МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

РУТ (МИИТ)

Кафедра «Управление транспортным бизнесом и интеллектуальные системы»

**КУРСОВАЯ РАБОТА**

**По дисциплине «Технологии программирования»**

**По теме: «Экспертная система диагностики заболевания пациента.»**

Группа: УВП-211

Студенты:

Тудаков Д.И.

Хмелева Л.А.

Преподаватель: Новиков А.И.

Москва

2020 г.

# Задания по курсовой работе для дисциплины «Технологии программирования»

Задание выдается на группу из 2-4 человек и выполняется на лабораторных занятиях и занятиях по курсовой работе.

Язык программирования -C++.

Описание проекта разрабатываемой информационной системы должно быть выполнено на языке UML.в среде UML Designer.

Документация по заданию должна содержать:

1. Задание по курсовой работе.
2. Диаграммы на языке UML.
3. Краткое описание каждого класса и метода.
4. Руководство пользователя со скриншотами интерфейса программы.
5. Набор тестов и результатов для отладки отдельных методов, модулей и комплексной отладки задания.
6. Тексты программ с комментариями

Диаграммы языка UML:

* Диаграмма вариантов использования (Use Case Diagram)
* Диаграмма классов (Class Diagam)
* Диаграмма деятельности (Activity Diagram)
* Диаграмма состояний (State Machine Diagram)
* Диаграмма последовательности (Sequence Diagram)
* Диаграмма компонентов (Component Diagram)
* Диаграмма развертывания (размещения) (Deployment Diagram)

Необходимо разработать информационную систему в соответствии с заданием. В задании перечислены пользователи системы (актеры) и варианты использования системы актерами. Можно скорректировать список актеров и вариантов использования по согласованию с преподавателем.

Предлагаемый перечень заданий может быть дополнен любым другим заданием по согласованию с преподавателем. Приветствуется разработка проекта на основе реальных заданий.

Общие требования по выполнению задания:

* Каждый пользователь информационной системы должен регистрироваться в системе сам или это делает администратор
* При входе в систему каждый пользователь вводит свой логин и пароль
* Данные в системе хранятся в файлах
* При запуске системы все данные из файлов считываются в оперативную память и хранятся в форме таблиц.
* По завершению работы с системой данные из таблиц записываются обратно в файлы. Файлы при этом не обновляются, а создаются заново на основании данных из таблиц.

1. **Экспертная система диагностики заболевания пациента**

Варианты использования для актеров:

Администратор

* Составление списков пациентов, экспертов, врачей

Эксперт

* Подготовка перечня вопросов для пациента, на основе которых определяется некоторое заболевание. Ответом на вопрос может быть да, нет, некоторое значение из указанного диапазона.

Вопросы разрабатываются группой экспертов. Каждый эксперт разрабатывает список вопросов и ответов отдельно, а затем списки вопросов объединяются в общий список. Правильные ответы составляются на основе усредненных ответов экспертов.

Врач

* Выбор вопросов из списка, на которые должен ответить пациент
* Просмотр результатов ответов и диагноза

Пациент

* Отвечает на вопросы

Система

* ставит диагноз пациенту на основе сравнения ответов пациента с усредненным ответом экспертов

**Поток событий системы**

|  |  |
| --- | --- |
| Основной поток событий | |
| Действия исполнителя | Отклик системы |
| **Под поток «администратор»** | |
| Составление списка врачей/пациентов/экспертов администратором | |
| 1. Регистрация в системе. Вход в учетную запись. | 1. Проверка системой правильного логина и пароля при входе8 |
| 1. Администратор выбирает нужную ему категорию. |  |
| 3.1 Категория «Врач» |  |
| 3.2 Категория «Эксперт» |  |
| 3.3 Категория «Пациент» |  |
| 1. Составить список врачей/экспертов/пациентов | 1. Открывает отдельную вкладку с таблицей записи врачей/экспертов/пациентов |
| 1. Заполнение списка. Его отправка. | 1. Принятие списка |
|  | 1. Сохранение списка в системе |
|  | 1. Закрытие вкладки |
|  | 1. Отображение меню |
| 1. Выход из системы | 1. Завершение работы |
| **Под поток событий «эксперт»** | |
| Подготовка списка вопросов экспертом | |
| 1. Регистрация в системе. Вход в учетную запись. | 2.Проверка системой правильного логина и пароля при входе8 |
|  | 3. Отображение меню |
| 4.Эксперт выбирает нужную ему категорию. |  |
| 4.1 Добавить вопрос |  |
| 4.2 Добавить ответ | 5. Сохранение вопросов и ответов в системе |
| 6.Завершение | 7.Выход из системы |
| **Под поток «врач»** | |
| Прием врачом ответов | |
| * 1. Запись ответов пациентов на вопросы |  |
| * 1. Регистрация в системе. Вход в учетную запись | * 1. Отображение меню |
| * 1. Врач выбирает нужную ему категорию | * 1. Открывается вкладка для записи ответов |
| 6.1 Добавить ответ |  |
| 6.2 Изменить ответ |  |
| 6.3 Удалить ответ | 7. Включаем в системе нужную программу, которая автоматически считывая ответы, выдает диагноз |
| 8.Закрыть программу |  |
| 9.Выход из системы | 10.Завершение работы |

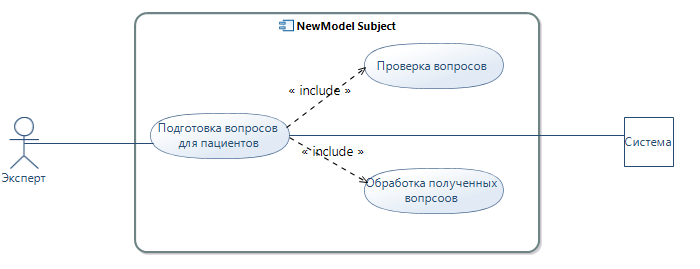
# Диаграммы на языке UML

1) Диаграмма вариантов использования (Use Case Diagram)

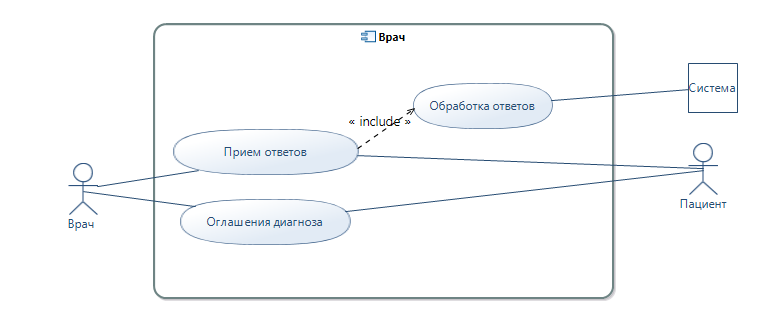
1.Администратор



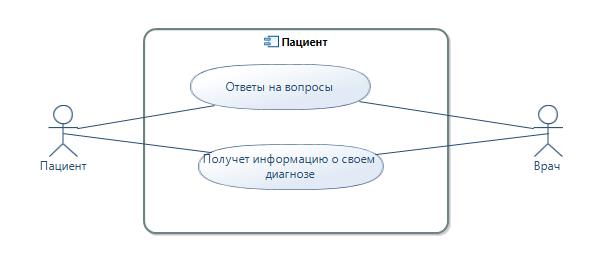
2.Эксперт



3.Врач



4.Пациент



5. Система

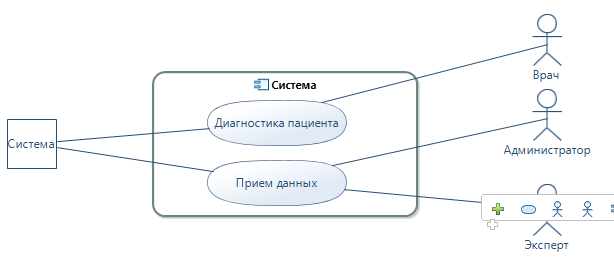


Диаграмма описывает взаимодействия актёров (администратора, врача, эксперта) с системой. Система разделена на 5 подсистем для каждого актера.

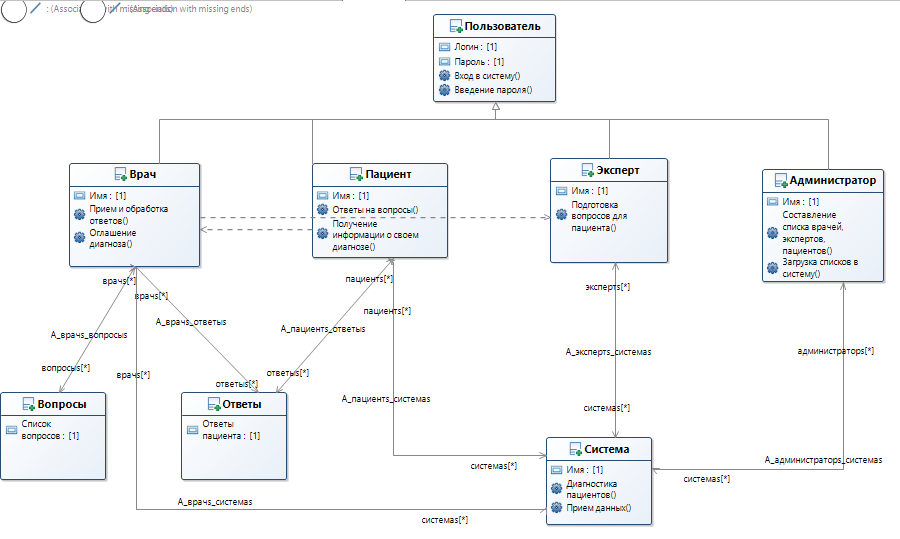
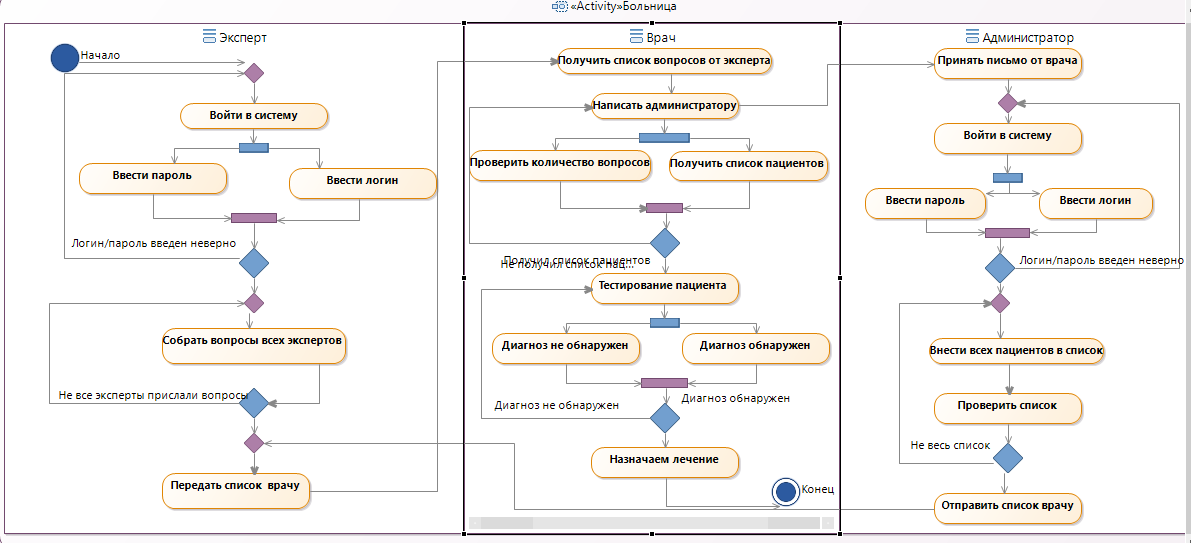
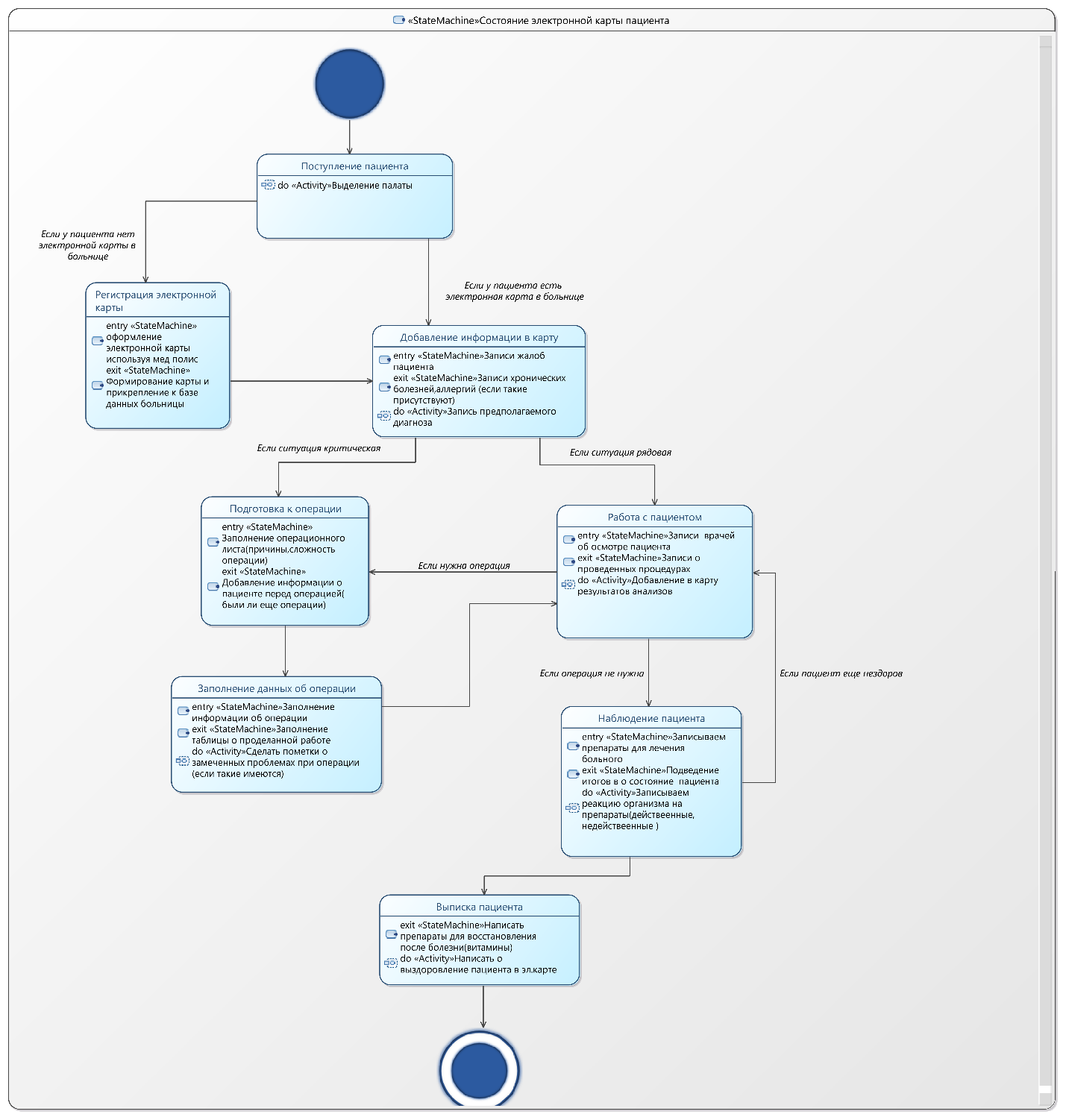
****2) Диаграмма классов (Class Diagam)

Диаграмма описывает иерархию классов системы и их взаимодействие.

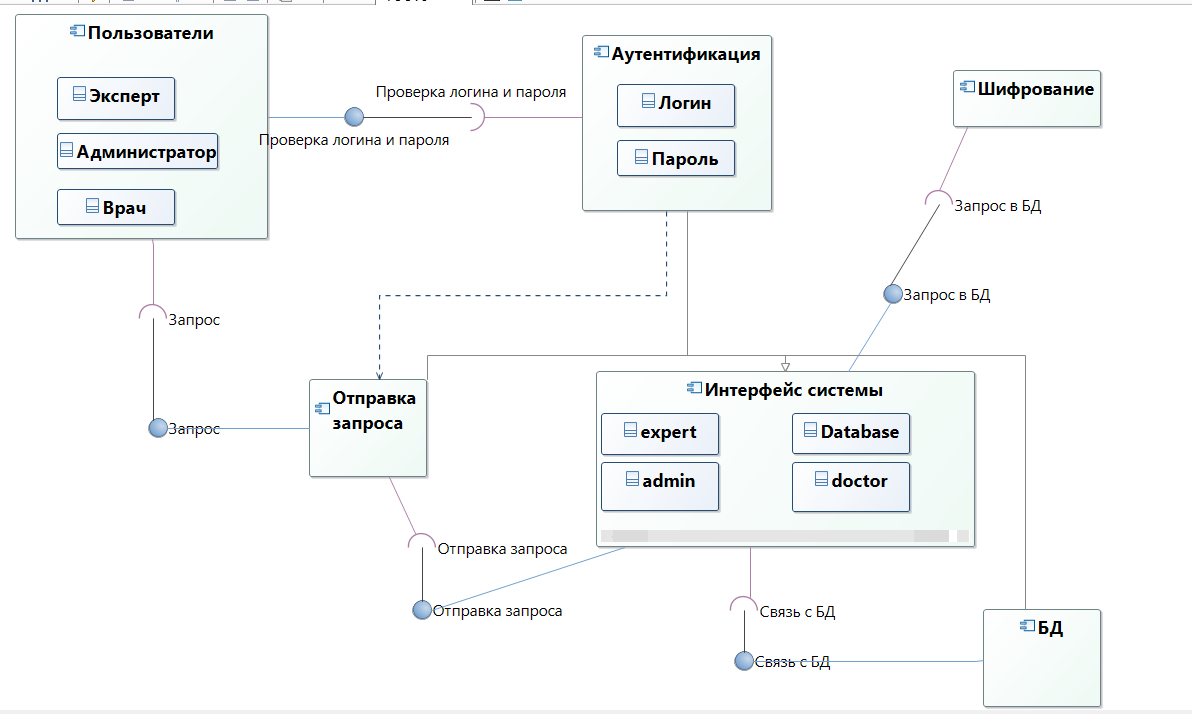
****3) Диаграмма деятельности (Activity Diagram)

В системе взаимодействуют три актера: эксперт, врач и администратор. Данная диаграмма показывает, как происходит процесс выявления диагноза у пациента.

****4) Диаграмма состояний (State Machine Diagram)

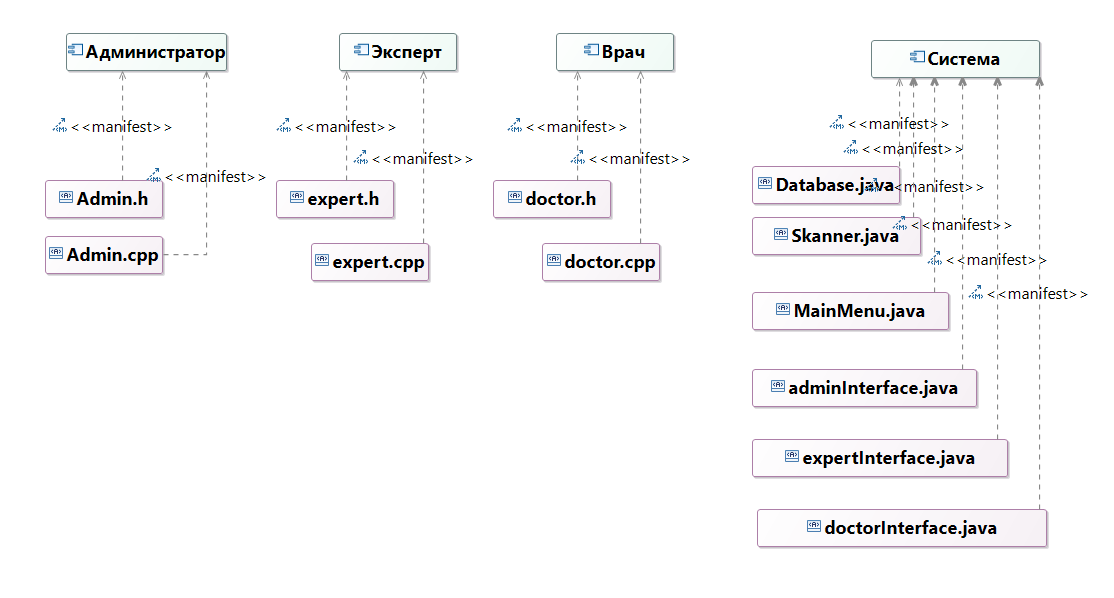
Данная диаграмма описывает состояние электронной карты пациента от этапа поступления его в больницу и до выписки.

5) Диаграмма компонентов (Component Diagram)



6) Диаграмма развертывания (Deployment Diagram)

1.Артефакты

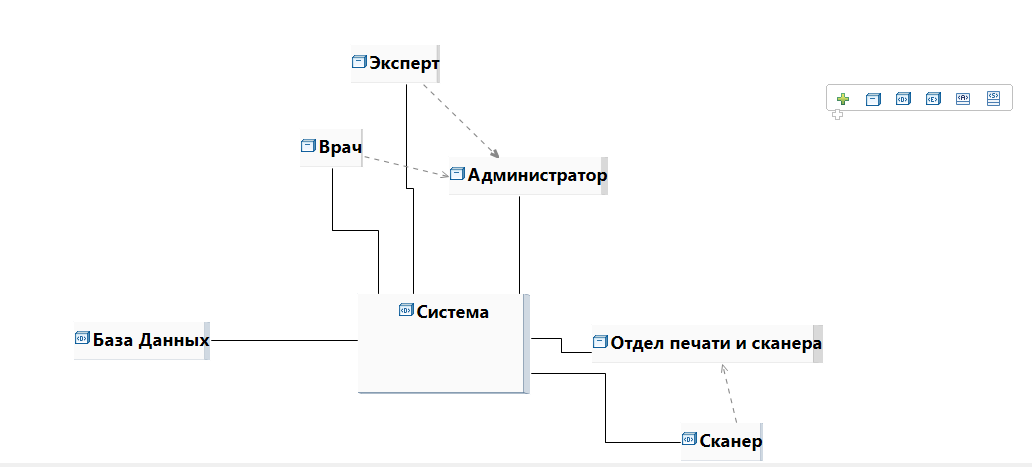


На диаграмме изображены компоненты всей системы.

Администратор, Эксперт и Врач представлены артефактами «\*объект\*.h» и «\*объект\*.cpp»

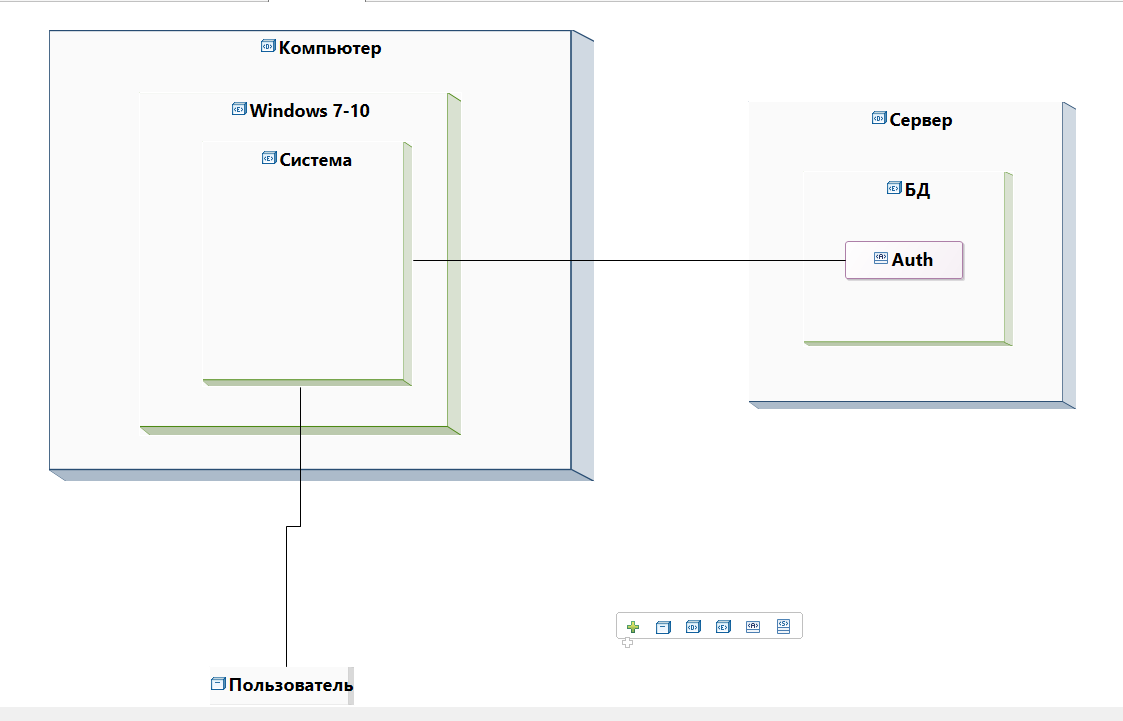
Система же состоит их интерфейсов объектов, меню, Базы Данных и сканера.

2. Узлы системы



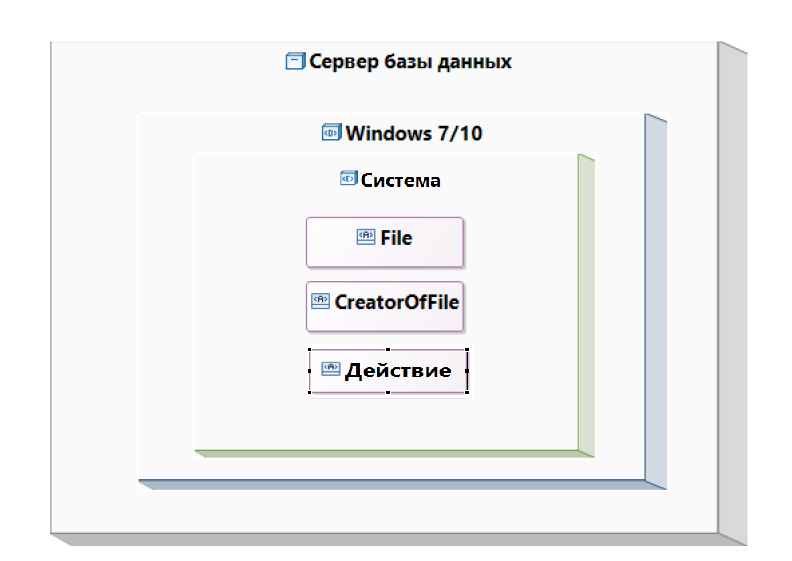
Все пользователи связаны с системой. Так же есть, например, администратор регистрирует врачей и экспертов в систему, тем самым имея зависимость с ними.

3.Авторизация пользователя



Пользователь авторизуется, вводя пароль и логин. БД проверяют, все ли правильно введено. Если да, то пользователь заходит.

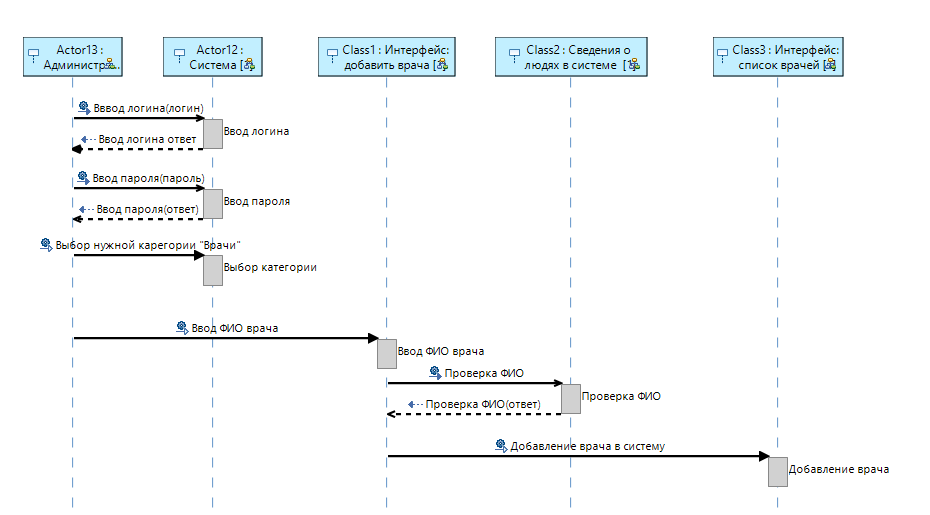
4. База данных



Система -> имя файла -> создатель файла -> дальнейшее действие

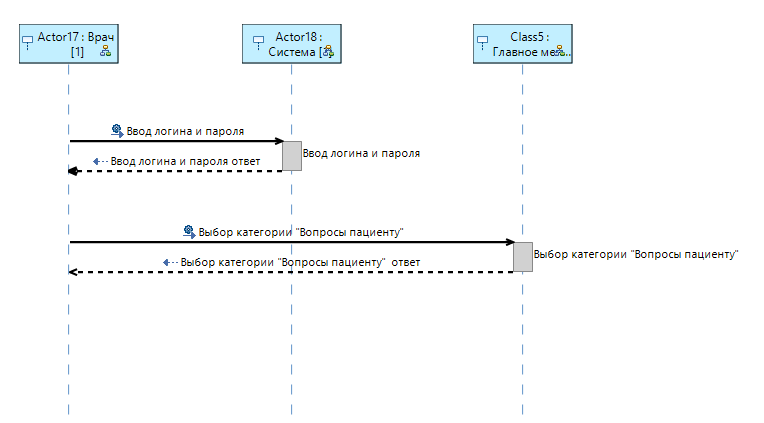
7) Диаграмма последовательности (Sequence Diagram)

1. Администратор (Добавление врача/пациента/эксперта) в систему

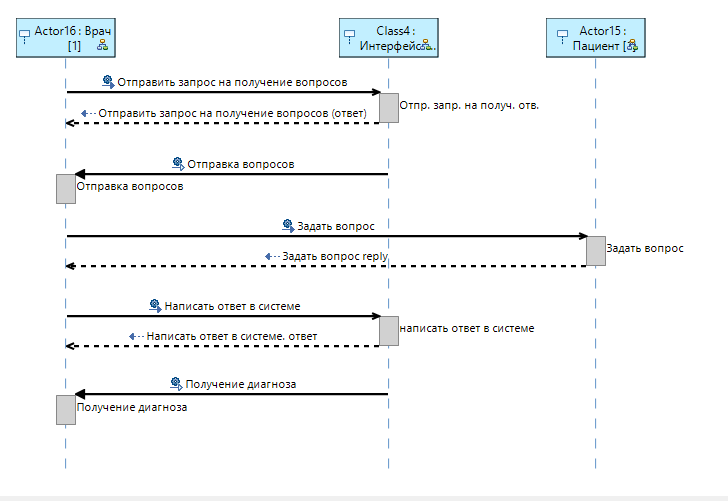
****

2.Врач

-Регистрация в системе и выбор категории



-Взаимодействие с пациентом (задает вопросы для выяснения диагноза)



Здесь врач отправляет интерфейсу системы запрос на вопросы для пациента, задает их (получая обратную связь), отправляет ответы в систему и получает диагноз пациента.

Диаграмма последовательности для пациента есть в пункте с врачом, так что ее второй раз дублировать не будем.