

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математического и компьютерного моделирования

Разработка информационной системы для центра диетологии

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса 451 группы

направление 38.03.05 — Бизнес-информатика

механико-математического факультета

Кузнецовой Елены Алексеевны

Научный руководитель
доцент, к.э.н.

Ю.В. Мельникова

Зав. кафедрой
зав.каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2019

Введение. Информационные технологии являются движущей силой развития различных сфер общества, начиная со всевозможных узкоспециализированных областей промышленности и заканчивая социальными сетями. На любом современном предприятии самые простые операции производятся с помощью компьютеризованных систем. В том случае, даже если решить задачу с помощью достижений в области информационных технологий не удастся, то можно попробовать с их помощью задачу упростить, что уже намного облегчит работу и приблизит к выполнению поставленной проблемы.

На сегодняшний день основным фактором создания конкурентоспособной и высокоэффективной компании являются оптимальные стратегии управления бизнесом. Можно говорить о том, что эффективное управление такой же по важности ресурс, как и, например, материальные блага. Именно этот ресурс дает возможность с наибольшей эффективностью и наименьшим количеством затраченного времени реагировать на непрерывно изменяющуюся рыночную ситуацию, осуществлять контроль над всеми аспектами деятельности предприятия, сосредотачивать усилия в тех областях, где они необходимы на данный момент времени.

На сегодняшний день для динамичных, развивающихся предприятий стало просто необходимым использование разнообразных информационных систем. С их помощью организации имеют возможность автоматизировать работу и значительно сэкономить время. Самое трудное и важное — выстроить единую систему, которая будет отвечать всем запросам всех пользователей, то есть сотрудников всей организации. Каждое подразделение может иметь свое программное обеспечение, которое оптимизировано исключительно под особенности отдела. Информационная система призвана совместить их все в рамках одной интегрированной программы, которая работает с общей базой данных так, что все отделы могут делиться информацией и поддерживать связь друг с другом. Такой интегрированный подход может дать большую отдачу, если фирма сможет четко установить систему. В последнее время центров диетологии появляется все больше и больше, поскольку год от года растет процент граждан желающих вести здоровый образ жизни. Кроме того, правильное питание становится необходимостью в случае ряда заболеваний.

В связи с этим, возникает необходимость разработки информационной системы для центра диетологии. Поэтому тема данной бакалаврской работы является актуальной.

Цель бакалаврской работы - разработка информационной системы для центра диетологии. Практическая значимость исследования заключается в разработке системы, обеспечивающей эффективность результатов коммерческой или некоммерческой деятельности центра. Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Рассмотреть бизнес-процессы, протекающие в выбранной предметной области;
2. Разработать информационную систему;
3. Создать графическую оболочку для информационной системы «Центр диетологии».

Следует отметить, что в ходе эксплуатации информационной системы может появиться необходимость внесения изменений в связи с возможными изменениями бизнес-процессов организации.

Цель дипломной работы будет считаться достигнутой в том случае, если после выполнения поставленных задач мы получим функционирующую информационную систему, предназначенную для эксплуатации в центре диетологии.

Выпускная квалификационная работа состоит из введения, четырех глав, заключения, списка используемых источников и приложений. В первой главе описываются задачи проектирования информационных систем, описывается предметная область, которая представляет собой работу центра диетологии. Также рассматриваются основные бизнес-процессы протекающие в данной информационной системе. Вторая глава посвящена ознакомлению с графическим языком для создания диаграмм UML. В данной главе были также построены диаграммы прецедентов, последовательности и деятельности, а также был описан поток событий для диаграмм последовательности и деятельности. Третья глава посвящена средствам разработки для создание базы данных. В частности, рассматривается методология проектирования реляционных баз данных, а также СУБД PostgreSQL. Также в данной главе была создана ER-диаграмма базы данных и создана непосредственно

сама база данных. В четвертой главе рассказывается о средствах разработки интерфейса- языке программирования Python и библиотеки PyQt. Также здесь был разработан графический интерфейс для информационной системы.

Основное содержание работы. Проектирование любой информационной системы всегда начинается с определения цели проекта. Основная задача любого успешного проекта заключается в том, чтобы на момент запуска системы и в течение всего времени ее эксплуатации можно было обеспечить:

- требуемую функциональность системы и степень адаптации к изменяющимся условиям ее функционирования;
- требуемую пропускную способность системы;
- требуемое время реакции системы на запрос;
- безотказную работу системы в требуемом режиме, иными словами — готовность и доступность системы для обработки запросов пользователей;
- простоту эксплуатации и поддержки системы;
- необходимую безопасность.

Производительность является главным фактором, определяющим эффективность системы. Хорошее проектное решение служит основой высокопроизводительной системы.

Проектирование информационных систем охватывает три основные области:

- проектирование объектов данных, которые будут реализованы в базе данных;
- проектирование программ, экранных форм, отчетов, которые будут обеспечивать выполнение запросов к данным;
- учет конкретной среды или технологии, а именно: топологии сети, конфигурации аппаратных средств, используемой архитектуры (файл-сервер или клиент-сервер), параллельной обработки, распределенной обработки данных и т.п.

В реальных условиях проектирование — это поиск способа, который удовлетворяет требованиям функциональности системы средствами имеющихся технологий с учетом заданных ограничений.

К любому проекту предъявляется ряд абсолютных требований, например максимальное время разработки проекта, максимальные денежные вложения в проект и т.д. Одна из сложностей проектирования состоит в том, что оно не является такой структурированной задачей, как анализ требований к проекту или реализация того или иного проектного решения. Бизнес-процесс - последовательность действий, направленная на получение заданного результата, ценного для организации. Системы управления, построенные на принципах управления функциями, представляют собой иерархическую пирамидальную структуру подразделений, сгруппированных по выполняемым функциям. Моделирование бизнес-процессов помогает решить сразу две задачи:

1. Изучение бизнеса. Графическое изображение в виде схем, т.е. моделирование бизнес-процессов позволяет быстрее понять особенности работы компании и выявить возможные «узкие места».
2. Обеспечение наглядности. Схематическое изображение работы компании помогает руководителю и владельцу бизнеса намного быстрее понять суть проблемы и оценить предложенные варианты решения.

Основным бизнес-процессом, происходящим в центре диетологии, является продажа услуг центра (план питания, тренировки) . Это является бизнес-процессом, т.к. процесс состоит из нескольких следующих друг за другом операций, которые в свою очередь направлены на преобразование входных информационно-материальных потоков (желания клиента, денежные средства) с целью получения результатов, представляющих ценность для клиента, т.е. приобретенная услуга. Продажа услуги – это внутренний бизнес-процесс, описывающий внутреннюю деятельность организации, поток управления находится внутри одного отдела и не пересекает его границ.

Важно помнить, что продажа услуги – основная статья дохода центра, поэтому этому процессу нужно уделять особое внимание, а для удобства клиента - важно оперативно и незамедлительно оформить заказ на услугу, внести всю необходимую информацию о клиенте в базу, в общем сделать все, что бы человек мог без лишних проблем получить желанную услугу.

UML является языком широкого профиля, это - открытый стандарт, который использует графические обозначения для создания абстрактной модели

системы, которая называется UML-моделью. UML был создан для определения, визуализации, проектирования, документирования программных систем. UML не является языком программирования. Для демонстрации функциональности и поведения системы были рассмотрены три диаграммы:

1. диаграмма вариантов использования;
2. диаграмма деятельности;
3. диаграмма последовательности;

Диаграммы прецедентов (вариантов использования) описывают взаимоотношения и зависимости между группами вариантов использования и действующих лиц, участвующими в процессе. Важно понимать, что диаграммы вариантов использования не предназначены для отображения проекта и не могут описывать внутреннее устройство системы. Диаграммы вариантов использования предназначены для упрощения взаимодействия с будущими пользователями системы, с клиентами, и особенно пригодятся для определения необходимых характеристик системы. Другими словами, диаграммы вариантов использования говорят о том, что система должна делать, не указывая сами применяемые методы.

Вариант использования описывает, с точки зрения действующего лица, группу действий в системе, которые приводят к конкретному результату. Варианты использования являются описаниями типичных взаимодействий между пользователями системы и самой системой. Они отображают внешний интерфейс системы и указывают форму того, что система должна сделать.

Действующее лицо является внешним источником (не элементом системы), который взаимодействует с системой через вариант использования. Действующие лица могут быть как реальными людьми (например, пользователями системы), так и другими компьютерными системами или внешними событиями.

Действующие лица представляют не физических людей или системы, а их роли. Это означает, что когда человек взаимодействует с системой различными способами (предполагая различные роли), он отображается несколькими действующими лицами. Например, человек, работающий в службе поддержки и принимающий от клиентов заказы, будет отображаться в системе как «участник отдела поддержки» и «участник отдела продаж».

Диаграмма последовательностей (Sequence diagram) - диаграмма поведения, на которой показано взаимодействие и подчеркнута временная последовательность событий.

В UML взаимодействие объектов понимается как обмен информацией между ними. При этом информация принимает вид сообщений. Кроме того, что сообщение несет какую-то информацию, оно некоторым образом также влияет на получателя. В этом плане язык UML полностью соответствует основным принципам ООП, в соответствии с которыми информационное взаимодействие между объектами сводится к отправке и приему сообщений.

Диаграмма последовательностей относится к диаграммам взаимодействия UML, описывающим поведенческие аспекты системы, но рассматривает взаимодействие объектов во времени. Другими словами, диаграмма последовательностей отображает временные особенности передачи и приема сообщений объектами.

Диаграммы последовательностей необходимы для уточнения диаграмм прецедентов, более детального описания логики сценариев использования. Диаграммы последовательностей обычно содержат объекты, которые взаимодействуют в рамках сценария, сообщения, которыми они обмениваются, и возвращаемые результаты, связанные с сообщениями.

Диаграмма деятельности — диаграмма UML, выглядящая наиболее простой, поскольку напоминает привычную всем блок-схему. На самом же деле диаграмма активности - это нечто большее, чем блок-схема, хотя цели у них похожи: обе они отображают некий алгоритм. Диаграммы деятельности позволяют моделировать сложный жизненный цикл объекта, с переходами из одного состояния (деятельности) в другое. Но этот вид диаграмм может быть использован и для описания динамики совокупности объектов. Они применимы и для детализации некоторой конкретной операции, причем, предоставляют для этого больше возможностей, чем «классическая» блок-схема. Диаграммы деятельности описывают переход от одной деятельности к другой, в отличие от диаграмм взаимодействия, где акцент делается на переходах потока управления от объекта к объекту. На диаграмме деятельности можно не только показать параллельно выполняемые действия, но и указать состояния объектов (так же, как и на представлениях конечных автоматов, о

которых нам так много говорили в университетах), также есть возможность показывать распределение ролей и т. д.

На практике диаграммы деятельности применяются в основном двумя способами:

- Для моделирования процессов;

В этом случае внимание фокусируется на деятельности с точки зрения «Actor», которые работают с системой.

- Для моделирования операций

В этом случае диаграммы деятельности играют роль «продвинутых» блок-схем и применяются для подробного моделирования вычислений.

На первое место при таком использовании выходят конструкции принятия решения, а также разделения и слияния потоков управления (синхронизации).

Реляционная база данных — это совокупность взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа.

Методология проектирования реляционных баз данных состоит из 7 этапов:

1. Определение основных типов сущностей, присутствующих в представлении данного пользователя о предметной области приложения. Документирование выделенных типов сущностей.
2. Определение важнейших типов связей, существующих между сущностями, выделенными на предыдущем этапе. Определение кардинальности связей и ограничений участия его членов. Документирование типов связей. При необходимости могут использоваться диаграммы «сущность-связь» (ER- диаграммы).
3. Связывание атрибутов с соответствующими типами сущностей или связей. Идентификация простых и составных атрибутов, простых и множественных атрибутов, а также производных атрибутов. Документирование сведений об атрибутах.
4. Определение доменов для всех атрибутов в каждой локальной концептуальной модели данных. Документирование сведений о доменах атрибутов.

5. Определение потенциального ключа для каждого типа сущности, если таких ключей окажется несколько, выбор среди них первичного ключа. Специализация или генерация типов сущностей (необязательный этап). Определение суперклассов и подклассов для типов сущностей.
6. Разработка диаграмм «сущность - связь» (ER- диаграмм), содержащих концептуальное отражение представлений пользователя о предметной области приложения.
7. Обсуждение локальных концептуальных моделей данных с конечными пользователями для получения подтверждения того, что данная модель корректно отражает представления пользователя о приложении и предприятии.

PostgreSQL - это мощная объектно-реляционная система управления базами данных с открытыми исходными текстами. Она разрабатывается на протяжении более 15 лет и улучшает архитектуру, завоевывая репутацию надежной, ингерированной и масштабируемой СУБД. Она запускается на всех основных платформах, включая Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), и Windows. Она полностью соответствует ACID, имеет полную поддержку ключей, объединений, представлений, триггеров, и хранимых процедур (на разных языках). Она включает большинство типов данных SQL92 и SQL99, включая INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, и TIMESTAMP. Она также поддерживает хранение больших двоичных объектов (BLOB's), включая картинки, звук, или видео. Она имеет API для C/C++, Java, Perl, Python, Ruby, Tcl, ODBC, и многие другие, и документацию на русском языке.

Модель сущность-связь (ER-модель) — модель данных, позволяющая описывать концептуальные схемы предметной области.

ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями. На основе сего этого была построена ER-диаграмма для центра диетологии.

Python — высокоуровневый язык программирования с динамической типизацией, поддерживающий объектно-ориентированный, функциональный и императивный стили программирования. Это язык общего назначения, на

котором можно одинаково успешно разрабатывать системные приложения с графическим интерфейсом, утилиты командной строки, научные приложения, игры, приложения для web и много другое. Он является одним из наиболее популярных современных языков программирования.

Возможности и особенности:

- классы являются одновременно объектами со всеми ниже приведёнными возможностями;
- наследование, в том числе множественное;
- полиморфизм (все функции виртуальные);
- инкапсуляция (два уровня — общедоступные и скрытые методы и поля).

Особенность — скрытые члены доступны для использования и помечены как скрытые лишь особыми именами;

- специальные методы, управляющие жизненным циклом объекта: конструкторы, деструкторы, распределители памяти;
- перегрузка операторов (всех, кроме is, '.', '=' и символьных логических);
- свойства (имитация поля с помощью функций);
- управление доступом к полям (эмуляция полей и методов, частичный доступ, и т. п.);
- методы для управления наиболее распространёнными операциями (истинностное значение, len(), глубокое копирование, сериализация, итерация по объекту, и т.п.);
- метапрограммирование (управление созданием классов, триггеры на создание классов, и др.);
- полная интроспекция;
- классовые и статические методы, классовые поля;
- классы, вложенные в функции и классы.

Однако, как и все остальные языки программирования, Python имеет ряд минусов, таких как:

- низкое быстродействие;
- невозможность модификации встроенных классов;
- глобальная блокировка интерпретатора (GIL) - GIL (Global Interpreter Lock) — особенность, присущая CPython, Stackless и PyPy, но отсутствующая в Jython и IronPython.

Qt – одна из ведущих платформ для разработки приложений с графическим пользовательским интерфейсом (GUI) под большинство существующих ныне операционных систем. Также Qt – одноименный набор библиотек, лежащий в основе платформы.

PyQt – набор «привязок» графического фреймворка Qt для языка программирования Python, выполненный в виде расширения Python. PyQt работает на всех платформах, поддерживаемых Qt :Linux и другие UNIX подобные ОС, Mac OS X и Windows. Существует 2 версии: PyQt5, поддерживающий Qt 5, и PyQt4, поддерживающий Qt 4.

PyQt практически полностью реализует возможности Qt. А это более 600 классов, более 6000 функций и методов, включая:

- Существующий набор виджетов графического интерфейса;
- стили виджетов;
- доступ к базам данных с помощью SQL (ODBC, MySQL, PostgreSQL, Oracle);
- QScintilla, основанный на Scintilla виджет текстового редактора;
- поддержку интернационализации (i18n);
- парсер XML;
- поддержку SVG;
- интеграцию с WebKit, движком рендеринга HTML;
- поддержку воспроизведения видео и аудио.

Основными модулями PyQt являются:

- QtCore – основные не графические классы: система сигналов и слотов, платформонезависимые абстракции для Unicode, потоков, разделяемой памяти, регулярных выражений и т. д.;
- QtGui – компоненты графического интерфейса (элементы управления), основанные на визуальном представлении;
- QtNetwork – классы для сетевого программирования. Например, клиентов и серверов через UDP и TCP;
- QtOpenGL – классы, позволяющие использовать OpenGL и 3Dграфику в приложениях PyQt;
- QtScript – классы, позволяющие использовать встроенный в Qt интерпретатор JavaScript для управления приложением;

- QSql — классы для интеграции с базами данных с помощью SQL;
- QtSvg — классы для отображения векторной графики в формате SVG;
- QDom — классы, реализующие обработку XML;
- uic — реализация обработки XML-файлов, созданных в QtDesigner, для генерации из них Python-кода графического интерфейса

Сочетая Python и Qt можно быстро создавать удобные приложения с графическим интерфейсом. Кроме того кроссплатформенность как библиотеки Qt, так и интерпретатора Python, позволяет переносить разработанные на PyQt приложения из одной операционной системы в другую без каких-либо изменений.

Заключение. Благодаря информационной системе, разработанной с учетом специфики конкретной предметной области, администраторы получают возможность осуществлять учет и хранение информации о работе центра, а также правильно выстроенная информационная система способствует достижению максимальной эффективности его работы.

Целью данной работы являлась разработка информационной системы для диетологического центра, которая позволила упростить некоторые бизнес-процессы и сделать работу центра более эффективной. В процессе разработки системы были получены навыки построения UML-диаграмм, ER-диаграмм и баз данных. Также изучены язык программирования Python и библиотека PyQt, с помощью которых был разработан графический интерфейс для информационной системы, что позволило приобрести профессиональные навыки в области проектирования информационных систем.

В ходе проделанной работы удалось более детально изучить процесс проектирования информационных систем, разобраться в тонкостях создания и управления базами данных.

В результате работы:

- цели и задачи, поставленные в бакалаврской работе, были выполнены;
- разработана и реализована ИС, которую можно улучшать и менять ее функционал в соответствии с требованиями внутренней и окружающей среды.