

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.
2. Что представляют собой артефакты программного продукта?
3. Опишите основные стратегии разработки ПО.
4. Кратко охарактеризуйте системы принципов разработки ПО (PSP, TSP, CMM).
5. Состав персонала разработки программного продукта.
6. Укажите основные метрики качества программного продукта.
7. В чем заключается процесс инспектирования проекта? Укажите принципы инспектирования.
8. Приведите примеры функциональных и нефункциональных требований к программному продукту.
9. Что представляют собой варианты использования программного продукта?
10. Приведите схему процесса анализа С-требований.
11. Приведите схему процесса анализа D-требований.

Рекомендуемые источники.

1. Брауде Э. Технология разработки программного обеспечения. – СПб.: Питер, 2004. – 655с., ил.
 2. Орлов С. Технологии разработки программного обеспечения. – СПб.: Питер, 2002. – 464с.: ил.
 3. Рудаков А.В. Технология разработки программных продуктов : учеб. пособие / А.В.Рудаков. – 2-е изд., стер. – М.: Издательский центр «Академия», 2006. – 208с.
 4. Константайн Л., Локвуд Л. Разработка программного обеспечения. – СПб.: Питер, 2004. – 592с., ил.
 5. Мартин Р. Чистый код: Создание, анализ и рефакторинг. Библиотека программиста. – СПб.: Питер, 2010. – 464с.: ил.
-

Приложение 1.1. Предметная область ПО по вариантам.

1. Информационная система гостиничного комплекса.
2. Информационная система туристического агентства.
3. Информационная система кинотеатра.
4. Информационная система больницы.
5. Информационная система железнодорожного вокзала.
6. Информационная система цветочного магазина.
7. Информационная система аэропорта.
8. Информационная система ресторана.
9. Информационная система мебельной компании.
10. Информационная система библиотеки.
11. Информационная система футбольного клуба.
12. Информационная система заповедника.
13. Информационная система фотоцентра.
14. Информационная система магазина компьютерной техники.
15. Информационная система аптеки.
16. Информационная система таксопарка.
17. Информационная система филармонии.
18. Информационная система клуба знакомств.
19. Информационная система телеканала.
20. Информационная система театра.
21. Информационная система магазина музыкальных инструментов.
22. Информационная система книжного магазина.
23. Информационная система ювелирного магазина.
24. Информационная система магазина канцелярии.
25. Информационная система сауны.
26. Информационная система радиостанции.
27. Информационная система кафедры.
28. Информационная система зоомагазина.
29. Информационная система центра обучения языкам.
30. Информационная система магазина игрушек.

Приложение 1.2. Пример простейшего описания объекта автоматизации.

Примечание.

- 1) Реальные объекты автоматизации традиционно описываются с использованием специальных технологий (IDEF3, ARIS, DFD, UML и т.д.). Если степень сложности объекта невысока, его можно описать в виде простых блок-схем, таблиц и обычного текста.
- 2) Обратите внимание, что здесь формально описываются требования (*что* должно быть сделано), а не аспекты планируемой реализации (*как* должно быть сделано)!

Пример 1 – информационная система факультета.

Программная система должна вести нижеуказанную информацию по следующим объектам:

Студенты	Сотрудники	Ведомости
ФИО	ФИО	Дисциплина
Дата рождения	Должность	Дата сдачи
Ид. номер	Ид. номер	Преподаватель
Номер зачетки	Ученая степень	Пары «Студент : кол-во баллов»
Группа	Стаж	

С системой могут работать 2 группы пользователей:

- администраторы
- пользователи

Обычным пользователям и администраторам должны предоставляться отдельные приложения. Администраторы могут также работать с модулем обработки журнала событий. Это приложение может быть запущено как автономно, так и из приложения администратора. Функциональная схема ПО может быть оформлена в упрощенном виде, как в примере на рис.1.

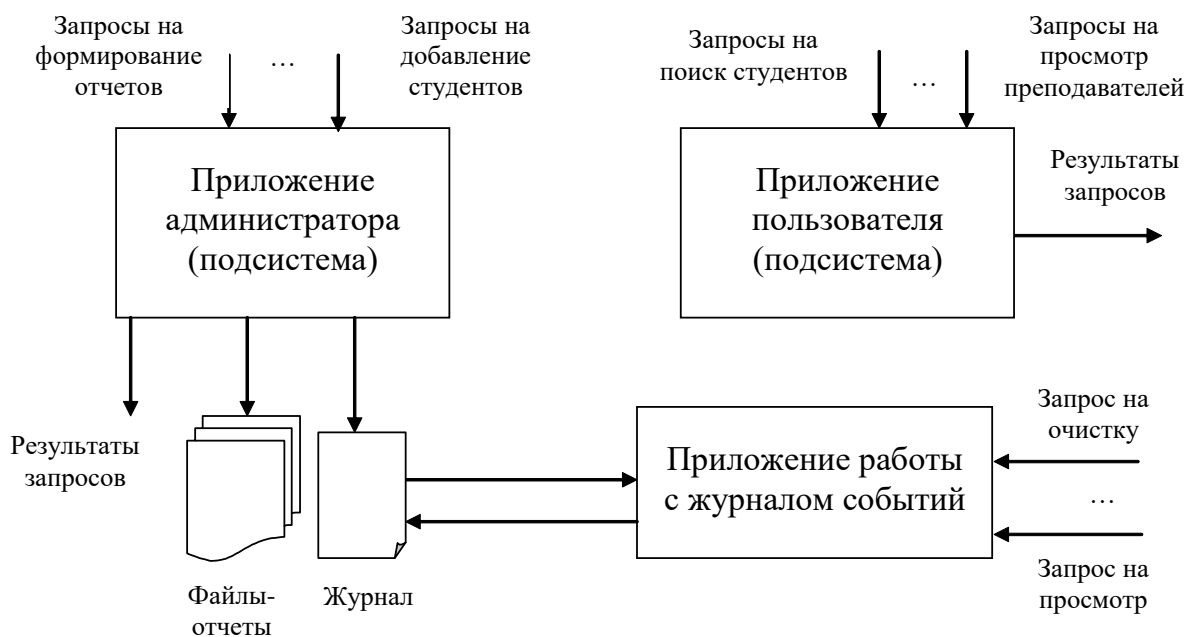


Рисунок 1 – Схема функциональной структуры системы «Факультет»

Пример оформления общих требований к продукту.

Пример 2 – система электронного документооборота.

Требования к пользовательским интерфейсам

Интерфейс должен предполагать стандартную системную цветовую палитру и разрабатываться под разрешение экрана 800х600. Оконные элементы управления и размер шрифта должны быть несколько больше стандартных размеров (оператор будет находиться относительно далеко от монитора). Окна должны обладать системным меню с кнопкой закрытия. Активное текущее окно должно не давать возможности работать с другими окнами программы.

ГЛАВНОЕ ОКНО, в котором должны быть вкладки «Документы», «Сотрудники», «Пропуски». В каждой вкладке отображаются списки и информация по выбранным документам, сотрудникам и пропускам, соответственно. На каждой вкладке должны быть кнопки «Изменить», «Добавить», «Удалить» для редактирования данных. В главном окне должно быть также меню с пунктом «Настройки».

...

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 PostgreSQL 8.4.
- с операционными системами Windows XP/7 и Linux Debian.

Требования к коммуникационным интерфейсам

Необходима поддержка сетевого протокола TCP/IP для обмена сообщений между модулями системы.

Требования к памяти

Программа должна занимать не более 64 Мб оперативной памяти.

Модули программы должны занимать не более 25 Мб памяти на жестком диске.

Операции

Необходима поддержка следующих операций:

- резервное копирование (бекап) БД
- конфигурирование БД и настроек подключения
- сканирование ШК
- отправка уведомлений на адрес электронной почты
- версирование электронных документов (создание ревизий)

Требования к адаптации на месте

Необходимы программы-установщики для осуществления развертывания модулей администратора, архивариуса, актуализаторов и директора на соответствующих целевых компьютерах. Необходимо предоставить всем категориям пользователей справочную информацию.

Требования к пользователям продукта

Администратор:

- Понимание принципов функционирования систем управления БД

Архивариус:

- Доступ к архиву документов
- Навыки работы со сканером ШК

Директор:

- Владение компьютером на уровне пользователя ОС Windows

Клиент:

- Владение компьютером на уровне пользователя ОС Windows

Все пользователи, кроме класса «Клиент», должны быть сотрудниками предприятия.

Ограничения

- Должна использоваться кодировка UTF-8.
- Продукт будет поддерживать только русский язык пользовательского интерфейса.
- Продукт не предусматривает автоматического перехода на платформы, не перечисленные в данном документе.
- Продукт не будет корректно вести историю версий документов, если заказчик будет вручную удалять или изменять имена файлов документов, хранящихся в БД.

Предположения и зависимости

- Скорость передачи изображений будет определяться только состоянием сети.
- Скорость выполнения запросов к БД будет определяться производительностью и степенью загрузки системы управления БД.

Примеры вариантов использования.

Пример 1 – информационная система факультета.

16) ...

17) Вариант использования «Зачислить студента»:

- 1) *Администратор* в окне с информацией о студентах нажимает кнопку «Добавить»
- 2) *Система* отображает окно «Добавление студента» с полями для ввода ФИО, даты рождения, ид.номера, номера зачетной книжки, группы и с кнопками «ОК» и «Отмена».
- 3) *Администратор* в появившемся окне заполняет все поля.
- 4) *Администратор* нажимает кнопку «ОК».
- 5) *Система* закрывает окно «Добавление студента».
- 6) *Система* добавляет в список нового студента и устанавливает выделение на нем.

18) Вариант использования «Сформировать отчет по успеваемости»:

- 1) *Администратор* в окне отчетов, в выпадающем списке с типами отчета, выбирает «Отчет по успеваемости».
- 2) *Администратор* нажимает кнопку «Сформировать».
- 3) *Система* отображает окно с выбором пути для сохранения файла-отчета и 2 кнопками «ОК» и «Отмена».
- 4) *Администратор* указывает путь для сохранения файла-отчета и нажимает кнопку «ОК». Если *Администратор* нажимает кнопку «Отмена», сценарий останавливается.
- 5) *Система* создает текстовый файл с выбранным именем, содержащий отчет (структура отчета дана в приложении С.29).
- 6) *Система* отображает окно с вопросом «Просмотреть отчет?» и 2 кнопками «Да» и «Нет».
- 7) Если *Администратор* нажимает «Да», то *Система* отображает содержимое файла.

19) ...

Примечание.

Степень детализации вариантов использования может быть разной, но требования должны быть полными. В частности, если в разделе SRS «Пользовательский интерфейс» описаны все окна, то указывать их структуру в вариантах использования необязательно.

Пример варианта использования, уточненного в процессе анализа требований.

17) Вариант использования «Зачислить студента»:

- 1) *Администратор* в окне с информацией о студентах нажимает кнопку «Добавить»
- 2) *Система* отображает окно «Добавление студента» с полями для ввода ФИО, даты рождения, ид.номера, номера зачетной книжки, группы и с кнопками «ОК» и «Отмена».
- 3) *Администратор* в появившемся окне заполняет все поля.
- 4) *Администратор* нажимает кнопку «ОК».
- 5) **Выполнить сценарий варианта использования №36 «Проверить корректность данных студента».**
- 6) *Система* отмечает событие добавления студента в журнале событий (логе).
- 7) *Система* закрывает окно «Добавление студента».
- 8) *Система* добавляет в список нового студента и устанавливает выделение на нем.

Примеры детальных функциональных требований.

Ниже приводится пример организации детальных функциональных требований по вариантам использования. Требования должны быть верифицируемыми, т.е. однозначно проверяемыми. Все требования должны быть пронумерованы (промаркированы). Таким образом, при грамотном документировании, в коде будет понятно, выполнение какого конкретного требования он обеспечивает. Например, код обновления списка студентов будет реализовывать выполнение требования «3.2.17.7».

3.2.17. Зачисление студента

Вариант использования	Зачислить студента
Приоритет	<u>Важно</u> / Желательно / Необязательно
Триггер	Нажатие кнопки «Добавить студента» в окне «Студенты»
Предусловие	1. Система должна отобразить администратору <i>модальное</i> окно «Добавление студента».
Основной сценарий	2. Все поля должны быть доступны для редактирования, а кнопки «ОК» и «Отмена» должны быть активными. 3. Система должна проверить корректность заполненных данных (см. <u>3.2.36</u>). 4. Система должна добавлять запись типа «Зачисление студента» в лог. Дата и время должны быть равны системной дате и времени. 5. Система должна закрыть окно «Добавление студента».
Альтернативный сценарий	6. В любой момент администратор может отменить зачисление студента путем нажатия на кнопку «Отмена». Система должна закрыть окно «Добавление студента» и не обновлять список студентов в окне «Студенты».
Постусловие	7. В окне «Студенты» система должна обновить список студентов и установить выделение на добавленном студенте.
Сценарий исключительных ситуаций	8. Система должна генерировать исключение при проблеме с записью в файл лога.
Ссылки	Пункты SRS (ТЗ): 2.3.17 3.1.1.5, 3.2.36

3.2.36. Проверка корректности данных студента

Вариант использования	Проверить корректность данных студента
Приоритет	<u>Важно</u> / Желательно / Необязательно
Триггер	Вариант использования «Добавить студента»
Предусловие	Активно <i>модальное</i> окно «Добавление студента»
Основной сценарий	<ol style="list-style-type: none"> 1. Все поля должны быть заполнены. 2. Значения всех полей должны соответствовать требованиям <u>3.1.3.1</u>. 3. Система должна проверить дублирование ид.номера.
Альтернативный сценарий	<ol style="list-style-type: none"> 4. Если любая из проверок окончилась неудачей, все поля в окне «Добавление студента» очищаются, а остальные проверки не производятся. Администратору выводится окно с сообщением об ошибке (п.<u>3.1.2.1</u>).
Постусловие	-
Сценарий исключительных ситуаций	-
Ссылки	Пункты SRS (ТЗ): 2.3.17 3.1.1.5, 3.1.2.1 Пункты STD (Тесты): 1.36

Примечание. Для данного сценария разрабатываются также тесты.

Примеры детальных нефункциональных требований.

3.1. Нефункциональные требования

3.1.1. Требования к пользовательскому интерфейсу

3.1.1.1. ...

...

3.1.1.5. Окно «Добавление студента»

<картинка> или <текстовое подробное описание>

3.1.2. Требования к текстам сообщений об ошибках

3.1.2.1. Ошибки в данных студента

Ошибка	Текст сообщения
3.2.36.1 (не заполнено поле)	Необходимо заполнить все поля!
3.2.36.2 (значение поля некорректно)	Неверное значение поля <i><поле></i>
3.2.36.3 (дублирование ид.номера)	Такой студент уже присутствует в базе! Идентификационный номер: <i><idN></i> , ФИО: <i><fio></i>

3.1.2.2. ...

3.1.3. Требования к данным

3.1.3.1. Требования к данным о студентах

Данные	Требования
Фамилия	Произвольная строка
Имя	Произвольная строка
Отчество	Произвольная строка
Дата рождения	Строка в формате «dd.mm.yyyy»
Ид. номер	Строка из 10 символов-цифр
Номер зачетки	Произвольная строка
Группа	Произвольная строка

3.1.3.2. ...