

Санкт-Петербургский государственный университет

Программная инженерия

«Разработка рабочего модуля «Травма»»

Отчет по учебной (ознакомительной) практике

Выполнил:

Захарова Оксана Михайловна, группа 22.М07-мм

Научный руководитель:

Доцент каф. Системного программирования, Луцив Дмитрий Вадимович

Санкт-Петербург

2022 год

Оглавление

Введение.....	3
Цели и задачи.....	3
Описание существующей системы.....	5
Анализ существующей базы данных.....	9
Выбор новых технологий	11
Заключение	13
Список литературы	14

Введение

В современном мире уже нельзя представить жизнь без использования информационных технологий – как в рабочем процессе, так и в личной жизни. Медицина – одна из сфер человеческой жизни, где сбор, накопление больших наборов данных происходит каждый день. Поэтому внедрение информационных технологий в медицинские организации стало необходимостью.

Проведение телеконсультаций пациентов, передача информации о больных в другие отделения или поликлиники/больницы, использование машинного обучения для точной и быстрой постановки диагноза, открытие и закрытие больных листов, назначение анализов – только малая часть современного использования информационных технологий в медицинских организациях.

Внедрение началось уже давно, но технологии написания развиваются и становятся все лучше и лучше с каждым днем. Поэтому поддержка разработанных модулей, приложений – как web-приложений, так и desktop-приложений, становится все труднее и требует большего количества времени для добавления новых пользовательских возможностей.

Исходя из данных проблем были выдвинуты следующие цели и задачи.

Цели и задачи

Цель работы – разработка рабочего модуля «травма» для медицинской организации

Задачи:

1. Обзор предметной области
2. Анализ и обратный инжиниринг существующей базы данных
3. Выбор новых технологий
4. Реализация возможности подписания документов при помощи электронной подписи

5. Реализация нового рабочего модуля с интуитивно-понятным для пользователей интерфейсом

Для данной работы были поставлены следующие задачи:

1. Обзор предметной области
2. Анализ и обратный инжиниринг существующей базы данных
3. Выбор новых технологий

Описание существующей системы

Разработка данного модуля ведется в компании, занимающейся разработкой ПО для медицинских учреждений не только в городе Санкт-Петербург, но и в других регионах нашей страны.

При обсуждении проекта с генеральным директором компании были поставлены следующие требования к системе:

1. Пользовательский интерфейс должен быть интуитивно понятен и прост
2. Доступ к наиболее используемым функциям должен быть быстр, элементы модуля должны быть видны
3. При заполнение документов (например посещение пациента), переключение между полями для заполнения осуществляется с помощью клавиши «Tab» на клавиатуре
4. Осуществить подписание документов электронной подписью
5. Осуществить возможность загрузки подписанных документов в государственные учреждения, т.е. реализовать возможность электронного документооборота

Причинами разработки нового модуля являются:

1. Сложность поддержки старого модуля из-за старой версии используемого языка программирования
2. Нет осуществления электронного документооборота, а также подписание документов электронной подписью
3. Прошлая попытка «переписания» модуля не отвечала требованиям заказчика

Модуль «Травма» представляет собой web-приложение, в котором работают врачи травматологи. В данном приложении есть 4 вида пользователей:

1. Администратор базы данных

Данный тип пользователей имеет доступ к любым изменениям шаблонов в базе данных, а также к таблице с всеми пользователями, т.е может создавать, изменять, удалять пользователей.

2. Регистратура

Имеется возможность регистрации нового пациента при его обращении в медицинское учреждение, а также формирует необходимые журналы для сбора статистики.

3. Врач

Создает обращение пациента, вносит данные о его самочувствии, ставит диагноз, открывает больничные листы при необходимости и т.д.

4. Заведующий отделением

Формирует необходимые отчеты, контролирует листы нетрудоспособности.

Физическое представление системы:

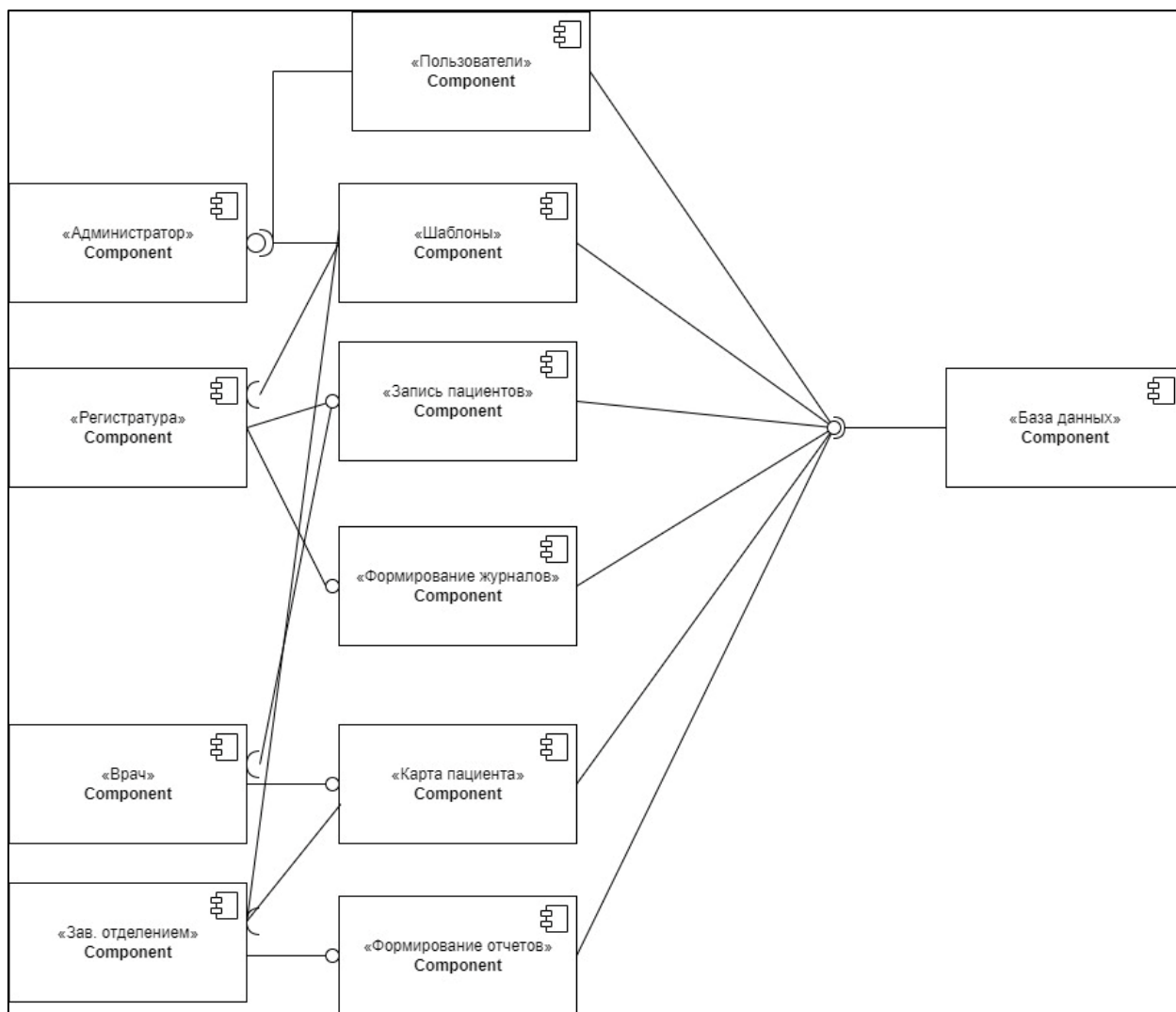


Рисунок 1. Диаграмма компонент

Краткое описание модуля:

1. Администратор базы данных вносить изменения в таблицы, хранящие шаблоны для заполнения, а также вносить данные в таблицу с пользователями, например изменять логин и/или пароль, добавлять новых пользователей
2. Регистратура регистрирует пациентов, заполняя некоторые поля шаблонами, которые создал администратор базы данных. Также, если пациент обращается не в первый раз, реализован поиск пациентов по базе данных, при выборе пациента все поля заполняются автоматически согласно записям в БД
3. После регистрации пациента, он отображается у врача во вкладке «Регистрационный журнал». При создании обращения, данные, указанные

регистратурой, заполняются автоматически. Врач заполняет только необходимые ему поля. Некоторые поля также можно заполнить шаблонами, созданными администратором. Если врач назначил пациенту следующую дату приема, то обращение пациента попадает во вкладку «Очередь», где будет находиться, пока пациента «не выпишут», то есть не закроют больничный лист

4. Регистратура и Заведующий отделением формируют журналы и отчеты соответственно согласно данным, сохраненным в базе данных, по всем пациентам. Отчеты формируются в формате PDF, изредка экспортируются в Excel, в зависимости от требований заказчика

Вывод: Данная система функционирует уже очень давно, и такой процесс работы травматологии полностью устраивает врачей и интуитивно-понятен для новых пользователей и программистов. Поэтому при разработке нового модуля процесс должен обязательно остаться таким же.

Анализ существующей базы данных

Так как данная реализация проекта проходит в конкретной компании, и данный модуль используется в некоторых медицинских учреждениях, база данных не требует создания.

База данных состоит из 36 таблиц (см. рис. 2). 12 из них связаны между собой, оставшиеся используются как шаблоны для быстрого заполнения врачом, например, диагноза или жалоб, которые в других таблицах хранятся не по id, а по наименованиям в текстовом формате. В таблицах, содержащих только 2 - столбца, являются простыми шаблонами, при большем количестве столбцов либо добавляется указание включения в отчеты наименования шаблона, либо такие таблицы используются как дерево вариантов, где значения вложены друг в друга.

При первом просмотре таблиц, главным недостатком является большое количество столбцов в некоторых таблицах – от 70 и выше. Причем некоторые столбцы являются пустыми. Также для добавления новых функций скорее всего потребуются создание новых таблиц.

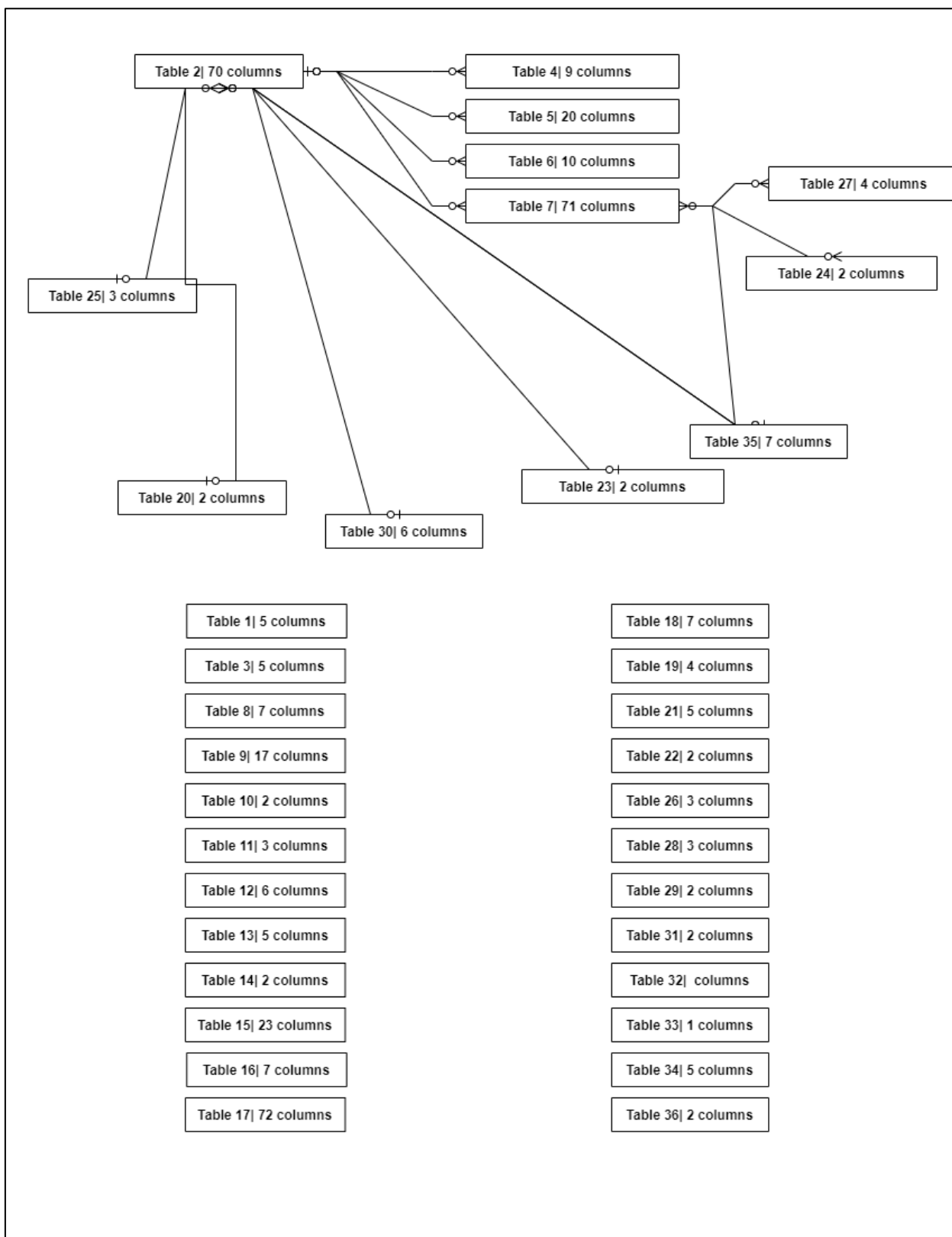


Рисунок 2. Схема базы данных

Выбор новых технологий

В ходе рассмотрения функционирования уже написанного рабочего модуля было выявлено, что он был написан с использованием языка программирования PHP версии 5, а также JavaScript, языка разметки HTML и языка стилей CSS.

Для разработки нового рабочего модуля были выбраны следующие языки программирования и фреймворки:

Backend:

1. СУБД MariaDB

В топе по использованию СУБД, не зависимо от года, можно заметить, что лидирующее место занимают Oracle, MySQL, PostgreSQL и MariaBD.

Изначально, MariaBD разрабатывалась как продукт полностью совместимый с MySQL, но лучше и быстрее. Главные плюсы данного продукта

- a. Open-Source проект, что позволяет самостоятельно улучшать работу данной СУБД
- b. Новый тип хранения таблиц – колоночное хранилище, позволяет увеличивать производительность, а также масштабировать хранилище данных
- c. Параллельное выполнение запросов
- d. Представляет несколько мощных представлений хранения данных, которых нет в MySQL

2. Python (фреймворк FastAPI)

Для web-разработки у Python есть множество фреймворков – Django, Flask, Pyramid, FastAPI и многие другие. Самым популярным является фреймворк Django, FastAPI – достаточно новая инфраструктура для разработки web-приложений и поддерживается только с Python версии 3.6 и

выше. Но он имеет большие достоинства перед фреймворком Django, которые необходимы для выполнения дальнейшей работы:

- a. Является одним из самых популярных фреймворков Python для создания сервисов REST
- b. Высокая производительность
- c. Встроенная проверка данных
- d. Поддержка асинхронного кода
- e. Масштабирование при разворачивании моделей машинного обучения

3. SQLAlchemy

Данная библиотека предназначена для работы с реляционными СУБД с использованием технологий ORM¹. Позволяет описывать структуры БД на языке Python без использования SQL, что обеспечивает:

- a. Безопасность
- b. Повышение производительности
- c. Переносимость с одной СУБД на другую

Frontend:

1. JQuery

Одна из самых популярных библиотек JS. Основные преимущества:

- a. Реализация AJAX
- b. Кроссплатформенность
- c. Быстрая связь JS с HTML

¹ ORM – технология программирования, которая связывает бд с концепциями ООП, создавая «виртуальную объектную базу данных»

Заключение

В ходе учебной практики были рассмотрены технологии проектирования web-приложения, указаны их основные преимущества, выявлены причины написания модуля на новых технологиях, был осуществлен разбор существующей базы данных – ее описание и недостатки. Все поставленные задачи были выполнены.

Дальнейшая работа представляет собой написание web-приложения и доработка его функциональности согласно поставленным требованиям.

Список литературы:

1. Преимущества использования FastAPI [электронный ресурс] – режим доступа: <https://worksolutions.ru/useful/fastapi-vs-django/> (дата обращения 25.12.22)
2. Преимущества использования SQLAlchemy [электронный ресурс] – режим доступа: <https://lectureswww.readthedocs.io/6.www.sync/2.codding/9.databases/2.sqlalchemy/> (дата обращения 25.12.22)
3. Описание JQuery [электронный ресурс] – режим доступа: <https://boodet.online/blog/что-такое-jquery-i-dlya-chego-eto-nuzhno-pljusy-i-minusy-boodet-online> (дата обращения 25.12.22)
4. Сравнение MariaBD и MySQL [электронный ресурс] – режим доступа: <https://netpoint-dc.com/blog/mariadb-mysql-2019/> (дата обращения 25.12.22)