

Воронов Артем Васильевич

МЕДИЦИНСКАЯ ИНФОРМАЦИОННАЯ СИСТЕМА НА ОСНОВЕ MVC-ФРЕЙМВОРКА PLAY FRAMEWORK

Статья описывает результаты работы, которая выполнялась в рамках проекта по разработке информационной системы для медицинского центра Новосибирского государственного университета (НГУ). Проект изначально определялся как некоммерческий, и предполагалось создание программного продукта в рамках архитектуры web-приложений на основе современных технологий и свободно распространяемого программного обеспечения. Целью работы являлось исследование задачи создания медицинской информационной системы и непосредственное решение этой задачи для медицинского центра НГУ – лечебно-профилактического учреждения амбулаторного типа. Основное внимание автор акцентирует на архитектуре полученного в результате программного продукта.

Адрес статьи: www.gramota.net/materials/1/2013/7/10.html

Статья опубликована в авторской редакции и отражает точку зрения автора(ов) по рассматриваемому вопросу.

Источник

Альманах современной науки и образования

Тамбов: Грамота, 2013. № 7 (74). С. 35-39. ISSN 1993-5552.

Адрес журнала: www.gramota.net/editions/1.html

Содержание данного номера журнала: www.gramota.net/materials/1/2013/7/

© Издательство "Грамота"

Информация о возможности публикации статей в журнале размещена на Интернет сайте издательства: www.gramota.net

Вопросы, связанные с публикациями научных материалов, редакция просит направлять на адрес: almanac@gramota.net

УДК 004.4

Технические науки

Статья описывает результаты работы, которая выполнялась в рамках проекта по разработке информационной системы для медицинского центра Новосибирского государственного университета (НГУ). Проект изначально определялся как некоммерческий, и предполагалось создание программного продукта в рамках архитектуры web-приложений на основе современных технологий и свободно распространяемого программного обеспечения. Целью работы являлось исследование задачи создания медицинской информационной системы и непосредственное решение этой задачи для медицинского центра НГУ – лечебно-профилактического учреждения амбулаторного типа. Основное внимание автор акцентирует на архитектуре полученного в результате программного продукта.

Ключевые слова и фразы: медицинская информационная система; электронная регистратура; web-приложение; архитектура MVC; Play Framework; свободное программное обеспечение; open source.

Воронов Артем Васильевич

Новосибирский государственный университет

voronov54@gmail.com

МЕДИЦИНСКАЯ ИНФОРМАЦИОННАЯ СИСТЕМА НА ОСНОВЕ MVC-ФРЕЙМВОРКА PLAY FRAMEWORK[©]

В 2011 году в Новосибирском государственном университете (НГУ) был инициирован проект по разработке медицинской информационной системы. Изначально он определялся как некоммерческий: предполагалось создание программного продукта в рамках архитектуры web-приложений на основе современных технологий и свободно распространяемого программного обеспечения. Целью проекта являлось создание современной информационной системы для медицинского центра НГУ с уклоном на интеграцию с различными учебными процессами.

В ходе проекта автором работы был решён ряд задач, связанных с исследованием проблемы разработки медицинской информационной системы (МИС) для лечебно-профилактических учреждений (ЛПУ) и существующих подходов к её решению, подготовкой и формализацией требований к электронной регистратуре (ЭР), проектированием архитектуры МИС, реализацией функционала ЭР и подготовкой площадки для развития полученного программного продукта.

Ряд аспектов данной работы освещается автором в другом источнике [1, с. 158], в представленной статье основное внимание будет уделено архитектуре полученного в результате программного продукта.

Медицинские информационные системы де-факто должны поддерживать web-интерфейс. В связи с этим, решено было проектировать систему как web-приложение. Соответственно, в основу легла архитектура MVC, которая широко применяется в таких задачах. Технологической платформой является Play Framework – фреймворк, который предназначен для разработки web-приложений, обладающих высокой степенью масштабируемости и интероперабельности. В качестве системы управления базами данных (СУБД) используется PostgreSQL. Языки программирования для реализации функционала на стороне сервера – Java, Scala и инструментарий для объектно-реляционного отображения Ebean, а для разработки клиентской стороны – Javascript с соответствующим набором современных библиотек: jQuery, Twitter Bootstrap. На Рисунке 1 изображены архитектурно-значимые элементы системы.

Структура данных системы базируется на ряде объектов с уровня бизнес-логики приложения, которые при помощи инструментария объектно-реляционного отображения Ebean преобразовываются в схему базы данных. К таким объектам можно отнести персону, пациента, врача и календарь. На Рисунке 2 представлена часть объектной модели, участвующая в работе электронной регистратуры.

На Рисунке 3 изображено два основных контроллера, отвечающих за работу с пациентами и расписанием врачей на уровне бизнес-логики модуля электронной регистратуры.

Рассмотрим взаимодействие системы и пользователя на примере прецедента записи пациента на приём к врачу. Для того чтобы записать пациента на приём, вводятся его фамилия, имя и отчество: на основе этого он идентифицируется в системе. Далее выбираются специальность врача, предполагаемая дата приёма и конкретный медицинский специалист (Рисунок 4).

Далее предполагается получить расписание врача для того чтобы сделать в нём новую запись. Для этого нажимается кнопка «Выбрать», после чего формируется AJAX-запрос и запускается следующий сценарий. Контроллер, отвечающий за записи в календаре врача, получает запрос и инициализирует объектную модель при помощи объектно-реляционного отображения (Рисунок 5). В результате мы получаем все события из календаря указанного врача за некоторый период времени. Далее используется шаблон, который заранее был создан на серверной стороне для графического отображения календаря, и заполняется данными событиями. И уже после этого на сторону клиента возвращается результат (Рисунок 6). Применение AJAX

позволяет на клиентской стороне инициировать другие сценарии и взаимодействовать с системой, в то время пока выполняется данная операция.

Всё остальное взаимодействие пользователя и системы выглядит аналогично: асинхронные запросы, взаимодействие контроллеров, объектной модели и шаблонов на серверной стороне.

Также одной из особенностей архитектуры системы являются элементы безопасности. Главным принципом при разработке МИС считалось обеспечение конфиденциальности. Модель угроз включает в себя все широко известные уязвимости *web*-приложений. Основными из них являются: межсайтовый скриптинг (Cross Site Scripting, XSS), подделка межсайтовых запросов (Cross Site Request Forgery, CSRF) и SQL-инъекции.

В системе применены следующие меры предосторожности для обеспечения защиты от вышеперечисленных угроз:

1. Для борьбы с SQL-инъекциями все запросы в базе данных выполняются через параметризованные запросы, а не через простую конкатенацию. В качестве примера может служить выборка пациента по номеру и серии документа удостоверяющего личность: на Рисунке 7 используются параметры при составлении запроса, а не конкатенация строки *sql* и аргументов метода.

2. Для борьбы с межсайтовым скриптингом используется экранирование спецсимволов HTML, которое автоматически происходит, когда работает модуль *Play Framework*, отвечающий за работу с шаблонами *web*-страниц на серверной стороне.

3. Для борьбы с подделкой межсайтовых запросов используется механизм, при котором с каждой сессией пользователя ассоциируется дополнительный секретный ключ, предназначенный для выполнения запросов. В рамках разработанной системы этот ключ представляет строку длиной в 64 символа, зашифрованную при помощи криптосистемы AES. Кроме того, платформа *Play Framework* имеет API для генерации таких ключей.

Поскольку проект выполнялся в рамках подхода по разработке свободного программного обеспечения, исходный код полученного приложения был размещен в репозитории *GitHub* и находится в публичном доступе, где архитектуру системы можно изучить более подробно, а также доработать в случае необходимости: https://github.com/ArtemVoronov/mis_nsu.

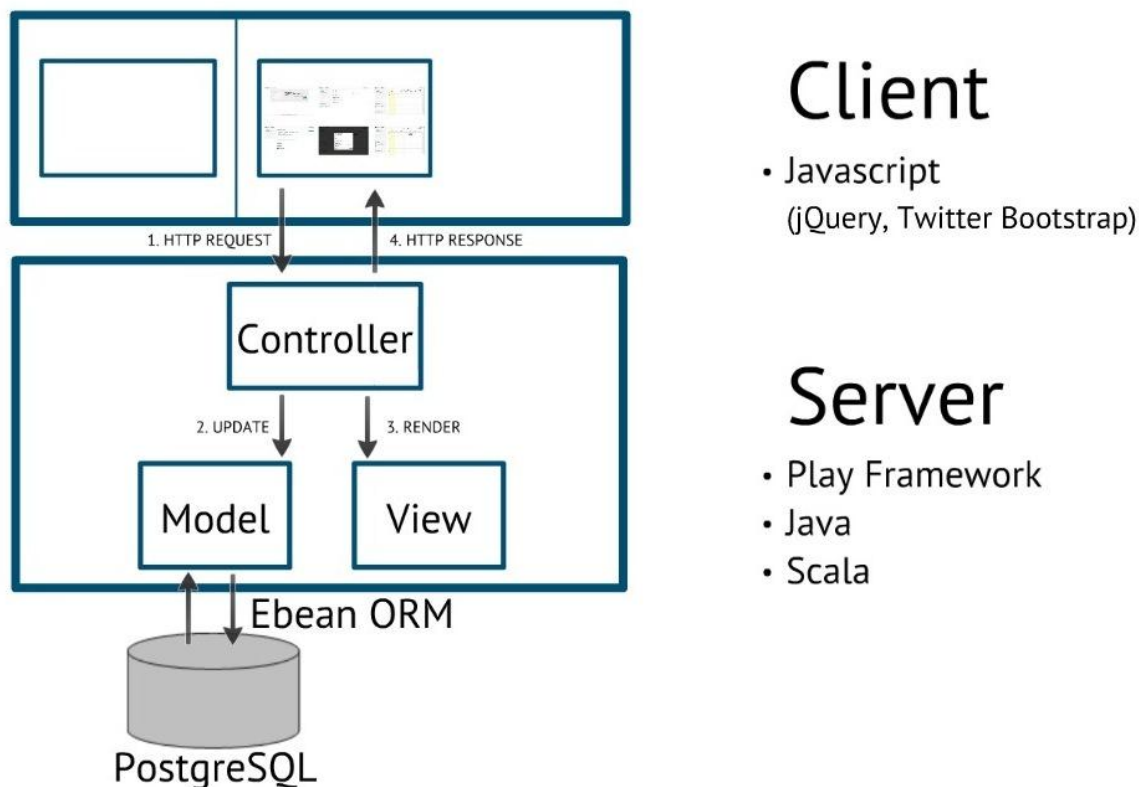


Рис. 1. Архитектура системы

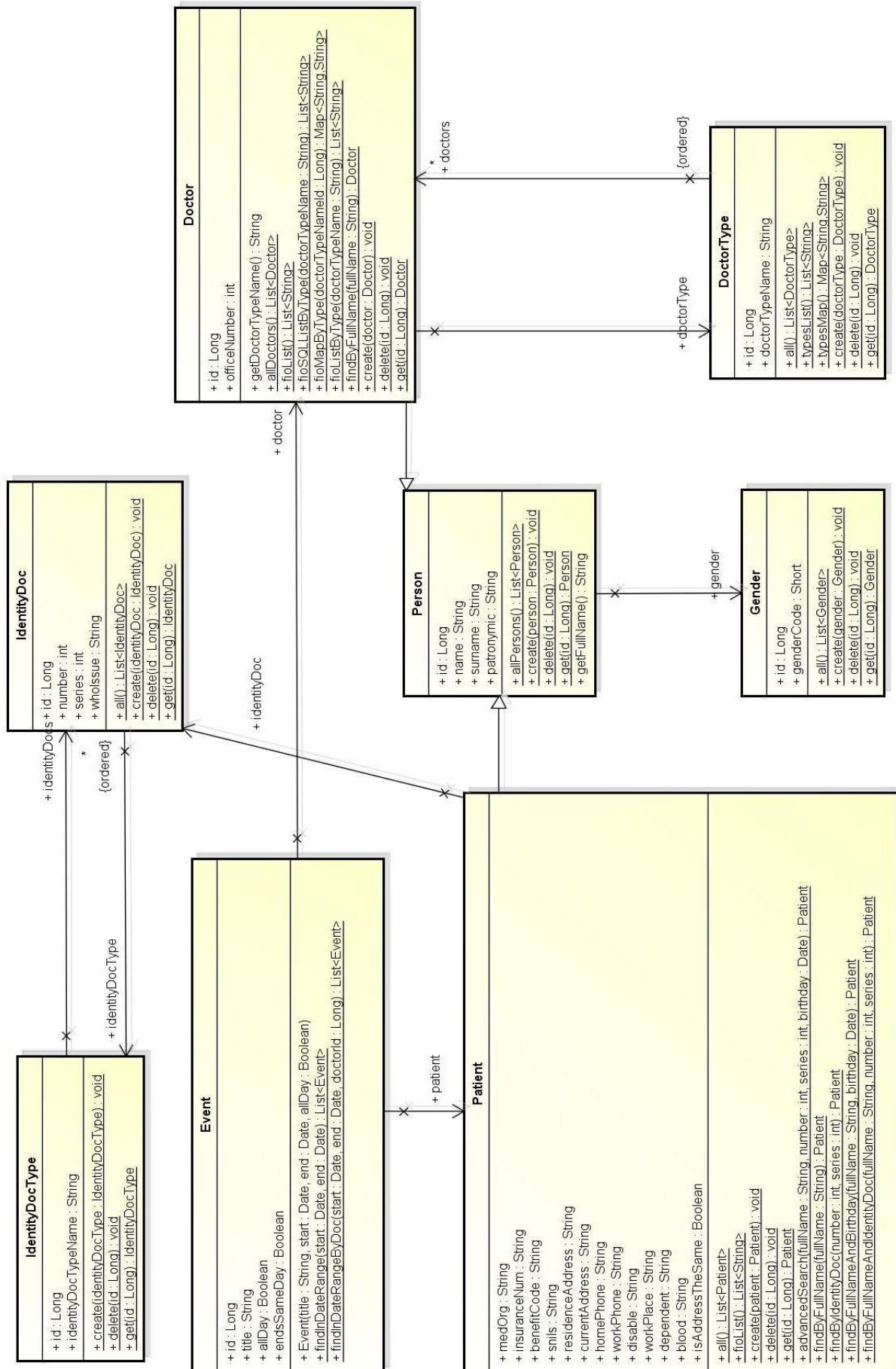


Рис. 2. Объектная модель электронной регистратуры

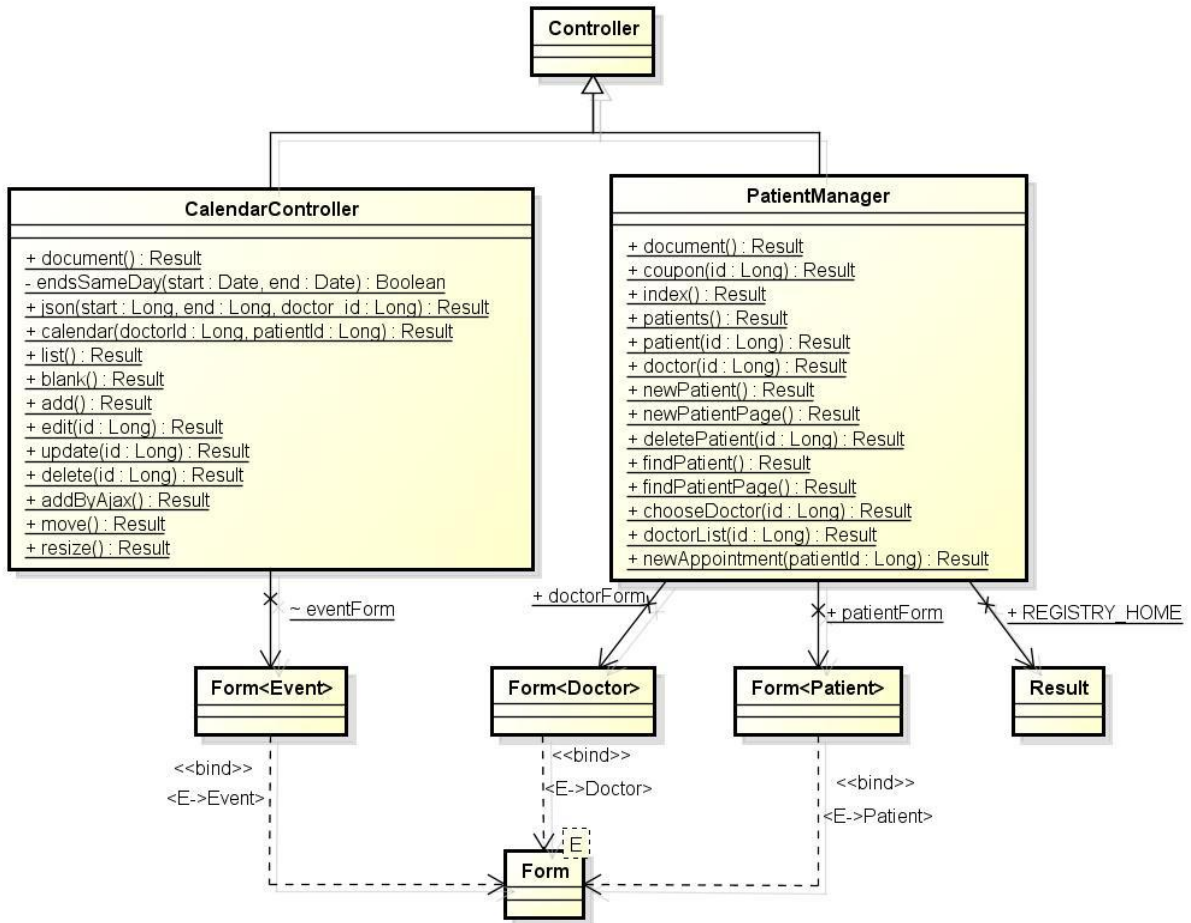


Рис. 3. Основные контроллеры электронной регистратуры

↑ Главная страница Регистратура Артём Воронов

ИНСТРУМЕНТЫ

- Запись на приём
- Создание нового пациента

в РАЗРАБОТКЕ

- Записанные пациенты
- Все пациенты
- Амбулаторная карта
- Расписание врача
- Создать шаблон
- Статистика

Пациент

Иванов Иван Иванович

Дата рождения: 19 янв 1978

Адрес: 630124, Россия, Новосибирская область, г. Новосибирск, ул. Пирогова 2, 121

Запись на приём

Иванов Иван Иванович

Специальность врача:

Предполагаемая дата приёма:

Свободные врачи:

Рис. 4. Запись на приём к врачу

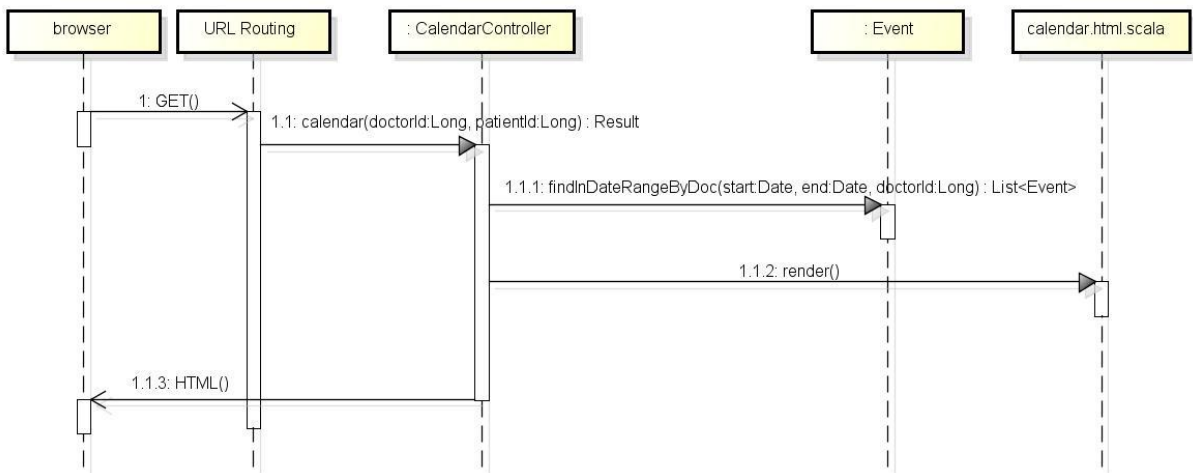


Рис. 5. Реализация прецедента записи на приём к врачу

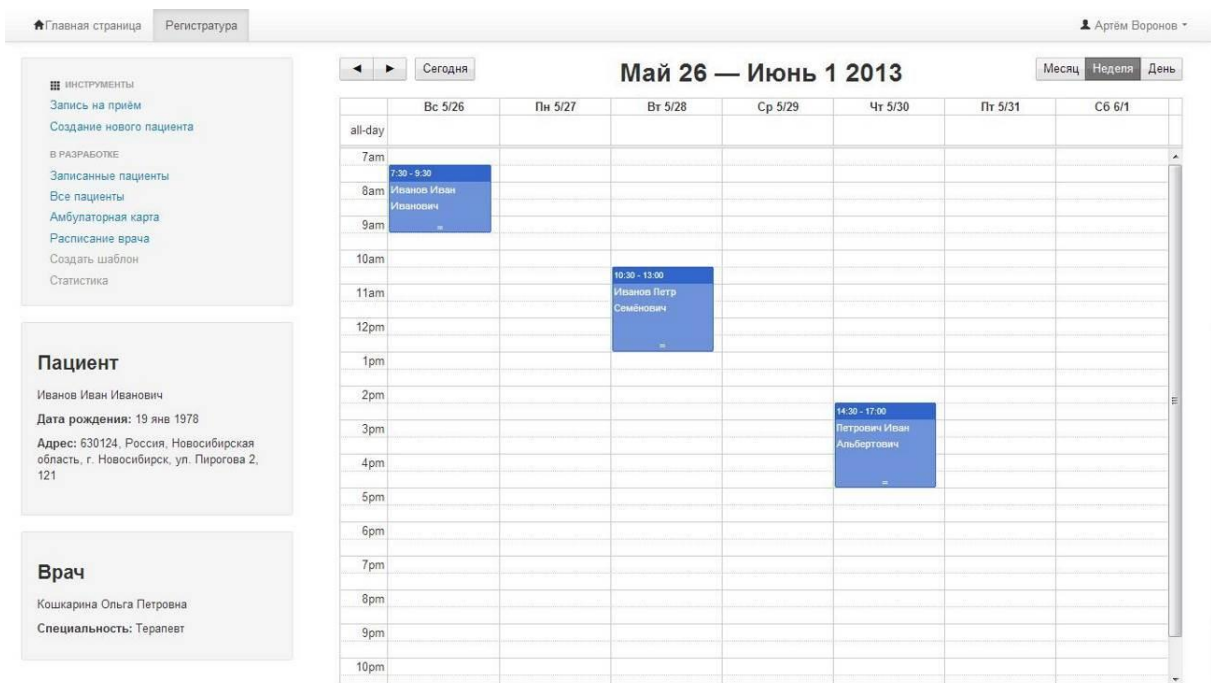


Рис. 6. Расписание врача

```

1 public static Patient findByIdentityDoc(int number, int series) {
2     String sql = "select * from patient
3                 left join identity_doc on patient.identity_doc_id = identity_doc.id
4                 where number=:number and series=:series";
5     SqlQuery sqlQuery = Ebean.createQuery(sql)
6     .setParameter("number", number)
7     .setParameter("series", series);
8     ...
9 }
  
```

Рис. 7. Параметризованный SQL-запрос

Список литературы

1. Воронов А. В. Разработка медицинской информационной системы НГУ // Материалы 51-й международной научной студенческой конференции «Студент и научно-технический прогресс». Информационные технологии. Новосибирск, 2013.