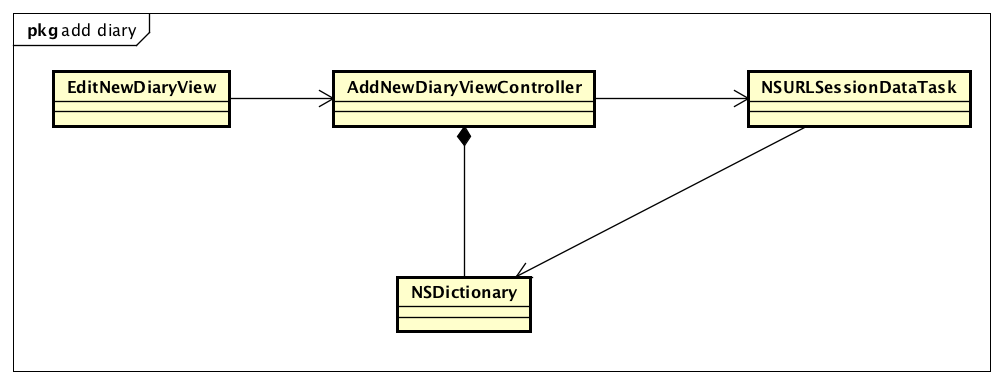
Attribute of AddNewDiaryViewController class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | timePicker | UIDateTimePicker | Content time of diary |
| 2 | noteTextView | UITextView | Content doctor note |

Method of AddNewDiaryViewController class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | addNewDiaryNoteAPI |  | Void | Add new diary to server |

6.2.4.19.Edit diary



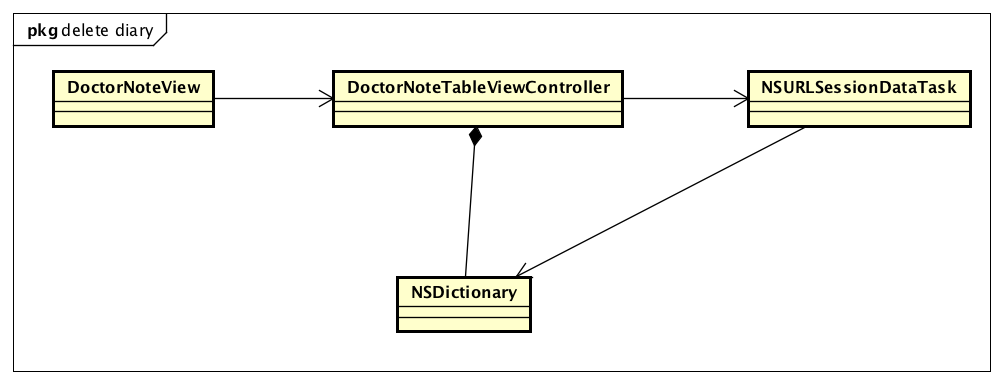
Attribute of AddNewDiaryViewController class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | timePicker | UIDateTimePicker | Content time of diary |
| 2 | noteTextView | UITextView | Content doctor note |

Method of AddNewDiaryViewController class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | editDiaryAPI |  | Void | update diary to server |

6.2.4.20.Delete diary



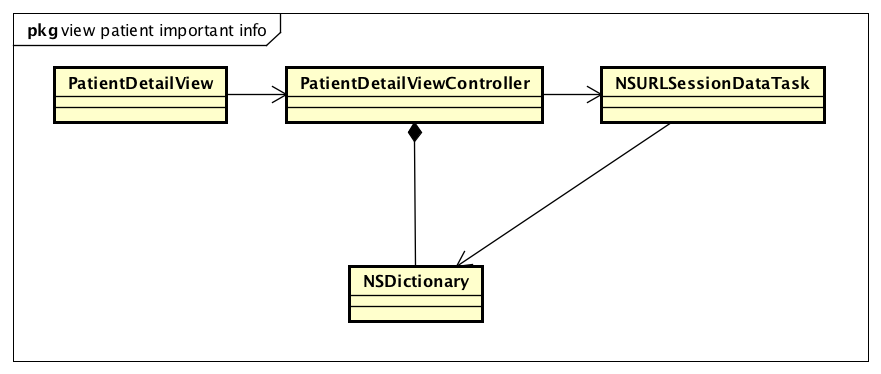
Attribute of DoctorNoteTableViewController class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | selectedDiary | NSDictionary | Content diary user selected |

Method of DoctorNoteTableViewController class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | deleteDiaryAPI: | diaryID:NSString | Void | Delete selected diary to server |

6.2.4.21.View patient statistic info



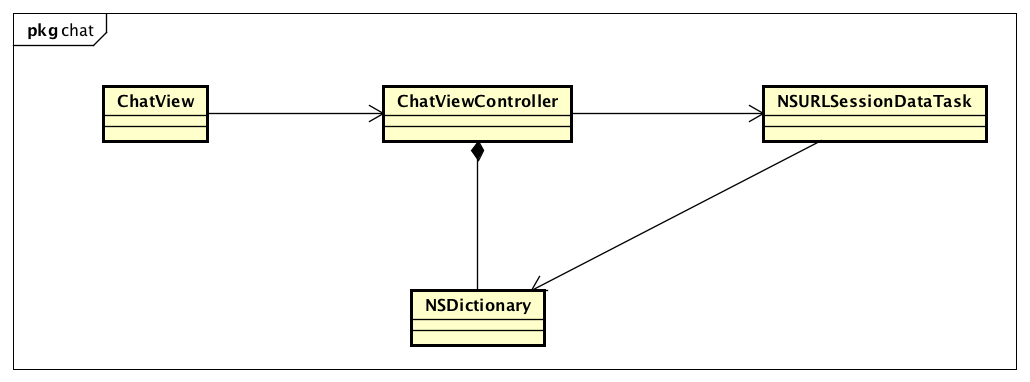
Attribute of PatientDetailViewController class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | responseJSONData | NSDictionary | Content data return from server |
| 2 | experimentArray | NSDictionary | Content experiment note data |

Method of PatientDetailViewController class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | loadExperimentNoteDataFromAPI |  | Void | Load experiment note data from server |

6.2.4.22.Chat



Attribute of ChatViewController class

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Type** | **Description** |
| 1 | responseJSONData | NSDictionary | Content data return from server |
| 2 | messagetArray | NSDictionary | Content message data |
| 3 | tableView | UITableView | Content all messages |
| 4 | hubConnectionForChat | SRHubConnection | Proxy help create and maintain connection to server |

Method of ChatViewController class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Method** | **Parameters** | **Return** | **Description** |
| 1 | getAllMessageFromAPIWithPatientIdD: | patientID:NSString | Void | Load all message with patient |
| 2 | sendMessageToAPI | messageSent:NSString | Void | Send doctor message to server |
| 3 | viewDidLoad |  | Void | Setup hub for real time reaction |

### 6.3. API Documentation

* As mentioned above, Olives system includes 2 API Services as core component, which means another front-end components have to contact with them by using HTTP Request to send API and retrieve HTTP Response.
* To develop front-end application, developers need to know the API in the system to exchange data between each other, these APIs are defined in API Documents (Refer: API Documentation folder for more detail)
* Some example of API documents developers use:

**Get a diary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Url** | | |
| **GET** | api/diary | | |
| Type | Params | Values | Description |
| HEADER | **Email** | string | Email which is used for accessing server api. |
| HEADER | **Password** | string | Password of email |
| HEADER | **Accept-Language** | string | Language which messages sent from server will be formatted.   * Default : **en-US** |
| URL | **Id** | int | Id of diary note. |
|  | | | |
| Role | * Doctor | | |
| Use case | * Only doctor can see his own diary. | | |

* + Response

|  |  |
| --- | --- |
| Status | Content |
| **200** | {  "Diary": {  "Id": 0,  "Owner": 0,  "Target": 0,  "Note": "",  "Time": 0,  "Created": 0,  "LastModified": 0  }  } |
| **401**  No email or password contained in request header. Front-end should redirect user to login page. | {  "Error": "W012"  } |
| **401**  Account has been disabled. | {  "Error": "W002"  } |
| **401**  Account is pending. | {  "Error": "W001"  } |
| **403**  Account is not allowed to access the function. | {  "Error": "W013"  } |
| **404**  Record is not found. | {  "Error": "W006"  } |
| **500**  There is something wrong with server. |  |