

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)



Факультет _____ «Информатика и системы управления» _____

Кафедра _____ «Компьютерные системы и сети» _____

Отчет по преддипломной практике

Студент Суханов Александр Александрович
(фамилия, имя, отчество)

Группа ИУ6-81

Название предприятия ВАИСПРО

Руководитель
от МГТУ им. Н.Э. Баумана

Должность _____ Смирнова Е.В.
ФИО _____ подпись _____

Руководитель от предприятия

Должность _____ ФИО _____ подпись _____

Оглавление

Введение	3
Проектирование базы данных	4
Разработка моделей данных	4
Нормализация и денормализация таблиц.....	5
Расстановка индексов для увеличения производительности базы данных	6
Разработка архитектуры проекта	7
Проектирование клиентской части приложения	7
Проектирование модулей приложения.....	8
Модуль авторизации и регистрации	9
Модуль просмотра списка больничных листов пациентов	9
Модуль создания нового листка нетрудоспособности	10
Модуль администрирования системы	11
Модуль настройки пользовательского интерфейса	11
Заключение.....	12

Введение

Цель преддипломной практики – проектирование базы данных облачного приложения «Молине», реализация пользовательского интерфейса и отдельных модулей приложения.

В результате прохождения преддипломной практики эти цели были достигнуты. В отчете представлены этапы разработки приложения.

Проектирование базы данных

Проектирование базы данных является одним из основных этапов разработки любого облачного программного обеспечения. Связано это с тем, что облачные приложения особенно сильно подвержены масштабированию, в связи с чем многократно возрастает цена ошибки, допущенной на стадии проектирования.

Проектирование базы данных можно условно разделить на три этапа:

- Разработка моделей данных
- Нормализация и денормализация таблиц
- Расстановка индексов для увеличения производительности базы данных

Разработка моделей данных

Модель данных это интегрированный набор понятий для описания данных, связей между ними и ограничений, накладываемых на данные.

Не следует путать модель данных на стадии проектирования и модель данных на стадии разработки программного обеспечения с использованием средств позволяющих реализовать ORM концепцию построения базы данных.

Для реализации успешного проектирования БД необходимо для каждой модели – инструмента знать три её основных компонента:

- набор правил, определяющих структуру данных;
- определение типов допустимых операций с данными;
- набор ограничений поддержки целостности данных.

Модели данных, как инструменты, делятся на 3 основные категории:

1. Объектные модели данных. В этих моделях используются такие понятия как: классы объектов (типы сущностей), объекты (экземпляры сущностей), свойства классов объектов (атрибуты сущностей), связи между классами объектов.

Среди объектных моделей выделяют наиболее общие типы:

— семантические модели, которые нужны для обеспечения возможности выражения смысла (семантики) предметной области.

— функциональные модели, которые дают представления о функциях автоматизированных процессов.

— объектно—ориентированные модели, то есть те модели, которые расширяют определение класса объектов (сущности) предметной области с целью включения в определение не только свойств, описывающих состояние объекта, но и действий, которые с ним связаны. Данные модели необходимы при разработке NoSQL решений.

Основными сущностями, данные о которых должны храниться в базе данных являются пользователь, больничный лист, печатное устройство и организация.

Таким образом данная модель данных подразумевает наличие следующих таблиц:

- таблица с информацией о пользователях;
- таблица с информацией о больничных;
- таблица с информацией о модели печатного устройства;
- таблица с информацией об организации;

Нормализация и денормализация таблиц

Одним из ключевых понятий СУБД является термин нормальная форма. Нормальная форма — свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

Процесс преобразования отношений базы данных к виду, отвечающему нормальным формам, называется нормализацией. Нормализация предназначена для приведения структуры БД к виду, обеспечивающему минимальную логическую избыточность, и не имеет целью уменьшение или увеличение производительности работы или же уменьшение или увеличение физического объёма базы данных. Конечной целью нормализации является уменьшение потенциальной противоречивости хранимой в базе данных информации.

Несмотря на то, что идеи нормализации весьма полезны для проектирования баз данных, они, к сожалению, не являются универсальным или исчерпывающим средством повышения качества проекта базы данных. Это связано с тем, что существует слишком большое разнообразие возможных ошибок и недостатков в структуре базы, которые невозможно устранить нормализацией.

Таким образом, приведение базы данных к нормальной форме не всегда является оптимальным решением. В связи с этим, для ускорения работы приложения было принято решение о проведении денормализации части таблиц.

Денормализация — намеренное приведение структуры базы данных в состояние, которое не соответствует критериям нормализации и проводится засчет добавления избыточности данных с целью ускорения операции чтения.

За счёт такого перепроектирования операция соединения при выборке становится ненужной и запросы выборки, которые ранее требовали соединения, работают быстрее.

Но следует учесть, что ускорение одних запросов на денормализованной БД может сопровождаться замедлением других запросов, которые ранее выполнялись отдельно на нормализованных отношениях. Также нельзя забывать о том, что денормализация базы данных может нарушить целостность данных, следовательно подвергать денормализации следует, в идеальном случае, только те таблицы, запросы к которым нацелены в большинстве случаев на чтение данных.

В разрабатываемом приложении, особенно в модуле администрирования, запросы на объединение двух и более таблиц встречаются очень часто, поэтому было принято решение произвести денормализацию базы данных. В результате денормализации схема стала выглядеть следующим образом:

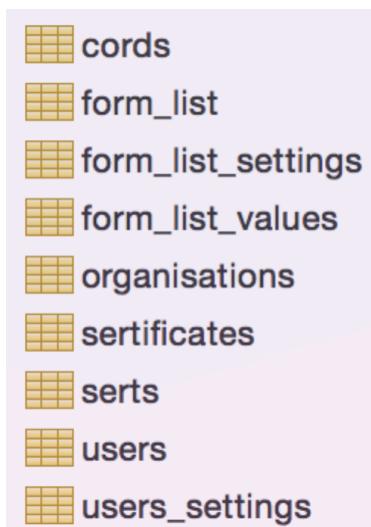


Рисунок 1 – Схема после денормализации

Расстановка индексов для увеличения производительности базы данных

Индексы применяются для быстрого поиска строк с указанным значением одного столбца. Без индекса чтение таблицы осуществляется по каждой записи в таблице, начиная с первой записи, пока не будут найдены соответствующие строки.

Соответственно, чем больше число записей в таблице, тем больше времени требуется для получения селективной выборки. В том случае, таблица содержит индекс по тому столбцу, в котором производится поиск, MySQL может быстро определить позицию без необходимости полного сканирования каждой записи таблицы. Для таблицы, содержащей 1000 строк, это будет как минимум в 100 раз быстрее по сравнению с fullscan поиском.

Таким образом, в результате проведения индексирования таблиц были проиндексированы следующие поля в таблицах:

- cords – кроме PRIMARY KEY по полю id был добавлен индекс по полю user, для ускорения процесса отрисовки документа с помощью сокращения времени запроса
 - organistions – был добавлен также индекс по полю user, что позволило сократить время подстановки данных об организации в форму заполнения документа
 - certificates – составной индекс на полях isDeleted и sert_id для ускорения формирования выдачи списка больничных листов
 - certs – составной индекс на полях owner и cert_date для ускорения формирования выдачи для списка больничных листов оформленных конкретным врачом

Разработка архитектуры проекта

Проектирование клиентской части приложения

Для разработки клиентской части были использованы фреймворки Materialize.css и библиотека jQuery.

Основной задачей использования фреймворка Materialize.css была задача добавления адаптивности.

Materialize.css – фреймворк разработанный в 2015 году. Одним из его основных достоинств является то, что он занимает гораздо меньше места на диске и не переписывает основные стили браузера. Таким образом получается встраивать в свою верстку только лишь отдельные и необходимые элементы.

В первую очередь редизайну подверглось меню входа в приложение. В результате оно выглядит следующим образом:

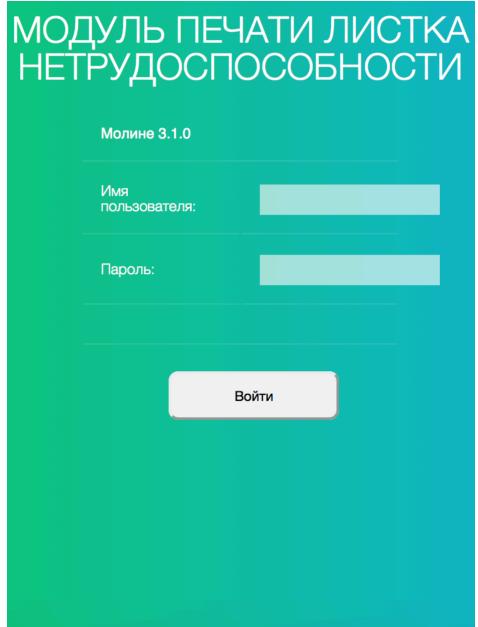


Рисунок 2 - Окно входа в мобильном устройства

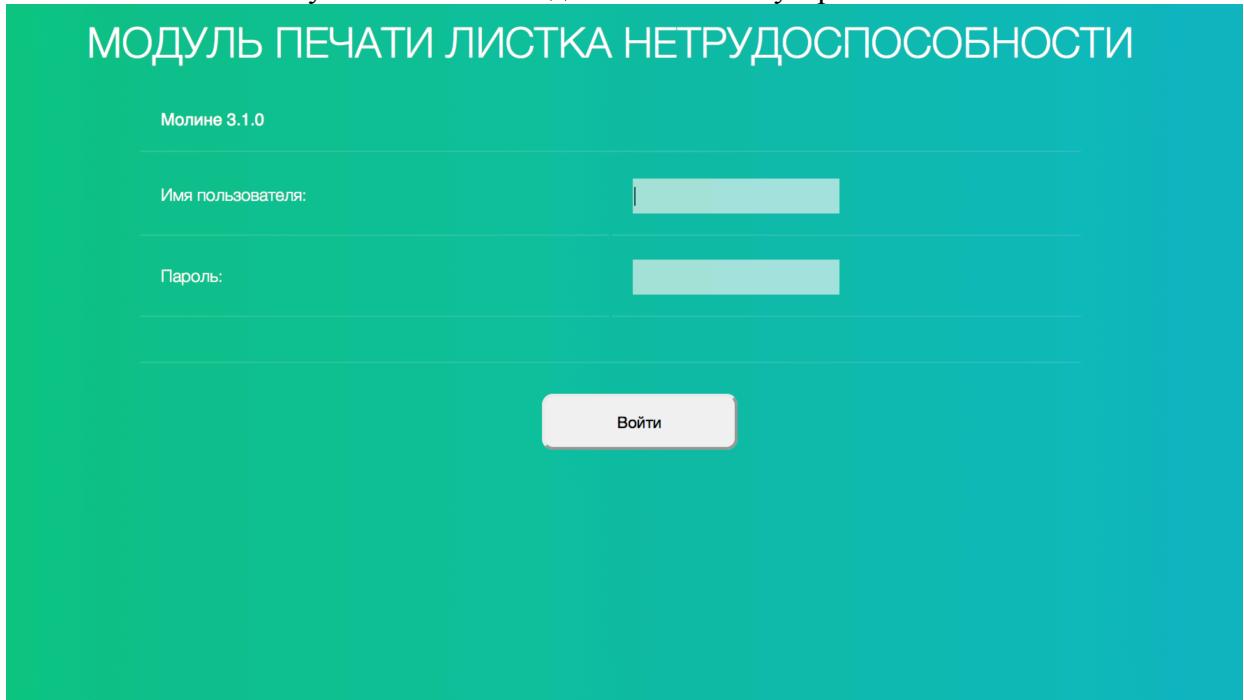


Рисунок 3 – Окно входа с десктопных устройств

Проектирование модулей приложения

Одним из основных требований к подобным приложением является возможность встраивания в существующие медицинские системы и возможность масштабирования приложений.

В связи с этим было принято решение реализовывать облачное приложение в виде сборки отдельных модулей, так как в таком случае каждый из модулей получает некоторые входные данные и генерирует выходные, что обеспечивает возможность встраивания их в любую медицинскую информационную систему.

Модуль авторизации и регистрации

Для реализации механизма авторизации и регистрации был использован встроенный в PHP механизм работы с пользовательскими сессиями.

Сессия представляет собой механизм, позволяющий хранить некоторые данные, индивидуальные для каждого пользователя (например, его имя и номер счета), между запусками сценария.

После прохождения авторизации к пользователю в Cookies записывается его уникальный идентификатор и токен для проверки подлинности пользователя.

Внешний интерфейс модуля представлен на рисунках 2 и 3.

Модуль просмотра списка больничных листов пациентов

Модуль просмотра общего списка больничных листов реализован в виде бесконечного списка пациентов, который можно сортировать по каждому из полей.

В текущей реализации этот модуль получает данные из базы данных облачного приложения, но благодаря тому, что он вынесен в отдельную сущность, данный модуль получать данные и из любого другого хранилища.

Кроме того, в данном модуле присутствует разделение прав администратора и пользователя. В том случае, если пользователь не является администратором системы, ему не будет показана кнопка администратор.

Разумеется, кроме скрытия кнопки, в самом модуле admin.php предусмотрена проверка прав пользователя, однако учитывая то, что данным приложением будут пользоваться малоопытные пользователи, упрощение интерфейса засчет избавления от неактивных элементов сможет существенно сократить время на обучение использованию приложения.

 Государственное бюджетное учреждение здравоохранения города Москвы «Научно-практический психоневрологический центр Департамента здравоохранения города Москвы»	Модуль печати листов нетрудоспособности "Молине 3.1i"	Сегодня: 01-06-2016	
Список листов нетрудоспособности:			
ФИО	Дата	№ листка нетрудоспособности	
1 Иванов Иван Иванович	2015-07-06 11:13:42	123412341234	 
2 Петров Сергей Николаевич	2015-11-15 13:16:05	123412341235	 

Рисунок 4 – Страница просмотра больничных листов в режиме администратора

Модуль создания нового листка нетрудоспособности

Модуль создания листка нетрудоспособности предназначен для внесения в базу нового листка нетрудоспособности.

Во время работы с этим модулем пользователь заполняет такие поля как фамилия, имя и отчество пациента, даты пребывания пациента в стационаре и прочую информацию.

Хранилищем данных может служить как встроенная база данных приложения, так и, в случае, если данных модуль встроен в иную медицинскую систему, хранилище медицинской системы.

Данный модуль также предусматривает возможность показа разных элементов в зависимости от прав пользователя.

 Государственное бюджетное учреждение здравоохранения города Москвы «Научно-практический психоневрологический центр Департамента здравоохранения города Москвы»	Модуль печати листов нетрудоспособности "Молине 3.1i"	Сегодня: 01-06-2016	
Список листов нетрудоспособности:			
ФИО	Дата	№ листка нетрудоспособности	
1 Иванов Иван Иванович	2015-07-06 11:13:42	123412341234	 
2 Петров Сергей Николаевич	2015-11-15 13:16:05	123412341235	 

Рисунок 5 – Страница создания нового листка нетрудоспособности

Модуль администрирования системы

Модуль администрирования системы необходим для возможности удаленного контроля за работой приложения и для возможности удаленной настройки пользовательских параметров.

Данный модуль предоставляет администратору системы следующие возможности:

- регистрация новых пользователей;
- подтверждение регистрации тех пользователей, которые зарегистрировались в системе самостоятельно;
- изменение персональных данных пользователей;
- изменение настроек печати пользователей;
- редактирование информации о пациентах.

Модуль настройки пользовательского интерфейса

Модуль настройки пользовательского интерфейса был реализован в связи с тем, что многие из настроек не требуют какой-либо технической экспертизы со стороны администратора системы.

К таким настройкам относятся:

- редактирование логина/пароля;
- редактирование данных собственных пациентов;

- настройка печати больничного листа;

Модуль настройки пользовательского интерфейса доступен как при создании пользователя в режиме администратора, так и каждому пользователю для своих собственных настроек.

 <p>Департамент здравоохранения г. Москвы Научно-практический психоневрологический центр им. З.П. Соловьева</p>	<p>Сохранить</p>												
<p>Назад</p>													
<table border="1"><thead><tr><th>Модуль печатки листка нетрудоспособности "Молине 3.1"</th><th>Сегодня: 01-06-2016</th></tr></thead><tbody><tr><td>Логин</td><td><input type="text"/></td></tr><tr><td>Пароль</td><td><input type="password"/></td></tr><tr><td>ФИО</td><td><input type="text"/></td></tr><tr><td>Смещение по оси X</td><td><input type="text"/></td></tr><tr><td>Смещение по оси Y</td><td><input type="text"/></td></tr></tbody></table>	Модуль печатки листка нетрудоспособности "Молине 3.1"	Сегодня: 01-06-2016	Логин	<input type="text"/>	Пароль	<input type="password"/>	ФИО	<input type="text"/>	Смещение по оси X	<input type="text"/>	Смещение по оси Y	<input type="text"/>	
Модуль печатки листка нетрудоспособности "Молине 3.1"	Сегодня: 01-06-2016												
Логин	<input type="text"/>												
Пароль	<input type="password"/>												
ФИО	<input type="text"/>												
Смещение по оси X	<input type="text"/>												
Смещение по оси Y	<input type="text"/>												

Рисунок 6 – Модуль настройки пользовательского интерфейса

Заключение

В результате проделанной работы работы была спроектирована архитектура облачного приложения, выбран подход для его программной реализации, выбрана архитектура базы данных и разработана клиентская часть приложения. Кроме того, был произведен редизайн окна авторизации для более комфортного доступа с мобильных устройств.