

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
Петрозаводский государственный университет
Математический факультет
Кафедра информатики и математического обеспечения

Отчет по дисциплине «Верификация ПО»

Выполнил:
студент 6 курса группы 22608 И. В. Гудач

подпись

Лектор:
к.т.н., доцент К. А. Кулаков
Итоговая оценка:

подпись

Петрозаводск

2014

Содержание

| | | |
|----------|---------------------------------------|-----------|
| 1 | Обзор проекта | 3 |
| 2 | План тестирования | 8 |
| 3 | Варианты тестов | 9 |
| 3.1 | Модульное тестирование | 9 |
| 3.2 | Интеграционное тестирование | 16 |
| 3.3 | Аттестационное тестирование | 17 |
| 4 | Результаты тестирования | 19 |
| 5 | Отчёт об ошибке | 23 |
| 6 | Оценка покрытия тестами | 23 |

1 Обзор проекта

В Петрозаводском государственном университете (ПетрГУ) информация об абитуриентах, студентах и учебной деятельности хранится в базе СУБД Oracle. На этой базе создана информационно-аналитическая интегрированная система (ИАИС). Благодаря этой системе, упрощается процесс управления деятельностью современного ВУЗа, за счёт автоматизации учётных и аналитических функций и информационной поддержки управленческих решений.

ИАИС ПетрГУ включает несколько отдельных компонент, каждая из которых связана с различным направлением деятельности университета, и выполняющая уникальные функции. В работе фигурирует одна базовая компонента, которая отвечает за учебный процесс – «Управление учебной деятельностью».

Цель разработки заключается в том, чтобы помочь студентам в некоторых аспектах их студенческой деятельности, а также упрощение взаимодействия профкома и студентов.

Тестироваться будут только серверные части подсистем, написанные на языке PL/SQL, разработанного компанией Oracle, и с использованием web-среды разработки Application Express.

Для аттестационного тестирования:

- Начальная загрузка страницы подсистемы Студент, загрузка других страниц по нажатию на соответствующей вкладке, нажатие на кнопку бронирования
- Начальная загрузка страницы подсистемы Профком, загрузка других страниц по нажатию на соответствующей вкладке, нажатие на кнопку создать, нажатие на кнопку удалить и нажатие на кнопку обновить

UI (user interface) - пользовательский интерфейс, не тестируется, потому что является сторонним объектом

На сервере установлена специальная среда Application Express. На её основе реализованы подсистемы: Студент и Профком, обе подсистемы тестируются, но тестируется сам код подсистем без взаимодействия со средой.

БД (база данных) - хранилище данных, не тестируется, потому что является сторонним объектом

Первая подсистема называется «Студент», её функционал:

- Получение уникального идентификатора пользователя. FUNCTION INIT(USER IN

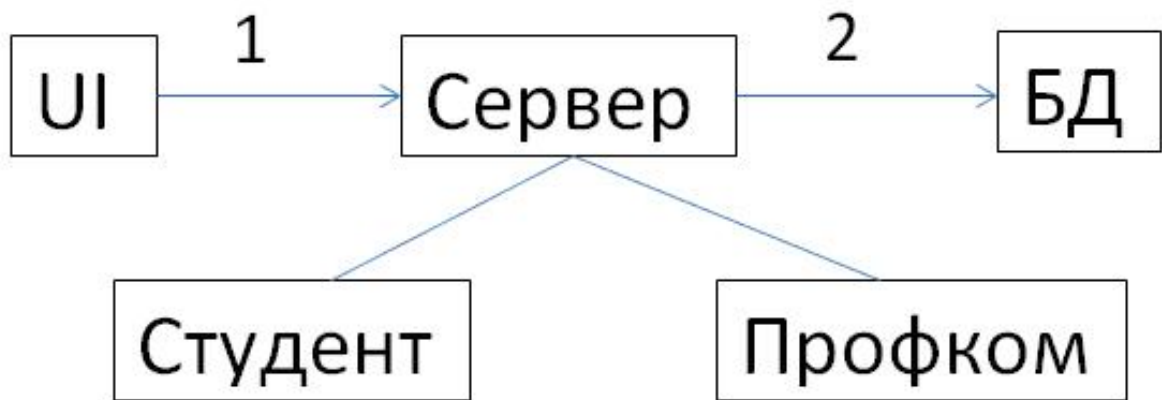


Рис. 1: Архитектура

VARCHAR2) return NUMBER; Принимает логин пользователя и по этому логину определяет уникальный идентификатор, тестируется

- Контактная информация. FUNCTION GETCONTACTINFO(PCARDID IN NUMBER) return VARCHAR2; Принимает уникальный идентификатор студента и возвращает строку запроса, выводящую контактную информацию: адрес студента, телефон, тестируется
- Сведения об образовании. FUNCTION GETEDUCATION(PCARDID IN NUMBER) return VARCHAR2; Принимает уникальный идентификатор студента и возвращает строку запроса, выводящую информацию об образовании: наименование учебного заведения, тип документа, тип образования, возможные медали и год окончания, тестируется
- Льготы. FUNCTION GETBENEFIT(PCARDID IN NUMBER) return VARCHAR2; Принимает уникальный идентификатор студента и возвращает строку запроса, выводящую информацию о льготах: тип льготы и год утверждения, тестируется
- Зачётная книжка. FUNCTION GETGRADEBOOK(PCARDID IN NUMBER) return VARCHAR2; Принимает уникальный идентификатор студента и возвращает строку запроса, выводящую информацию о зачётной книжке: семестр сдачи, дата, название предмета, вид контроля и оценку, тестируется
- Стипендия. FUNCTION GETGRANTS(PCARDID IN NUMBER) return VARCHAR2; Принимает уникальный идентификатор студента и возвращает строку запроса, вы-

водящую информацию о стипендии: даты с какого и по какое выдаётся и тип стипендии, тестируется

- Выплаты, помимо стипендии. FUNCTION GETPAYMENTS(PCARDID IN NUMBER) return VARCHAR2; Принимает уникальный идентификатор студента и возвращает строку запроса, выводящую информацию о выплатах, кроме стипендии: сумма выплат и тип выплат, тестируется
- Профком. FUNCTION PROFCOM(PCARDID IN NUMBER, TRADEUNION IN VARCHAR2, KURS IN NUMBER, FOCODE IN NUMBER, FACULCODE IN VARCHAR2, UPLSPECNAM IN VARCHAR2) return varchar2; Функция возвращает итоговую html-строку, тестируется
- Просмотр всех забронированных мероприятий. FUNCTION BOOKEDTICKETS(PCARDID IN NUMBER, EXC OUT VARCHAR2) return VARCHAR2; Принимает один параметр: ID студента. Функция генерирует список с названиями уже забронированных. Возвращает два параметра: список с названиями и список возникших ошибок, тестируется
- Просмотр всех возможных мероприятий. FUNCTION CHECKPLACES(COUNTPLACES IN NUMBER, GETTERTICKET IN NUMBER, ENDDATE IN DATE, IDNEWS IN NUMBER, EXC OUT VARCHAR2) return VARCHAR2; Принимает количество свободных мест, флаг возможности бронирования мест, дату проведения мероприятия и ID новости. Функция генерирует html строку, выводящую количество свободных мест и кнопку бронирования, если можно забронировать место. Возвращает строку и список возникших ошибок, тестируется
- FUNCTION CHECKTICKETS(IDNEWS IN NUMBER, PCARDID IN NUMBER, ENDDATE IN DATE, KURS IN NUMBER, EXC OUT VARCHAR2) return NUMBER; Принимает четыре параметра: ID новости, ID студента, дату проведения мероприятия и курс студента. Функция проверяет забронированные места в текущем месяце в зависимости от указанного курса. Возвращает два параметра: флаг возможности студентом бронирования мест и список возникших ошибок, тестируется
- Бронирование мест. PROCEDURE BOOK(NEWSID IN NUMBER, PCARDID IN NUMBER, EXC OUT VARCHAR2); Имеет косвенную переменную COUNTPLACES, которая содержит количество свободных мест. Принимает два параметра: ID новости, на

которую происходит бронирование места и ID студента, который бронирует место. Процедура либо бронирует место студенту, либо делает откат бронирования. Возвращает один параметр: список возникших ошибок, тестируется

Вторая подсистема называется «Профком», её функционал:

1. Добавление новостей. PROCEDURE ADDNEWS(HEAD IN VARCHAR2, EXPR IN VARCHAR2, FOCODE IN NUMBER, KURS IN NUMBER, COUNTPLACES IN NUMBER, DATEEND DATE); Процедура добавляет новость, тестируется
2. Редактирование новостей. PROCEDURE UPDATENEWS(IDNEWS IN NUMBER, HEAD IN VARCHAR2, EXPR IN VARCHAR2); Процедура обновляет новость, тестируется
3. Удаление новостей. PROCEDURE DELETENEWS(IDNEWS IN NUMBER); Процедура удаляет новость, тестируется
4. Просмотр новостей. FUNCTION GETNEWS(IDNEWS IN NUMBER) return VARCHAR2; Функция возвращает строку запроса, выводящую новости, тестируется
5. Просмотр студентов, забронировавших места. FUNCTION GETSTUDENTS(IDNEWS IN NUMBER) return VARCHAR2; Функция возвращает строку запроса, выводящую список студентов, забронировавших места для конкретной новости, тестируется

Описание:

- 0, 1, 2, 7 вызываются последовательно в соответствующем порядке при загрузке начальной страницы
- 3, 4, 5 вызываются последовательно, когда была вызвана 2
- 6 вызывается только по нажатию кнопки
- 8, 9, 10, 11 вызываются по нажатию соответствующей вкладки
- все кроме 6 можно вызывать одновременно, открыв все вкладки параллельно
- 0 вызывается с параметром USER, возвращает PCARDID
- 2 вызывается с параметрами PCARDID, TRADEUNION, KURS, FOCODE, FACULCODE, UPLSPECNAME, возвращает html-строку
- 1, 7, 8, 9, 10, 11 вызываются с параметров PCARDID и возвращают sql-строку

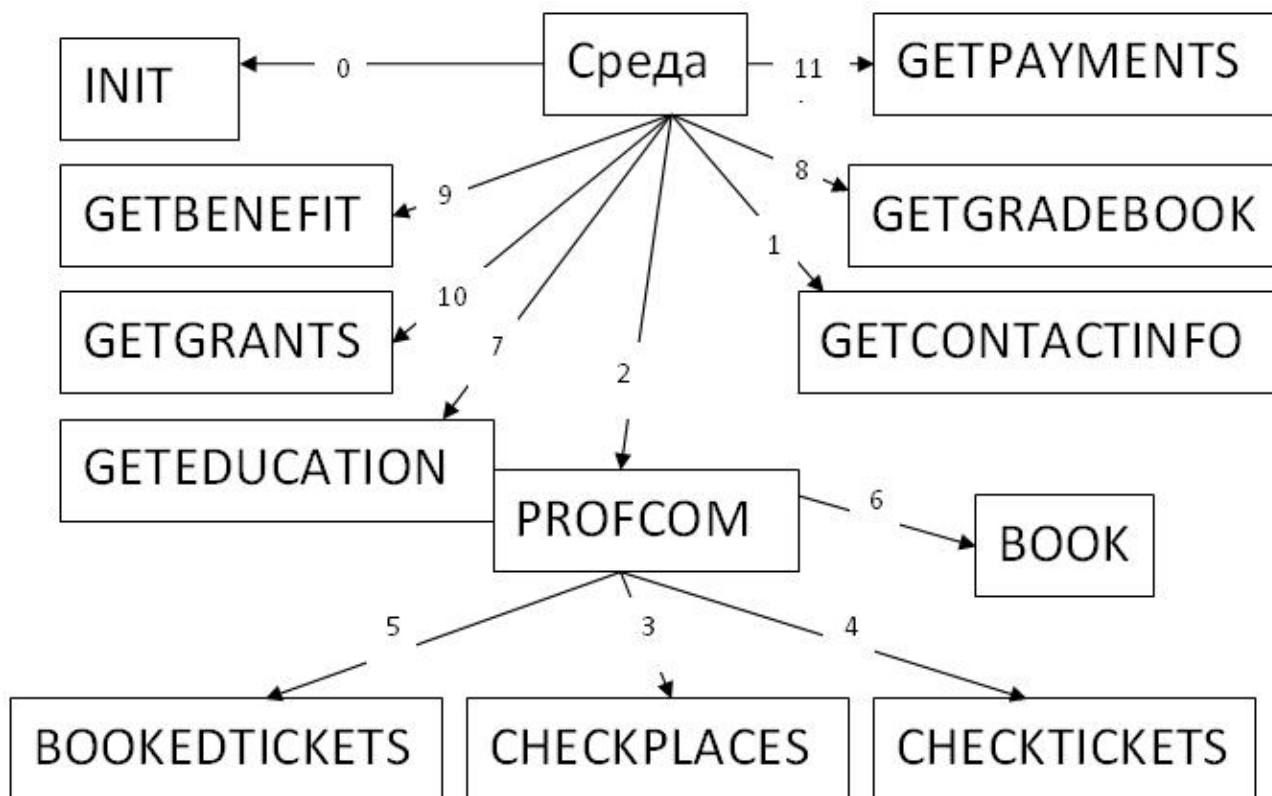


Рис. 2: Схема интеграций подсистемы Студент

- 3 вызывается с параметрами COUNTPLACES, GETTERTICKET, ENDDATE, IDNEWS, а возвращает html-строку
- 4 вызывается с параметрами IDNEWS, PCARDID, ENDDATE, KURS, а возвращает GETTERTICKET
- 5 вызывается с параметром PCARDID, возвращает строку

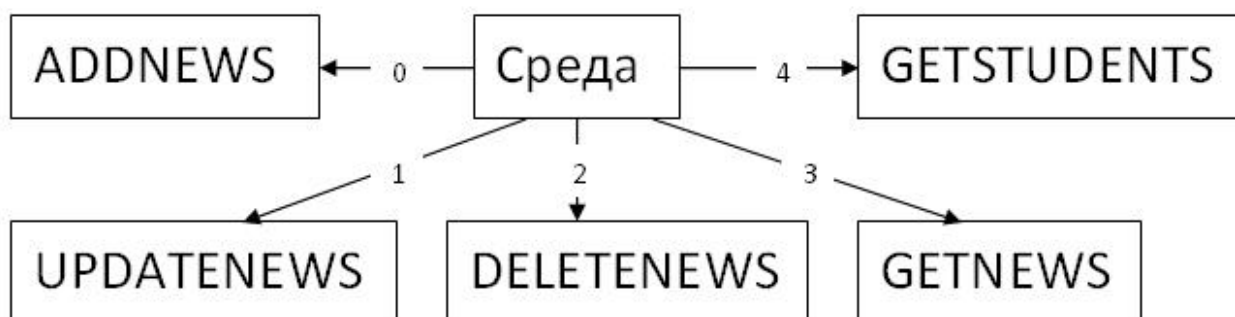


Рис. 3: Схема интеграций подсистемы Профком

Описание:

- 0, 1, 2 вызываются по нажатию на соответствующую кнопку
- 0 вызывается с параметрами HEAD, EXPR, FOCODE, KURS, COUNTPLACES, DATEEND
ничего не возвращает
- 1 вызываются с параметрами NEWSID, HEAD, EXPR, ничего не возвращает
- 2 вызываются с параметром NEWSID, ничего не возвращает
- 3, 4 вызываются последовательно при загрузке страницы
- 3 вызывается без параметров, ничего не возвращает
- 4 вызывается с параметром NEWSID, ничего не возвращает

2 План тестирования

Тестирование производится в среде Oracle Sql Developer с использованием встроенного модуля unit testing. Для каждой подсистемы существует свой пакет с функциями и процедурами.

Блочное тестирование:

Возможные параметры и их описание:

- NEWSID - уникальный идентификатор новости (мероприятия), положительное число
- PCARDID - уникальный идентификатор студента, положительное число
- ENDDATE - дата начала мероприятия, 'день.месяц.год'
- KURS - номер курса, для которого рассчитано мероприятие, положительное число
- COUNTPLACES - количество свободных/не забронированных мест, положительное число
- TRADEUNION - отметка о вступлении в профком, строка Y,N null
- FOCODE - код формы обучения, на которую рассчитано данное мероприятие (по умолчанию = 1, т.е. очная форма), положительное число
- FACULCODE - код факультета, на которое рассчитано данное мероприятие, положительное число

- UPLSPECNAME - название специальности, на которое рассчитано мероприятие, строка 'бцифр.2цифры', '2цифры.2цифры.2-4цифр'

Используемые таблицы или вью:

- VJURSTUD: PCARDID, KURS, TRADEUNION, FOCODE, FACULCODE, UPLSPECNAME
Вью отображает идентификационные данные любого студента.
- GDNEWSSTUD: NEWSID, HEAD, EXPR, DATEEND, TRADEUNION, FOCODE, FACULCODE, KURS. Таблица содержит информацию об мероприятиях.
- GDSTUDLIST: PCARDID, NEWSID. Таблица содержит информацию о том, на какие мероприятия конкретный студент забронировал места.

Интеграционное тестирование:

Тестируется взаимодействие подсистем Профком и Студент между собой, т.е. добавление, редактирование, удаление новостей в подсистеме Профком, и их отображение в подсистеме Студент. Тестируется взаимодействие функций и процедур в подсистеме Студент, т.е. последовательное выполнение функций BOOKEDTICKETS, CHECKTICKETS, CHECKPLACES, которые вызываются из PROFCOM. Тестируется взаимодействие функций и процедур в подсистеме Профком, т.е. редактирование и удаление новости в разных вкладках подсистемы.

Аттестационное тестирование:

Тестируется корректная работа подсистем Профком и Студент, а точнее корректная загрузка web-страниц, переходы между страницами, выполнение действий по нажатию на кнопки.

3 Варианты тестов

3.1 Модульное тестирование

PROCEDURE BOOK(P1NEWSID IN NUMBER, P1PCARDID IN NUMBER, EXC OUT VARCHAR2):

Тип теста: общий

Описание: тест проверяет правильность работы процедуры бронирования мест на мероприятия

Входные данные: положительное число, положительное число

Ожидаемый результат: не произойдёт откат системы и не будет возвращена строка с ошибками

Конкретные тесты:

1. **Входные данные:** P1NEWSID, P1PCARDID есть в БД, COUNTPLACES > 0

Ожидаемый результат: произойдёт коммит транзакций, EXC=""

2. **Входные данные:** COUNTPLACES <= 0

Ожидаемый результат: произойдёт откат транзакций, EXC='откат update'

Тип теста: негативный

Ожидаемый результат: произойдёт откат системы и будет возвращена строка с ошибками

1. **Входные данные:** P1NEWSID, P1PCARDID имеют такие же значения, с которыми ранее функция запускалась успешно

Ожидаемый результат: откат транзакций, EXC='откат insert'

2. **Входные данные:** P1NEWSID нет в БД, PCARDID - есть в БД

Ожидаемый результат: откат транзакций, EXC='откат update'

3. **Входные данные:** P1NEWSID = 10¹²⁶, PCARDID - есть в БД **Ожидаемый результат:** откат транзакций, EXC='??'

```
FUNCTION BOOKEDTICKETS(P1PCARDID IN NUMBER, EXC OUT VARCHAR2)
```

```
return VARCHAR2:
```

Тип теста: общий

Описание: тест проверяет правильность работы функции, которая генерирует для конкретного студента список забронированных мероприятий

Входные данные: положительное число

Ожидаемый результат: возврат строки, содержащей список мероприятий, не произойдёт откат системы и не будет возвращена строка с ошибками

Конкретные тесты:

1. **Входные данные:** P1PCARDID, когда нет забронированных мероприятий

Ожидаемый результат: возврат пустой строки, EXC=""

2. **Входные данные:** P1PCARDID, когда есть забронированные мероприятия

Ожидаемый результат: возврат строки, содержащей список мероприятий, EXC=""

3. **Входные данные:** P1PCARDID, когда есть забронированные мероприятия, но эти мероприятия уже прошли **Ожидаемый результат:** возврат пустой строки, EXC=""

FUNCTION CHECKTICKETS(IDNEWS IN NUMBER, P1PCARDID IN NUMBER, ENDDATE IN DATE, KURS IN NUMBER, EXC OUT VARCHAR2) return NUMBER;

Тип теста: общий

Описание: тест проверяет правильность работы функции, которая возвращает флаг возможности бронирования места на мероприятие

Входные данные: положительное число, положительное число, дата 'день.месяц.год', положительное число

Ожидаемый результат: возврат 0,1 (0-нельзя бронировать,1-можно), не произойдет откат системы и не будет возвращена строка с ошибками

Конкретные тесты:

1. **Входные данные:** IDNEWS, P1PCARDID, - есть в БД, ENDDATE - в этом месяце пользователь не бронировал места, KURS не null

Ожидаемый результат: возврат 1, EXC='NODATAFOUND'

2. **Входные данные:** IDNEWS, P1PCARDID, - есть в БД, ENDDATE - в этом месяце пользователь не бронировал места, KURS null

Ожидаемый результат: возврат 1, EXC='NODATAFOUND'

3. **Входные данные:** IDNEWS, P1PCARDID, - есть в БД, ENDDATE - в этом месяце пользователь бронировал, KURS не null

Ожидаемый результат: возврат 1, EXC=""

4. **Входные данные:** IDNEWS, P1PCARDID, - есть в БД, ENDDATE - в этом месяце пользователь бронировал, KURS null

Ожидаемый результат: возврат 0, EXC=""

FUNCTION CHECKPLACES(COUNTPLACES IN NUMBER, GETTERTICKET IN NUMBER, ENDDATE IN DATE, IDNEWS IN NUMBER, EXC OUT VARCHAR2) return VARCHAR2

Тип теста: общий

Описание: тест проверяет правильность работы функции, которая возвращает html-строку, содержащую либо количество свободных мест и кнопку для бронирования мероприятия, либо уведомление о невозможности бронирования места на это мероприятие

Входные данные: число, число 0,1, дата 'день.месяц.год', положительное число

Ожидаемый результат: произойдёт возврат строки и не будет строки с ошибками

Конкретные тесты:

1. **Входные данные:** COUNTPLACES > 0, GETTERTICKET = 1, ENDDATE, IDNEWS
- существующие в БД

Ожидаемый результат: возврат строки с html вставками, содержащей сведения о количестве оставшихся мест и кнопка для бронирования, EXC=»

2. **Входные данные:** COUNTPLACES = 0, GETTERTICKET = 1, ENDDATE, IDNEWS
- существующие в БД

Ожидаемый результат: возврат строки с html вставками, содержащей сведения о количестве оставшихся мест (а именно 0 мест)

3. **Входные данные:** COUNTPLACES > 0, GETTERTICKET = 0, ENDDATE, IDNEWS
- существующие в БД

Ожидаемый результат: возврат строки с html вставками, содержащей информацию о том, что можно бронировать только на одно мероприятие в месяц

FUNCTION GETCONTACTINFO(PCARDID IN NUMBER) return VARCHAR2; **Тип**

теста: общий

Описание: тест проверяет правильность работы функции, которая возвращает sql-строку

Входные данные: положительное число

Ожидаемый результат: произойдёт возврат строки и не возникнет ошибки

Конкретные тесты:

1. **Входные данные:** PCARDID - существует в БД

Ожидаемый результат: возврат sql-строки

FUNCTION GETEDUCATION(PCARDID IN NUMBER) return VARCHAR2;

Тип теста: общий

Описание: тест проверяет правильность работы функции, которая возвращает sql-строку

Входные данные: положительное число

Ожидаемый результат: произойдёт возврат строки и не возникнет ошибки

Конкретные тесты:

1. **Входные данные:** PCARDID - существует в БД

Ожидаемый результат: возврат sql-строки

FUNCTION GETBENEFIT(PCARDID IN NUMBER) return VARCHAR2;

Тип теста: общий

Описание: тест проверяет правильность работы функции, которая возвращает sql-строку

Входные данные: положительное число

Ожидаемый результат: произойдёт возврат строки и не возникнет ошибки

Конкретные тесты:

1. **Входные данные:** PCARDID - существует в БД

Ожидаемый результат: возврат sql-строки

FUNCTION GETGRADEBOOK(RCARDID IN NUMBER) return VARCHAR2;

Тип теста: общий

Описание: тест проверяет правильность работы функции, которая возвращает sql-строку

Входные данные: положительное число

Ожидаемый результат: произойдёт возврат строки и не возникнет ошибки

Конкретные тесты:

1. **Входные данные:** PCARDID - существует в БД

Ожидаемый результат: возврат sql-строки

FUNCTION GETGRANTS(PCARDID IN NUMBER) return VARCHAR2;

Тип теста: общий

Описание: тест проверяет правильность работы функции, которая возвращает sql-строку

Входные данные: положительное число

Ожидаемый результат: произойдёт возврат строки и не возникнет ошибки

Конкретные тесты:

1. **Входные данные:** PCARDID - существует в БД

Ожидаемый результат: возврат sql-строки

FUNCTION GETPAYMENTS(PCARDID IN NUMBER) return VARCHAR2;

Тип теста: общий

Описание: тест проверяет правильность работы функции, которая возвращает sql-строку

Входные данные: положительное число

Ожидаемый результат: произойдёт возврат строки и не возникнет ошибки

Конкретные тесты:

1. **Входные данные:** PCARDID - существует в БД

Ожидаемый результат: возврат sql-строки

PROCEDURE ADDNEWS(HEAD IN VARCHAR2, EXPR IN VARCHAR, FOCODE IN NUMBER, KURS IN NUMBER, COUNTPLACES IN NUMBER, DATEEND DATE);

Тип теста: общий

Описание: тест проверяет правильность работы функции, которая добавляет новости

Входные данные: строка, строка, положительное число, положительное число, не отрицательное число, дата 'день.месяц.год'

Ожидаемый результат: произойдёт запись кортежа в таблицу

Конкретные тесты:

1. **Входные данные:** HEAD, EXPR, EDUFORM, COUNTPLACES, DATEEND - не null, KURS - null

Ожидаемый результат: запись в БД

2. **Входные данные:** HEAD, EXPR, EDUFORM, KURS, COUNTPLACES, DATEEND - не null

Ожидаемый результат: запись в БД

3. **Входные данные:** HEAD, EXPR, EDUFORM, KURS, DATEEND - не null, COUNTPLACES - null

Ожидаемый результат: запись в БД

PROCEDURE DELETENEWS(IDNEWS);

Тип теста: общий

Описание: тест проверяет правильность работы процедуры, которая по идентификатору удаляет запись в таблице

Входные данные: положительное число

Ожидаемый результат: произойдёт удаление кортежа из таблицы

Конкретные тесты:

1. **Входные данные:** IDNEWS - существует в БД

Ожидаемый результат: удаление кортежа из таблицы

PROCEDURE UPDATENEWS(IDNEWS, HEAD IN VARCHAR2, EXPR IN VARCHAR2);

Тип теста: общий

Описание: тест проверяет правильность работы процедуры, которая по идентификатору обновляет кортеж в таблице

Входные данные: положительное число, строка, строка

Ожидаемый результат: произойдёт замена некоторых значений ячеек кортежа по указанному идентификатору в таблице

Конкретные тесты:

1. **Входные данные:** IDNEWS - существует в БД, HEAD, EXPR - не null

Ожидаемый результат: обновление некоторых значений ячеек кортежа из таблицы

```
FUNCTION GETNEWS(IDNEWS IN NUMBER) return VARCHAR2;
```

Тип теста: общий

Описание: тест проверяет правильность работы функции, которая возвращает sql-строку

Входные данные: положительное число

Ожидаемый результат: произойдёт возврат строки и не возникнет ошибки

Конкретные тесты:

1. **Входные данные:** IDNEWS - существует в БД

Ожидаемый результат: возврат sql-строки

```
FUNCTION GETSTUDENTS(IDNEWS IN NUMBER) return VARCHAR2;
```

Тип теста: общий

Описание: тест проверяет правильность работы функции, которая возвращает sql-строку

Входные данные: положительное число

Ожидаемый результат: произойдёт возврат строки и не возникнет ошибки

Конкретные тесты:

1. **Входные данные:** IDNEWS - существует в БД

Ожидаемый результат: возврат sql-строки

```
FUNCTION INIT(USER IN VARCHAR2) return NUMBER;
```

Тип теста: общий

Описание: тест проверяет правильность работы функции, которая возвращает уникальный идентификатор пользователя

Входные данные: строка

Ожидаемый результат: произойдёт возврат числа

Конкретные тесты:

1. **Входные данные:** USER - существует в БД

Ожидаемый результат: возврат положительного числа

Тип теста: негативный

Описание: тест проверяет правильность работы функций, которая возвращает уникальный идентификатор пользователя

Входные данные: строка

Ожидаемый результат: произойдет возврат числа

Конкретные тесты:

1. **Входные данные:** USER - существует в БД, но пользователь удален из БД
Ожидаемый результат: произойдет ошибка и приложение вылетит с сообщением о том, что точная выборка не возвращает значений
2. **Входные данные:** USER - существует в БД, но пользователь неактивен в БД
Ожидаемый результат: произойдет ошибка и приложение вылетит с сообщением о том, что точная выборка не возвращает значений

3.2 Интеграционное тестирование

```
FUNCTION PROFCOM(PCARDID IN NUMBER, TRADEUNION IN VARCHAR2, KURS  
IN NUMBER, FOCODE IN NUMBER, FACULCODE IN VARCHAR2, UPLSPECNAME IN  
VARCHAR2) return varchar2;
```

Тип теста: интеграционный

Описание: тест проверяет правильность работы функций: BOOKEDTICKETS, CHECKTICKETS, CHECKPLACES - они запускаются последовательно и от результата CHECKTICKETS функция CHECKPLACES будет возвращать различные значения

Входные данные: число, строка, число, число, строка, строка

Ожидаемый результат: произойдет возврат строки и не возникнет ошибки

Конкретные тесты:

1. **Входные данные:** TRADEUNION - null, остальные параметры существуют в БД
Ожидаемый результат: возврат строки: 'Вы не являетесь членом профкома'
2. **Входные данные:** TRADEUNION - 'N', остальные параметры существуют в БД
Ожидаемый результат: возврат строки: 'Вы не являетесь членом профкома'
3. **Входные данные:** TRADEUNION - 'Y', остальные параметры существуют в БД, но в БД на данный момент нет мероприятий
Ожидаемый результат: возврат пустой строки

4. **Входные данные:** TRADEUNION - 'Y', остальные параметры существуют в БД, в БД на данный момент есть мероприятий

Ожидаемый результат: возврат html-строки, содержащей результаты BOOKEDTICKETS, CHECKEDPLACES

1. **Тип теста:** интеграционный

Описание: тест проверяет правильность работы функций: DELETENEWS, UPDATENEWS в двух вкладках открываем одну и ту же новость, в одной удаляем, в другой редактируем

Входные данные: нет

Ожидаемый результат: произойдёт удаление новости и переход на страницу со списком новостей, а при редактировании функция не выполнится, ошибки не возникнет и произойдёт переход на страницу со списком новостей

2. **Тип теста:** интеграционный

Описание: тест проверяет правильность работы функций: BOOK при запуске двумя пользователями, COUNTPLACES > 0

Входные данные: нет

Ожидаемый результат: произойдёт бронирование новости каждым пользователем

3. **Тип теста:** интеграционный

Описание: тест проверяет правильность работы функций: BOOK при запуске двумя пользователями, COUNTPLACES после первого бронирования = 0

Входные данные: нет

Ожидаемый результат: произойдёт бронирование новости первым пользователем, а для второго пользователя произойдёт откат

3.3 Аттестационное тестирование

1. **Тип теста:** аттестационный

Описание: запуск подсистемы Студент

Входные данные: нет

Ожидаемый результат: web-страница отображается корректно и не появляются сообщения об ошибках

2. **Тип теста:** аттестационный

Описание: бронирование места на выбранное мероприятие

Входные данные: нажатие на кнопки забронировать

Ожидаемый результат: происходит вызов функции BOOK, которая корректно работает

3. **Тип теста:** аттестационный

Описание: переход на любую другую страницу приложения

Входные данные: нажатие на вкладку другой страницы

Ожидаемый результат: web-страница отображается корректно и не появляются сообщения об ошибках

4. **Тип теста:** аттестационный

Описание: запуск подсистемы Профком

Входные данные: нет

Ожидаемый результат: web-страница отображается корректно и не появляются сообщения об ошибках

5. **Тип теста:** аттестационный

Описание: создание новости

Входные данные: нажатие на кнопку создать

Ожидаемый результат: переход на web-страницу с формой для заполнения, web-страница отображается корректно и не появляются сообщения об ошибках

6. **Тип теста:** аттестационный

Описание: создание новости

Входные данные: нажатие на кнопку создать на странице с формой заполнения

Ожидаемый результат: переход на главную страницу, вызов функции ADDNEWS, которая работает корректно

7. **Тип теста:** аттестационный

Описание: редактирование новости

Входные данные: нажатие на кнопку изменить

| Дата | Номер теста | Входные данные | Результат | Статус |
|------------------------|-------------|-----------------------------|---|---------|
| 2014-12-30 12:12:39 | 1 | 1, 132605 | ” | Успех |
| 2014-12-30 12:12:39 | 2 | 2, 132606 | 'откат update' | Успех |
| 2014-12-30 12:12:39 | 3 | 2, 132607 | 'откат insert' | Успех |
| 2014-12-30 12:12:39 | 4 | 5, 132606 | ” | Неудача |
| 2015-01-12 12:19:44 | 4 | 5, 132606 | 'откат update' | Успех |
| 2015-01-12 12:19:44 | 5 | 1, 10 ¹²⁶ | 'ORA-06502: PL/SQL: numeric or value error' | Успех |
| 2015-01-12 12:19:44 | 6 | 132606 | ”, ” | Успех |
| 2015-01-12 12:19:44 | 7 | 132607 | '<p>Забронирован билет на: "Хаос"</p>', ” | Успех |
| 2015-01-12 12:19:44 | 8 | 132608 | ”, ” | Успех |
| 2014-12-30 12:15:56 | 9 | 1, 132606, 11.12.2014, 6 | 1, NODATAFOUND | Успех |
| 2014-12-30 12:15:56 | 10 | 1, 132606, 11.12.2014, null | 1, NODATAFOUND | Успех |
| 2014-12-30 12:15:56 | 11 | 1, 132606, 11.12.2014, 6 | 1, ' ' | Успех |
| 2014-12-30 12:15:56 | 12 | 1, 132606, 11.12.2014, null | 0, ” | Успех |
| 2014-12-30 12:15:56 | 13 | 20, 1, 12.11.2014, 1 | html-строка, ” | Успех |
| 2014-12-30 12:15:56 | 14 | 0, 1, 12.11.2014, 1 | html-строка, ' ' | Успех |

| | | | | |
|------------------------|----|---------------------------------------|------------------|-------|
| 2014-12-30 12:15:56 | 15 | 20, 0, 12.11.2014, 1 | html-строка, ' ' | Успех |
| 2014-12-30 12:18:02 | 16 | 132606 | sql-строка | Успех |
| 2014-12-30 12:18:02 | 17 | 132606 | sql-строка | Успех |
| 2014-12-30 12:18:02 | 18 | 132606 | sql-строка | Успех |
| 2014-12-30 12:18:02 | 19 | 132606 | sql-строка | Успех |
| 2014-12-30 12:18:02 | 20 | 132606 | sql-строка | Успех |
| 2014-12-30 12:18:02 | 21 | 132606 | sql-строка | Успех |
| 2014-12-30 12:25:16 | 22 | Хаос, ..., 1, null, 20, 12.11.2014 | запись в БД | Успех |
| 2014-12-30 12:25:16 | 23 | Хаос, ..., 1, 1, 20, 12.11.2014 | запись в БД | Успех |
| 2014-12-30 12:25:16 | 24 | Хаос, ..., 1, 1, null, 12.11.2014 | запись в БД | Успех |
| 2014-12-30 12:25:16 | 25 | 1 | удаление из БД | Успех |
| 2014-12-30 12:25:16 | 26 | 1, Хаос, ... | обновление в БД | Успех |
| 2014-12-30 12:25:16 | 27 | 1 | sql-строка | Успех |
| 2014-12-30 12:25:16 | 28 | 1 | sql-строка | Успех |
| 2014-12-30 12:25:16 | 29 | 1321144 | 132606 | Успех |

| | | | | |
|------------------------|----|--------------------------------------|--------------------------------------|-------|
| 2015-01-12 12:19:44 | 30 | 132606, null, 6, 1, 12, 200000.68 | 'Вы не являетесь членом профкома' | Успех |
| 2015-01-12 12:19:44 | 31 | 132606, N, 6, 1, 12, 200000.68 | 'Вы не являетесь членом профкома' | Успех |
| 2015-01-12 12:19:44 | 32 | 132606, Y, 6, 1, 12, 200000.68 | ” | Успех |
| 2015-01-12 12:19:44 | 33 | 132606, Y, 6, 1, 12, 200000.68 | html-строка | Успех |

5 Отчёт об ошибке

Тест № 4.

Так как приложение является web-приложением и находится на сервере, то оно должно запускаться на любой ОС.

Ожидаемый результат: рядовой пользователь не додумается через консоль вызывать javascript-функции, поэтому ошибки не возникнет

Реальный результат: когда javascript-функция вызывается программно, не происходит никаких проблем, но при вызове её вручную и указании заведомо неверного значения параметра, возникает проблема на серверной части в процедуре BOOK, т.е. можно записать в БД мероприятие с несуществующим идентификатором.

Алгоритм воспроизведения ошибки: запустить javascript-функцию с указанием неправильных значений параметров через консоль браузера

Вариант решения: добавить дополнительную проверку в процедуру BOOK, которая будет отслеживать изменение БД

6 Оценка покрытия тестами

Расчёт тестового покрытия относительно исполняемого кода программного обеспечения проводится по формуле:

$Tcov = (Ltc/Lcode) * 100$ процентов, где:

$Tcov$ - тестовое покрытие

Ltc - количество строк кода, покрытых тестами

$Lcode$ - общее количество строк кода

$Ltc = 325$ физических строк кода (исключая комментарии и пробелы)

$Lcode = 325$ физических строк кода (исключая комментарии и пробелы)

$Tcov = 100$ процентов физических строк кода покрыто тестами