

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Ижевский государственный технический университет имени
М. Т. Калашникова»
Кафедра «Программное обеспечение»

Отчет
по лабораторной работе № 2
по дисциплине
«Управление программными проектами»

Выполнили
ст. гр. Б07-191-2

Камашев Р.А.
Худяков Т.А.

Принял

Еланцев М.О.

Ижевск
2015

ТРЕБОВАНИЯ

1. Web-приложение

- 1.1. Интерфейс преподавателя Преподавателю доступно:
 - 1.1.1. Сведения об успеваемости учеников
 - 1.1.2. Расписание занятий
 - 1.1.3. Возможность изменения расписания занятий
- 1.2. Интерфейс администратора
 - 1.2.1. Включает все возможности преподавателя
 - 1.2.2. Возможность добавления новых клиентов/преподавателей
 - 1.2.3. Возможность получения контактных данных клиентов/преподавателей
 - 1.2.4. Возможность получения сведений об оплате
 - 1.2.5. Возможность генерации отчетов/документов

2. Регистрация и учет клиентов, преподавателей

- 2.1. Клиент - физическое лицо, получающее услугу. В системе представлен в следующем виде:
 - 2.1.1. Объект класса Client, содержащий поля:
 - 2.1.2. Идентификатор
 - 2.1.3. ФИО
 - 2.1.4. Статус
 - 2.1.5. Телефон
 - 2.1.6. Группа
 - 2.1.7. Номер паспорта
 - 2.1.8. Серия паспорта
 - 2.1.9. Дата рождения
 - 2.1.10. Адрес регистрации
- 2.2. Клиенты объединяются в группы. Группировка по времени набора и по категории.
- 2.3. Преподаватель - сотрудник автошколы. В системе представлен в следующем виде:
 - 2.3.1. Объект класса Employee, содержащий поля:
 - 2.3.2. Идентификатор
 - 2.3.3. ФИО
 - 2.3.4. Должность
 - 2.3.5. Телефон
 - 2.3.6. Номер паспорта
 - 2.3.7. Серия паспорта
 - 2.3.8. Дата рождения
 - 2.3.9. Адрес регистрации
- 2.4. Регистрация - добавление клиента в систему.
- 2.5. Учет - управление записями о клиентах/преподавателях.

3. Генерация отчетов

- 3.1. Отчет об оплате услуг
- 3.2. Отчет об успеваемости

- 3.3. Формат отчетов: csv
- 4. Денежных операций
 - 4.1. Денежные операции - проведение оплаты за предоставление услуг.
- 5. Информация об услугах
 - 5.1. Теоретическая часть
 - 5.2. Практическая часть
- 6. Генерация документов
 - 6.1. Документы - различные заявки, договора об оказании услуг
 - 6.2. Форматы документов: doc/docx
- 7. Расчет зарплаты сотрудников
- 8. Оповещения
 - 8.1. События для оповещения
 - 8.1.1. Оповещения о будущих занятиях
 - 8.1.2. Оповещения о неуплате
 - 8.1.3. Оповещения о переносе занятий
 - 8.2. Способы оповещения: SMS, Email.
- 9. Статистика
 - 9.1. Динамика успеваемости
 - 9.2. Процент успешно закончивших обучение
 - 9.3. Динамика поступления денежных средств
 - 9.4. Рейтинг преподавателей Формируется на основе:
 - 9.4.1. Отзывы клиентов
 - 9.4.2. Успеваемость групп, закрепленных за этим преподавателем
 - 9.4.3. Процент успешно сдавших экзамен ГИБДД
- 10. Авторизация, аутентификация.
 - 10.1. Аутентификация - проверка подлинности пользователя путём сравнения введённого им пароля с паролем, сохранённым в базе данных пользователей.
 - 10.2. Авторизация - предоставление определённому лицу или группе лиц прав на выполнение определённых действий; а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий.
 - 10.3. Аутентификация осуществляется на основе адреса электронной почты и пароля.
 - 10.4. Авторизация выполняется на основе прав доступа. Права доступа:
 - 10.4.1. Администратор
 - 10.4.2. Преподаватель

ПОДСИСТЕМЫ И МОДУЛИ

Система

1. Web-приложение CRM для автошкол

Подсистемы

1. Пользовательский интерфейс (Web-интерфейс)
2. Работа с БД
3. Работа с клиентами/преподавателями/сотрудниками

Описание сущностей

1. Класс Client, содержащий поля:

- Идентификатор
- ФИО
- Статус
- Телефон
- Группа
- Номер паспорта
- Серия паспорта
- Дата рождения
- Адрес регистрации

2. Класс Employee, содержащий поля:

- Идентификатор
- ФИО
- Должность
- Телефон
- Номер паспорта
- Серия паспорта
- Дата рождения
- Адрес регистрации

Модули

1. Пользовательский интерфейс

1. Модуль вывода клиентов

- Входные данные: идентификатор пользователя (необязательный параметр)
- Выходные данные: html страница, содержащая список клиентов, если не указан идентификатор, иначе вывод конкретного клиента

2. Модуль вывода группы

- Входные данные: идентификатор группы
- Выходные данные: html страница, содержащая список клиентов, относящихся к данной группе

3. Модуль вывода преподавателей

- Входные данные: идентификатор преподавателя (необязательный параметр)
- Выходные данные: html страница, содержащая список преподавателей, если не указан идентификатор, иначе вывод конкретного преподавателя

4. Модуль вывода расписания

- Входные данные: идентификатор преподавателя, период
- Выходные данные: html страница, содержащая расписание на заданный период

2. Работа с БД

1. Модуль аутентификации

- Входные данные: адрес электронной почты, пароль
- Выходные данные: идентификатор сессии в случае успеха, иначе сообщение об ошибке

2. Модуль авторизации

- Входные данные: идентификатор пользователя
- Выходные данные: идентификатор уровня доступа

3. Модуль генерации отчетов

- Входные данные: вид отчета (отчет об оплате услуг/успеваемости), формат отчета
- Выходные данные: файл в запрошенном формате
- 4. Модуль генерации документов
 - Входные данные: вид документа (договор об оказании услуг, заявка в ГИБДД), формат документа
 - Выходные данные: файл в запрошенном формате
- 5. Модуль регистрации клиентов
 - Входные данные: объект класса Client
 - Выходные данные: сообщение об успешном выполнении или сообщение об ошибке
- 6. Модуль регистрации сотрудника
 - Входные данные: объект класса Employee
 - Выходные данные: сообщение об успешном выполнении или сообщение об ошибке
- 7. Модуль проведения денежных операций
 - Входные данные: идентификатор клиента, сумма платежа, тип услуги
 - Выходные данные: сообщение об успешном выполнении или сообщение об ошибке

3. Работа с клиентами/преподавателями/сотрудниками

- 1. Модуль оповещений
 - Входные данные: тип события, способ оповещения, интервал оповещения, текст сообщения
 - Выходные данные: количество отправленных сообщений
- 2. Модуль расчета зарплаты сотрудников
 - Входные данные: идентификатор сотрудника, рейтинг сотрудника, отчет о деятельности
 - Выходные данные: сумма зарплаты
- 3. Модуль расчета рейтинга преподавателя
 - Входные данные: идентификатор преподавателя

- Выходные данные: рейтинг

4. Модуль расчета статистики

- Входные данные: требуемый показатель (динамика успеваемости, процент успешно закончивших обучение, динамика поступления денежных средств, рейтинг преподавателей), период, формат отчета (файл, график)
- Выходные данные: отчет в требуемом формате

Схема системы

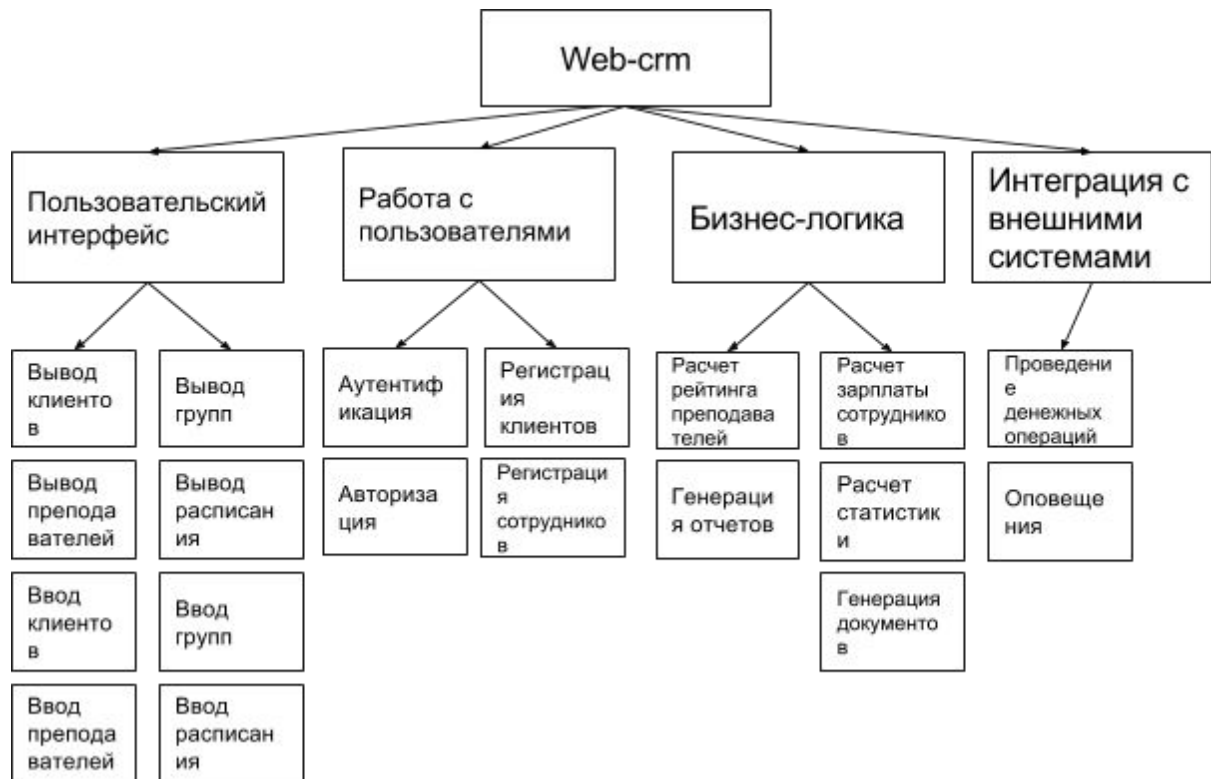


Рис. 1

Содержание

Перечень рисунков

Перечень таблиц

1. Введение
 - 1.1. Назначение системы
 - 1.2. Область применения системы
 - 1.3. Определения, акронимы, аббревиатуры
 - 1.4. Ссылки
 - 1.5. Обзор системы
2. Общее описание системы
 - 2.1. Системный контекст
 - 2.2. Режимы и состояния системы
 - 2.3. Основные функциональные возможности системы
 - 2.4. Основные условия системы
 - 2.5. Основные ограничения системы
 - 2.6. Характеристики пользователя
 - 2.7. Допущения и зависимости
 - 2.8. Оперативные сценарии
3. Функциональные возможности, условия и ограничения системы
 - 3.1. 3.1 Физические
 - 3.1.1. Конструкция
 - 3.1.2. Износостойкость
 - 3.1.3. Адаптируемость
 - 3.1.4. Условия окружающей среды
 - 3.2. Рабочие характеристики системы
 - 3.3. Безопасность системы
 - 3.4. Информационный менеджмент
 - 3.5. Работа системы
 - 3.5.1. Эргономика системы
 - 3.5.2. Ремонтопригодность системы
 - 3.5.3. Надежность системы
 - 3.6. Стратегия и регулирование
 - 3.7. Устойчивость жизненного цикла системы
4. Интерфейсы системы

1. Введение

1.1. Назначение системы

Основное назначение системы - автоматизация учета клиентов и сотрудников автошколы

1.2. Область применения системы

Система применяется в автошколах для автоматизации учета клиентов, сотрудников, сокращения временных затрат на ручное ведение документации. В системе предусмотрены группы пользователей: преподаватель, администратор. Система может использоваться для решения таких задач, как:

- Регистрация нового клиента
- Регистрация нового сотрудника
- Учет успеваемости клиента
- Ведение расписания преподавателя
- Генерация отчетов
- Генерация документов
- Расчет статистики
- Оповещения об изменениях в расписаниях, планируемых событиях
- Расчет рейтинга преподавателей
- Проведение денежных операций

1.3. Определения, акронимы, аббревиатуры

Клиент (Ученик) - физическое лицо, получающее услугу.

Преподаватель - сотрудник автошколы, за которым закреплены ученики (клиенты). Имеет доступ к сведениям об успеваемости своих учеников, к своему расписанию занятий. Имеет возможность изменения расписания.

Администратор - сотрудник автошколы. Имеет все возможности преподавателя. Кроме этого, администратор обладает возможностью добавления новых клиентов/преподавателей, получения информации об оплате обучения клиентами, генерации отчетов/документов, имеет доступ к контактным данным клиентов/преподавателей.

1.4. Ссылки

1.5. Обзор системы

Система реализована как web-приложение. Доступ к приложению имеют только сотрудники автошколы. В зависимости от группы сотрудников (преподаватель, администратор) система предоставляет различные возможности. Тип пользователя определяется при авторизации. Преподавателю доступны возможности редактирования расписания, просмотра расписания, получения сведений об успеваемости учеников. Администратору доступны все возможности преподавателя, кроме того, возможности добавления новых учеников или преподавателей, получения контактных данных учеников или преподавателей, получения сведений об оплате и задолженности, генерации документов или отчетов.

2. Общее описание системы

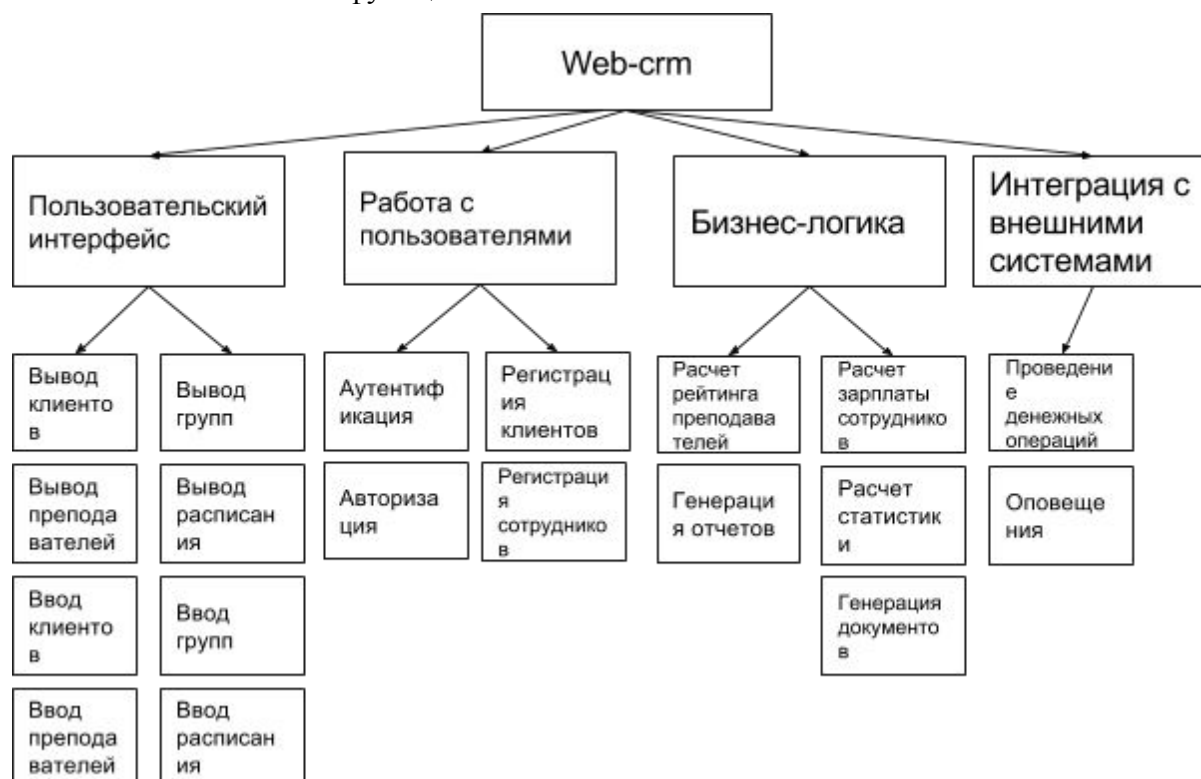
2.1. Системный контекст

2.2. Режимы и состояния системы

Система может находиться в следующих состояниях:

- состояние системы при неавторизованном пользователе;
- состояние системы при авторизованном преподавателе;
- состояние системы при авторизованном администраторе.

2.3. Основные функциональные возможности системы



2.4. Основные условия системы

2.5. Основные ограничения системы

2.6. Характеристики пользователя

Преподаватель - сотрудник автошколы, за которым закреплены ученики (клиенты). Имеет доступ к сведениям об успеваемости своих учеников, к своему расписанию занятий. Имеет возможность изменения расписания. Пользователи

данного типа создаются администратором. Количество пользователей данного типа не ограничено.

Администратор - сотрудник автошколы. Имеет все возможности преподавателя. Кроме этого, администратор обладает возможностью добавления новых клиентов/преподавателей, получения информации об оплате обучения клиентами, генерации отчетов/документов, имеет доступ к контактным данным клиентов/преподавателей. Пользователь данного типа создается при разворачивании системы. Только администратор может выдать права администратора другому пользователю. Пользователь данного типа существует в системе в единственном экземпляре.

2.7. Допущения и зависимости

2.8. Оперативные сценарии

После того как клиент принят в автошколу, заводится учетная запись, связанная с этим клиентом. После этого заполняется вся информация об этом клиенте. Указывается группа и назначаются преподаватели. Теперь после редактирования расписания назначенным преподавателем, клиенту автоматически отправится уведомление о времени занятия. Преподаватель может отследить динамику успеваемости ученика. По запросу ученика администратор может сгенерировать документы, необходимые для обращения в ГИБДД.

3. Функциональные возможности, условия и ограничения системы

3.1. Физические

3.1.1. Конструкция

Система располагается на облачном сервере под управлением ОС, для которой есть реализация виртуальной машины Java (например, Microsoft Windows 7+, Ubuntu 14.04+, Debian 7+).

3.1.2. Износостойкость

Сервер расположен в облаке, поэтому этот параметр нами не контролируется.

3.1.3. Адаптируемость

Так как сервер расположен в облаке, при росте нагрузки произойдет автоматическое масштабирование и система сможет адаптироваться к высокой нагрузке.

3.1.4. Условия окружающей среды

Работа сервера не зависит от условий окружающей среды.

3.2. Рабочие характеристики системы

Критическое количество запросов к системе - 100 запросов в секунду. До превышения критического количества запросов система работает без видимых задержек.

3.3. Безопасность системы

Система защищена от sql-инъекций и xss-атак. При неправильном вводе пароля более 3 раз, нужно ввести код, отправленный в sms на привязанный номер.

3.4. Информационный менеджмент

3.5. Работа системы

3.5.1. Эргономика системы

Требования по распределению функций персонала определены в ТЗ.

3.5.2. Ремонтопригодность системы

3.5.3. Надежность системы

3.6. Стратегия и регулирование

3.7. Устойчивость жизненного цикла системы

4. Интерфейсы системы

Сервис уведомлений

```
public interface NotificationService {  
    void notify(Customer customer, String message, NotificationType type, Duration  
duration);  
    void notify(List<Customer> customers, String message, NotificationType type, Duration  
duration);  
}
```

Сервис оплаты

```
public interface CurrencyService {  
    void transact(Customer customer, BigDecimal sum, BankAccount account);  
}
```

Сервис статистики

```
public interface StatisticsService {  
    StatisticReport generateStatisticRepost(StatisticType type, Period period, String  
format);  
}
```

Сервис подсчета рейтинга

```
public interface RatingService {  
    Double evalRating(TeacherEmployee employee);  
}
```

Сервис расчета ЗП сотрудников

```
public interface SalaryEvaluationService {  
    Double evalSalary(AbstractEmployee employee);  
}
```

Сервис генерации отчетов

```
public interface ReportService {  
    Report generateReport(ReportType type, String format, Integer clientId)  
}
```

Сервис генерации документов

```
public interface DocumentService {  
    String generateDocument();  
}
```

Сервис для работы с сотрудниками

```
public interface EmployeeService {  
    void registerEmployee(AbstractEmployee employee);  
    AbstractEmployee loginEmployee(String phoneNumber, String password);  
    Boolean isLogin(AbstractEmployee abstractEmployee);  
    List<String> getPermission(AbstractEmployee employee);  
    List<AbstractEmployee> findAll();  
    List<AbstractEmployee> findByPosition(String position);  
    Double evaluateEmployeeSalary(AbstractEmployee employee);  
    AbstractEmployee findByPhone(String phoneNumber);  
}
```

Сервис для работы с клиентами

```
public interface CustomerService {  
    void registerCustomer(Customer customer);  
    List<Customer> findAll();  
    List<Customer> findByGroup(String group);  
}
```