# **Тестирование и интеграция**

# 5.1 Описание процессов тестирования

Будет произведена разработка автоматических тестов. Какие ситуации возникают при работе с сервисом:

1. Авторизация;
2. Получение информации;
3. Запись информации;
4. Перемещение между страницами сервиса.

В зависимости от результатов первого пункта меняются остальные. Поэтому тестировать необходимо:

1. Чтение/запись данных;
2. Перемещение между страницами.

Соответствующие тесты будут произведены от неавторизованного и авторизованного пользователей (которые, в свою очередь, делятся на обычного пользователя и администратора).

Чтение и запись относится к работе с БД в MVC – модели это больше подходит модели, поэтому и тестировать этот функционал будем в модели.

Перемещение отнесем к контроллерам и рендерингу View, поэтому тестировать будем в контроллерах (а точнее в request в контексте контроллера).

*Таблица 1 - Возможности пользователя в зависимости от авторизации.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | View | | | Возможности | | |
| Guest  View | User  View | Admin  View | Изменение денег в банкомате | Снятие наличных | Пополнение счета |
| Guest | + |  |  |  |  |  |
| User |  | + |  |  | + | + |
| Admin |  |  | + | + |  |  |

Тестироваться будет веб-приложение «CuteBank». Определим объекты тестирования:

* Модели
  + «CashMachine»;
  + «User»;
  + «UserHistoryRecord».
* Контроллеры
* «AdminController»;
* «UserController»;
* «MainController».

# 

# 

# 5.2 Тесты

Показать всех пользователей (успешно)

/api/bankUser/accounts/allId

args = ('postgres3', '12345678')

expected :

users and status code 200

response :

[{'username': 'postgres3', 'id': '1'}, {'username': 'postgres4', 'id': '2'}, {'username': 'username1', 'id': '3'}]

show showAllBankUsers status code 200

Показать баланс (успешно)

/api/bankUser/atm/balance{}

args = ('postgres3', '12345678') accID= 1

expected :

balance and status code 200

response :

balance 0.00

show balance status code 200

Депозит (успешно)

/api/bankUser/atm/deposit

args = ('postgres3', '12345678')accID= 1{'accountId': 1, 'atmId': 1, 'toAdd': 20000}

expected : status code 200

response :

deposit status code 200

Показать баланс (успешно)

/api/bankUser/atm/balance{}

args = ('postgres3', '12345678') accID= 1

expected :

balance and status code 200

response :

balance 20000.00

show balance status code 200

Совершить перевод (успешно)

/api/bankUser/atm/transfer

args = ('postgres3', '12345678')accID= 1{'amount': 20000, 'fromAccountId': 1, 'toAccountId': 2}

expected : status code 200

transfer status code 200

Показать баланс (успешно)

/api/bankUser/atm/balance{}

args = ('postgres3', '12345678') accID= 1

expected :

balance and status code 200

response :

balance 0.00

show balance status code 200

(Отрицательный)

/api/bankUser/atm/withdraw

args = ('postgres3', '12345678')accID= 1{'accountId': 1, 'atmId': 1, 'toSub': 10000}

expected : status code 200

withdrw status code 400

Показать баланс (успешно)

/api/bankUser/atm/balance{}

args = ('postgres3', '12345678') accID= 1

expected :

balance and status code 200

response :

balance 0.00

show balance status code 200

Добавить (успешно)

/api/collector/atmCash/add

args = ('collector', '12345678') {'atmId': 1, 'toAdd': 20000}

expected : status code 200

add status code 200

Показать статус (успешно)

/api/collector/atm/allId

args = ('collector', '12345678')

expected : show atms and status code 200

show all status code 200

atms [1,2,3,4]

Удалить статус (успешно)

/api/collector/atm/{}

args = ('collector', '12345678')id- 2

expected : status code 200

delete status code 200

Регистрация (Отрицательный)

/api/reg/bankUser

args = {'password': 'qwerty123', 'username': 'username2'}

expected : status code 200

registration status code 201

Показать банковские счета текущего пользователя (успешно)

/api/bankUser/accounts/allId

args = ('postgres3', '12345678')

expected : ids and status code 200

response :

show bankAccountsOfCurrentUser status code 200

ids [1,2,3,4,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23]

Показать всех пользователей банка (Отрицательный)

/api/bankUser/accounts/allId

args = ('postgres3', '1234567')

expected :

users and status code 200

response :

{'timestamp': '2020-12-02T11:36:25.072+00:00', 'status': 401, 'error': 'Unauthorized', 'message': '', 'path': '/api/user/util/allBankUsers'}

show showAllBankUsers status code 401

Показать баланс (Отрицательный)

/api/bankUser/atm/balance{}

args = ('postgres3', '1234567') accID= None

expected :

balance and status code 200

response :

balance {"timestamp":"2020-12-02T11:36:26.799+00:00","status":401,"error":"Unauthorized","message":"","path":"/api/bankUser/atm/balanceNone"}

show balance status code 401

Перевод статус (Отрицательный)

/api/bankUser/atm/deposit

args = ('postgres3', '1234567')accID= None{'accountId': None, 'atmId': 1, 'toAdd': 20000}

expected : status code 200

response :

deposit status code 401

Показать баланс (Отрицательный)

/api/bankUser/atm/balance{}

args = ('postgres3', '1234567') accID= None

expected :

balance and status code 200

response :

balance {"timestamp":"2020-12-02T11:36:29.314+00:00","status":401,"error":"Unauthorized","message":"","path":"/api/bankUser/atm/balanceNone"}

show balance status code 401

Показать статус перевода (Отрицательный)

/api/bankUser/atm/transfer

args = ('postgres3', '1234567')accID= None{'amount': 20000, 'fromAccountId': None, 'toAccountId': 2}

expected : status code 200

transfer status code 401

Показать баланс (Отрицательный)

/api/bankUser/atm/balance{}

args = ('postgres3', '1234567') accID= None

expected :

balance and status code 200

response :

balance {"timestamp":"2020-12-02T11:36:32.008+00:00","status":401,"error":"Unauthorized","message":"","path":"/api/bankUser/atm/balanceNone"}

show balance status code 401

(Отрицательный)

/api/bankUser/atm/withdraw

args = ('postgres3', '1234567')accID= None{'accountId': None, 'atmId': 1, 'toSub': 10000}

expected : status code 200

withdrw status code 401

Показать баланс (Отрицательный)

/api/bankUser/atm/balance{}

args = ('postgres3', '1234567') accID= None

expected :

balance and status code 200

response :

balance {"timestamp":"2020-12-02T11:36:34.535+00:00","status":401,"error":"Unauthorized","message":"","path":"/api/bankUser/atm/balanceNone"}

show balance status code 401

Добавить (Отрицательный)

/api/collector/atmCash/add

args = ('collector', '1234567') {'atmId': 1, 'toAdd': 20000}

expected : status code 200

add status code 401

Показать атм (Отрицательный)

/api/collector/atm/allId

args = ('collector', '1234567')

expected : show atms and status code 200

show all status code 401

atms {"timestamp":"2020-12-02T11:36:36.225+00:00","status":401,"error":"Unauthorized","message":"","path":"/api/collector/atm/allId"}

Удалить (Отрицательный)

/api/collector/atm/{}

args = ('collector', '1234567')id- 1

expected : status code 200

delete status code 401

Регестрация (Отрицательный)

/api/reg/bankUser

args = {'password': 'qwerty123', 'username': 'username1'}

expected : status code 200

registration status code 409

Показать банковские счета текущего пользователя (Отрицательный)

/api/bankUser/accounts/allId

args = ('postgres3', '1234567')

expected : ids and status code 200

response :

show bankAccountsOfCurrentUser status code 401

ids {"timestamp":"2020-12-02T11:36:38.801+00:00","status":401,"error":"Unauthorized","message":"","path":"/api/bankUser/accounts/allId"}

Показать всех пользователей банка (Успешно)

/api/bankUser/accounts/allId

args = ('postgres3', '12345678')

expected :

users and status code 200

response :

[{'username': 'postgres3', 'id': '1'}, {'username': 'postgres4', 'id': '2'}, {'username': 'username1', 'id': '3'}, {'username': 'username2', 'id': '4'}]

show showAllBankUsers status code 200

Показать баланс (успешно)

/api/bankUser/atm/balance{}

args = ('postgres3', '12345678') accID= 1

expected :

balance and status code 200

response :

balance 0.00

show balance status code 200

Вклад (успешно)

/api/bankUser/atm/deposit

args = ('postgres3', '12345678')accID= 1{'accountId': 1, 'atmId': 1, 'toAdd': -20000}

expected : status code 200

response :

deposit status code 200

Показать баланс (успешно)

/api/bankUser/atm/balance{}

args = ('postgres3', '12345678') accID= 1

expected :

balance and status code 200

response :

balance -20000.00

show balance status code 200

Перевод (успешно)

/api/bankUser/atm/transfer

args = ('postgres3', '12345678')accID= 1{'amount': -20000, 'fromAccountId': 1, 'toAccountId': 2}

expected : status code 200

transfer status code 200

Показать баланс (успешно)

/api/bankUser/atm/balance{}

args = ('postgres3', '12345678') accID= 1

expected :

balance and status code 200

response :

balance 0.00

show balance status code 200

(Успешно)

/api/bankUser/atm/withdraw

args = ('postgres3', '12345678')accID= 1{'accountId': 1, 'atmId': 1, 'toSub': -10000}

expected : status code 200

withdrw status code 200

Показать баланс (Успешно)

/api/bankUser/atm/balance{}

args = ('postgres3', '12345678') accID= 1

expected :

balance and status code 200

response :

balance 10000.00

show balance status code 200

Добавить (Успешно)

/api/collector/atmCash/add

args = ('collector', '12345678') {'atmId': 1, 'toAdd': 20000}

expected : status code 200

add status code 200

/api/collector/atm/allId

Показать все (Успешно)

args = ('collector', '12345678')

expected : show atms and status code 200

show all status code 200

atms [1,2,3,4]

/api/collector/atm/{}

Удалить (Успешно)

args = ('collector', '12345678')id- 3

expected : status code 200

delete status code 200

/api/reg/bankUser

Регистрация (Отрицательный)

args = {'password': 'qwerty123', 'username': 'username1'}

expected : status code 200

registration status code 409

/api/bankUser/accounts/allId

Показать банковские счета текущего пользователя (успешно)

args = ('postgres3', '12345678')

expected : ids and status code 200

response :

show bankAccountsOfCurrentUser status code 200

ids [1,2,3,4,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23]

Общий процент прохождения составил: 58.9 %.

В ходе проведенного тестирования обнаружена и исправлена небольшая неточность. Конечный разработанный продукт, по результатам тестирования, пригоден к эксплуатации. Проблем в логике работы не обнаружено.