ГОУ ВПО – Донецкий Национальный Университет

Физико-Технический факультет

Лабораторная работа по дисциплине

«ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

Кафедра Компьютерных Технологий

Подготовил студент группы ИВТ-6

Нижникова О. Л.

Проверил преподаватель

Шарий

ЛАБОРАТОРНАЯ РАБОТА №1

Курс «Технологии разработки программного обеспечения»

Тема: Составление спецификаций требований заказчика программного продукта.

Цель: Научиться формализовать требования заказчика программного продукта и оформлять техническое задание на создание программного продукта.

Примечание к лабораторным работам.

В рамках лабораторных работ №1-№5 студент занимается разработкой одного программного продукта. В роли заказчика продукта выступает проработчик курса «Технологии создания программного продукта» на кафедре компьютерных технологий. Каждому студенту необходимо разработать простейшую информационную систему по определенной предметной области. Предметную область студент может либо предложить самостоятельно (например, совпадающую с тематикой его курсовой работы), либо получить по варианту (приложение 1.1). Предметные области у разных студентов не могут повторяться.

Задание:

1. Провести интервью с заказчиком, на основе которого сформулировать цель и задачи проекта по созданию программного продукта.

2. На основе интервью с заказчиком определить пользователей продукта.

3. На основе интервью с заказчиком определить нефункциональные требования к продукту.

4. Провести дополнительно самостоятельный анализ предметной области.

5. Сформировать спецификации вариантов использования продукта.

6. Представить модель объекта автоматизации (информационной системы) в виде блок-схем и/или простого текста (примеры – приложение 1.2).

7. Оформить техническое задание на создание программного продукта. В качестве стандарта использовать ГОСТ 34.602-89 или IEEE Std.830-1998.

8. Оформить отчет, включающий ответы на контрольные вопросы, составленное техническое задание.

Контрольные вопросы:

1. Кратко опишите основные этапы разработки программного продукта.

Анализ требований к проекту

На этом этапе формулируются цели и задачи проекта, выделяются базовые сущности и взаимосвязи между ними. То есть, создается основа для дальнейшего проектирования системы.

В рамках данного этапа не только фиксируются требования заказчика, но и проводится их формирование.

При анализе требований определяются сроки и стоимость разработки ПО, формируется и подписывается ТЗ на разработку программного обеспечения.

Проектирование

На основе предыдущего этапа проводится проектирование системы.

Разрабатывается функциональная спецификация ПО: выбирается архитектура системы, оговариваются требования к аппаратному обеспечению, определяется набор орг. мероприятий, которые необходимы для внедрения ПО, а также перечень документов, регламентирующих его использование.

Реализация

Строятся прототипы, как целой системы, так и ее частей. Прототипы дают возможность глубже вникнуть в проблему и принять все необходимые проектные решения еще на ранних этапах проектирования. Врезультата этапа реализации появляется рабочая версия продукта.

Тестирование продукта

Тестирование тесно связано с такими этапами разработки программного обеспечения как проектирование и реализация. В систему встраиваются специальные механизмы, которые дают возможность производить тестирование системы на соответствие требований к ней, проверку оформления и наличие необходимого пакета документации.

Результатом тестирования является устранение всех недостатков системы и заключение о ее качестве.

Внедрение и поддержка

Внедрения системы обычно предусматривает следующие шаги:

* установка системы,
* обучение пользователей,
* эксплуатация.

2. Что представляют собой артефакты программного продукта?

Артефакт — это любой созданный искусственно элемент программной системы.

К элементам программной системы, другими словами, артефактами являются те информационные элементы, которые тем или иным способом используются при работе программной системы и входят в ее состав.  
3. Опишите основные стратегии разработки ПО.

Каскадная модель (иногда переводят как модель «Водопад») — модель процесса разработки программного обеспечения, в которой процесс разработки выглядит как поток, последовательно проходящий фазы анализа требований, проектирования, реализации, тестирования, интеграции и поддержки

V-Model  является моделью разработки информационных систем , направленной на упрощение понимания сложностей, связанных с разработкой систем. Она используется для определения единой процедуры разработки программных продуктов, аппаратного обеспечения и человеко-машинных интерфейсов.

Итеративный подход (англ. iteration - «повторение») в разработке программного обеспечения — это выполнение работ параллельно с непрерывным анализом полученных результатов и корректировкой предыдущих этапов работы. Проект при этом подходе в каждой фазе развития проходит повторяющийся цикл PDCA: Планирование — Реализация — Проверка — Оценка (англ. plan-do-check-actcycle).

RAD (от англ. быстрая разработка приложений) — концепция создания средств разработки программных продуктов, уделяющая особое внимание быстроте и удобству программирования, созданию технологического процесса, позволяющего программисту максимально быстро создавать компьютерные программы. Практическое определение: RAD — это жизненный цикл процесса проектирования, созданный для достижения более высокой скорости разработки и качества ПО, чем это возможно при традиционном подходе к проектированию

4. Кратко охарактеризуйте системы принципов разработки ПО (PSP, TSP, CMM).

PersonalSoftwareProcess - Согласно этой модели каждый программист должен уметь:

* учитывать время, затраченное на работу над проектом;
* учитывать найденные дефекты;
* классифицировать типы дефектов;
* оценивать размер задачи;
* осуществлять систематический подход к описанию результатов тестирования;
* планировать программные задачи;
* распределять их по времени и составлять график работы.
* выполнять индивидуальную проверку проекта и архитектуры;
* осуществлять индивидуальную проверку кода;
* выполнять регрессионное тестирование.

TeamSoftwareProcessделает ставку на самоуправляемые команды численностью 3-20 разработчиков. Команды должны:

* установить собственные цели;
* составить свой процесс и планы;
* отслеживать работу;
* поддерживать мотивацию и максимальную производительность.

CapabilityMaturityModel

Данная модель определяет пять уровней зрелости процесса разработки ПО.

* Начальный — процесс разработки носит хаотический характер. Определены лишь немногие из процессов, и успех проектов зависит от конкретных исполнителей.
* Повторяемый — установлены основные процессы управления проектами: отслеживание затрат, сроков и функциональности. Упорядочены некоторые процессы, необходимые для того, чтобы повторить предыдущие достижения на аналогичных проектах.
* Определенный — процессы разработки ПО и управления проектами описаны и внедрены в единую систему процессов компании. Во всех проектах используется стандартный для организации процесс разработки и поддержки программного обеспечения, адаптированный под конкретный проект.
* Управляемый — собираются детальные количественные данные по функционированию процессов разработки и качеству конечного продукта. Анализируется значение и динамика этих данных.
* Оптимизируемый — постоянное улучшение процессов основывается на количественных данных по процессам и на пробном внедрении новых идей и технологий.

5. Состав персонала разработки программного продукта.

* Teamleader распределяет задачи, контролирует сроки, ищет решения и попутно программирует. В крупных компаниях специалист может не касаться программирования, а выполнять роль руководителя проекта. В этом случае обязанности тим лидера включают в себя общение с заказчиком, обсуждение требований к продукту, составление ТЗ и реже подбор персонала.
* Senior developer (Ведущий разработчик) – человек ответственный за качество и своевременность работ по разработке информационно-программных систем.Обладает глубокими, структурированными знаниями и работает внутри проектной команды, совершенно не имея необходимости контактировать с представителями менеджмента заказчика.
* MiddleDev– Программист выполняет задачи по написанию и базовому тестированию порученных ему компонентов системы, работает developer по внешним спецификациям. Поддерживает Junior разработчиков, занимается как архитектурой проектов, так и модульной реализацией, производит реализацию работоспособности прототипов.
* Junior Developer– молодой разработчик с малым количеством опыта или вовсе без такового, только начавший работу в избранной технологичной области.

6. Укажите основные метрики качества программного продукта.

* количество строк кода;
* цикломатическая сложность;
* анализ функциональных точек;
* количество ошибок на 1000 строк кода;
* степень покрытия кода тестированием;
* совместимость;
* количество классов и интерфейсов.

7. В чем заключается процесс инспектирования проекта? Укажите принципы

инспектирования.

Инспектирование состоит в проверке частей проекта (требований, результатов проектирования, программного кода и т. п.) на наличие дефектов. Принцип инспектирования может быть обобщен четырьмя правилами: вскрытие дефектов, участие коллег, распределение ролей, тщательная подготовка.

8. Приведите примеры функциональных и нефункциональных требований к

программному продукту.

Функциональные требования - перечень сервисов, которые должна выполнять система. К ним относятся: бизнес-требования, пользовательские требования, функциональные требования.

Пример:

* Пользователь должен иметь возможность проводить поиск необходимых ему книг и документов или по всему множеству доступных каталожных баз данных или по определенному их подмножеству.
* Система должна предоставлять пользователю подходящее средство просмотра библиотечных документов.
* Каждый заказ должен быть снабжен уникальным идентификатором, который копируется в формуляр пользователя для постоянного хранения.

Нефункциональные требования - описывают характеристики системы и ее окружения, а не поведение системы. К ним относятся: бизнес-правила, внешние интерфейсы, атрибуты качества, ограничения.

9. Что представляют собой варианты использования программного продукта?

Прецеденты (Usecases) — спецификация последовательностей действий, которые может осуществлять система, подсистема или класс, взаимодействуя с внешними действующими лицами. Степень формализма выполняемого проекта и стадия, на которой он находится, прямо влияют на степень детальности и проработанности вариантов использования.

10. Приведите схему процесса анализа С-требований.

Порядок составления C-требований:

1) Идентификация пользователей системы;

2) Проведение интервью с представителями заказчика (определение желаний, документирование требований, эскиз графического интерфейса);

3) Оформление C-требований;

4) Проверка и анализ C-требований.

11. Приведите схему процесса анализа D-требований.

Порядок составления D-требований:

1) Выбор системы организации D-требований;

2) Создание диаграммы последовательности для каждого варианта использования;

3) Получение D-требования из C-требований и от заказчика;

4) Подготовка черновика плана тестирования;

5) Утверждение требований заказчиком;

6) Печать требований.

1. Постановка задачи
   1. Определение целей создаваемого программного продукта

Задачей данной работы является разработка автоматизированной информационной системы предложения антикварного магазина «Шляпа».

Цели:

* повышение эффективности работы пользователя;
* упрощение работы с записями телефонных номеров.

Объект - является отдел оформления заказов антикварного магазина.

Пользователь может выбирать окна с необходимой ему таблицей и получать информацию о наличие товаров и их описание, ФИО клиентов и сотрудников, номер телефона, домашний адрес, должность.

Программная система должна вести нижеуказанную информацию по следующим объектам:

* персонал;
* товар;
* операция занесения товара;
* операция изъятия товара;
* операция купли;
* операция продажи.

На основании обследования предметной области и необходимых запросов выделим следующие сущности с атрибутами (ключевые атрибуты выделены подчеркиванием):

Персонал (фамилия работника, имя работника, отчество работника, должность, оклад, код работника, дата рождения)

Товар (код товара, наименование товара, производитель, номер склада, код работника, адрес склада, телефон склада)

Операция купли (дата покупки, стоимость покупки, наименование фирмы, адрес поставщика, телефон поставщика, код товара)

Операция продажи (номер кассового чека, дата реализации, количество проданного товара, цена товара, фамилия клиента, имя клиента, отчество клиента, адрес клиента, телефон клиента, код клиента, код товара)

Логическая структура базы данных определяется совокупностью логически взаимосвязанных таблиц.. Логические связи таблиц соответствуют структурным связям между объектами. Логическая структура реляционной базы данных приведена на рис.1. На этой схеме реляционные таблицы представлены структурой, определяемой составом и последовательностью полей (атрибутов). Ключи выделены подчеркиванием. Логические связи изображены линиями между одинаковыми ключами связи.

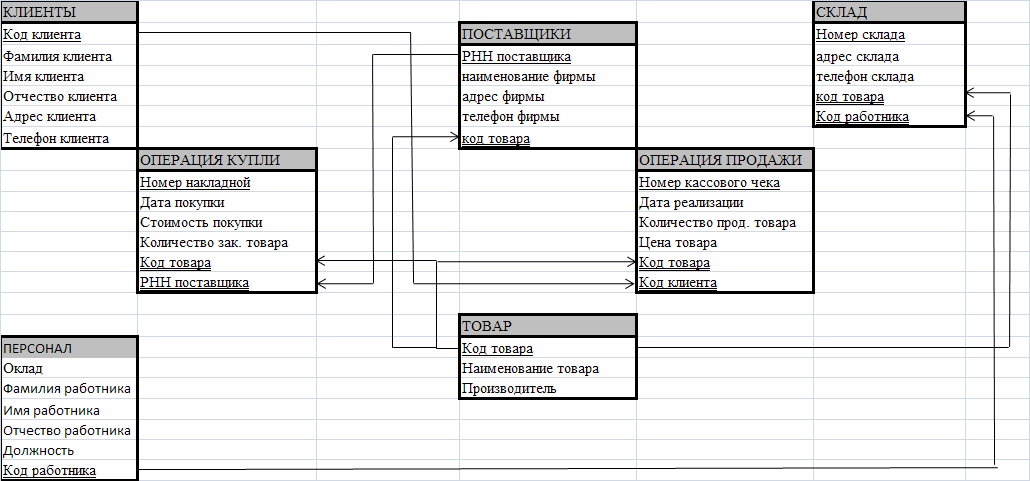


Рисунок 1 - Реляционная модель БД

Объектом является отдел оформления заказов антикварного магазина.

С системой могут работать 2 группы пользователей:

- администраторы;

- пользователи (только персонал).

Пользователь может выбирать окна с необходимой ему таблицей и получать информацию о наличие товаров, их описание и характеристики, ФИО клиентов и сотрудников, номер телефона, домашний адрес, должность.

Администратор и пользователь имеет возможность вносить коррективы в базу данных (добавлять, удалять, менять ассортимент товаров).

Только администратор может вносить новых пользователей программы и редактировать их данные, включая логин и пароль, необходимые для доступа к базе данных.

Функциями ПП администратора является:

* редактирование информации (ФИО, номер телефона, адрес, должность, тип, дата создания, дата продажи и способ оплаты);
* просматривание записей базы;
* добавление новых записей, а также их удаление;
* добавление и редактирование пользователей администратора;
* оформление, удаление и редактирование заказа.
  1. Требования к функциональным характеристикам:

Программа должна обеспечить возможность выполнения следующих функций:

* содержать БД со сведениями;
* обработка запросов;
* занесение пользователей в БД.

Требования к аппаратным интерфейсам

Необходима поддержка сканера ШК, подключаемого к компьютеру через COM-порт.

Требования к программным интерфейсам

Необходимо обеспечить программное взаимодействие системы:

* с системой управления БД SQLite.
* с операционными системами Windows XP/7.
* продукт предусматривает автоматического перехода на платформы семейства Windows XP/7, разрядность 32/64, не перечисленные в данном документе.

Требования к памяти

Нет ограничений кроме аппаратных.

Операции

Необходима поддержка следующих операций:

* вывод на печать.

Требования к адаптации на месте

Необходимы программы-установщики для осуществления развертывания модулей администратора.

Ограничения

* продукт будет поддерживать только русский язык пользовательского интерфейса;
* продукт не ведет учет изменений(кроме операции продажи).

Организация входных и выходных данных

Входные данные поступают с клавиатуры.

Выходные данные отображаются на экране.

Требования к составу и параметрам технических средств.

Система должна работать на IBM-совместимых персональных компьютерах.

Минимальная конфигурация:

тип процессора: Pentium и выше;

объем оперативного запоминающего устройства: 512 Мб и более;

объем свободного места на жестком: диске 40 Мб.

Рекомендуемая конфигурация: тип процессора. Pentium 4;

объем оперативного запоминающего устройства: 1024 Мб;

объем свободного места на жестком Диске: 60 Мб.

На диаграмме вариантов использования изображены список операций, которые выполняет система. На основе набора таких диаграмм создается список требований к системе и определяется множество выполняемых системой функций.

Требования к пользователям продукта:

Администратор:

* понимание принципов функционирования систем управления БД;

Директор:

* владение компьютером на уровне пользователя ОС Windows ;

Клиент:

* владение компьютером на уровне пользователя ОС Windows.

Варианты использования.

Для описания функциональной структуры

При авторизации от Администратора можно работать с пользователями т.е. «Добавлять», «Редактировать», «Удалить». Также за Администратора можно осуществлять просмотр ассортимента т.е «Добавлять», «Редактировать», «Удалить» и просмотри заказов в котором он может осуществить «Поиск»так же «Добавлять», «Редактировать», «Удалить».

При авторизации от Пользователя можно просматривать ассортимент товаров и просмотр заказов «Поиск».

Описание вариантов использования.

Авторизация: пользователь проходит авторизацию заполняя поля логин и пароль. В результате соответствующих манипуляций происходит проверка должности из БД и на её основании меняется функционал программы.

Работа с пользователями: администратор производит редактирование пользователей заполняя все поля в программе. А тем временим она отправляет запрос в БД, с помощью которого производится запись изменений.

Просмотр ассортимента: пользователь осуществляет выбор существующих записей товаров для дальнейшей работы с ними (удаление, добавление или редактирование).

Просмотр заказов: пользователь осуществляет выбор существующих записей заказов для дальнейшей работы с ними (удаление, добавление или редактирование).

Поиск: пользователь производит поиск записей заказов по трем критериям: ФИО, Вид товара, Способ оплаты.

Добавить: пользователь проводит добавление записей, тем самим внося изменения записей в БД.

Редактировать: администратор проводит редактирование записей, тем самим внося изменения записей в БД.

Удаление: администратор проводит удаление записей, тем самим удаляя запись с БД.

Основными прецедентами выбранной подсистемы являются первые три:

* Работа с пользователями;
* Просмотр ассортимента;
* Просмотр заказа.



Рисунок 2 – Диаграмма взаимодействия для варианта использования антикварного магазина «Шляпа»

Примеры вариантов использования:

При открытие БД отображается окно авторизации с логином и паролем.

Если авторизация происходит от user система отображает таблицу «ассортимент».

В таблице ассортимент представлены товары и цены на товары .

Снизу таблицы находятся статичные кнопки «добавить товар в корзину», «удалить товар из корзины», «обновить списки», «пролистать», «сортировать».

При нажатие user-ом на один из товаров в таблице «ассортимент» система отображает окно «Товар» в котором находиться фото товара, информация о товаре, стоимость, информация о продавце . Также с низу информации о товаре отображается кнопка «Оформить заказ».

Если авторизация происходит от администратора система отображает таблицу «Ассортимент» в которой внесены типы товаров их характеристики и стоимость. В отличие от user-а у администратора в таблице товаров есть кнопка «Изменить», которая может изменить стоимость товара, информацию о нем или изображение.

Снизу таблицы находятся статичные кнопки «добавить товар из БД», «удалить товар в БД», «обновить списки», «пролистать», «сортировать».

Сверху таблицы система отображает кнопку «клиенты», «пользователи».

При нажатие на кнопку «клиенты» система отображает окно таблицы клиентов и данные клиентов и купленные ими товары. Также есть кнопка «Изменить» которая меняет данные о клиентах.

При нажатие на кнопку «пользователи» система отображает окно таблицы работников их данные. Также есть кнопка «Изменить» которая меняет данные о работниках.