

Report.

Task:

Test Scenario Preparation.

1. Boundary Values and Equivalence Classes (EC) Analysis. Select 3 data inputs in your SUT (System Under Test) and perform for them a boundary values and equivalence classes analysis. Use a categorization discussed in the lecture slides. Data inputs can be: a form in user interface, an API interface or a method specification (signature).
2. Input Data Combinations. For the ECs for 3 data inputs selected in the previous point create input data combinations using 2-way uniform strength coverage, mixed-strength coverage and 3-way uniform strength coverage (see lecture slides).
3. Process Testing Design path-based test case scenarios for 3 selected processes in the tested application. Document the processes via a visual tool or Oxygen application. Create the test cases manually or using the Oxygen application. Create the test cases using TDL=1 (edge coverage), TDL=2 (edge-pair coverage) and TDL=3.

Automated Tests - Option 2 - Front-end based tests only

1. CI pipeline. Set-up of CI pipeline which will run the implemented tests
2. Front-end based tests. Create at least 9 front-end based tests
3. Page Objects. Page Object pattern used in at least 2
4. Screenplay pattern used in at least 2 tests
5. Data parametrization. At least 2 tests are using data parametrization (data provider) with external file or configuration class. The parametrization is loaded by test data combinations created in the step "Input Data Combinations" (see section "Test Scenario Preparation" above)

In project have 10 Front-end based tests.

Calculator.

1. WhenChangingCreditSum - Change Amount - Update Monthly Payment [parameterized].

Meaning:

We have two dependent values - creditSumValue and expectedCreditCost. We make sure that when changing creditSumValue, the value of CreditCost coincides with the value of expectedCreditCost, which was calculated in advance. For example, for a Loan Amount of 100000 rub, the Monthly Payment should be displayed of 8143 rub (all other parameters remain by default).

Рассчитать свой кредит

Сумма кредита
100 000
от 100 тыс. до 2 млн. ₽

Срок кредитования
13 месяцев

☒ Я получаю зарплату на карту Ак Барс Банка
☐ Я – сотрудник компании-партнера ?
☐ Я – индивидуальный предприниматель/собственник бизнеса
☒ Со страхованием жизни и здоровья

Ежемесячный платеж **8 143 ₽** Ставка 9.9 % Переплата 5 859 ₽

Расчёт произведен по минимальной процентной ставке. Точные условия по кредиту вы получите после подачи пакета документов.

Оформить кредит

Helper functions used in the test:

- CreditCalculatorPrice (located in the questions folder): the amount of the monthly payment issued by the bank is pulled out.

- Page Object CreditCalculator - page at <https://www.akbars.ru/individuals/credits/potrebitelskiy/>.

Page Object contains:

sumField - loan amount

termField - loan term (manual entry)

termListItem - loan term (drop-down list)

monthCostResult - monthly payment

Task ChangeCreditSum - changes the loan amount and updates the page.

2. **WhenChangingCreditTerm** - Changing the due date - updating the monthly payment [parameterized].

Meaning:

We have two dependent values - the loan term and the monthly payment. testData - the first value is the loan term (by number in the drop-down list), the second value is the calculated value Monthly payment). For example, {0, "8143"} is 13 months and 8143 rub.

We make sure that when the loan term is changed according to the drop-down list, the Monthly payment is correctly changed.

Рассчитать свой кредит

Сумма кредита: 100 000 ₽ (от 100 тыс. до 2 млн.)

Срок кредитования: 2 года

☒ Я получаю зарплату на карту Ак Барс Банка

☐ Я — сотрудник компании-партнера ?

☐ Я — индивидуальный предприниматель/собственник бизнеса

☒ Со страхованием жизни и здоровья

Ежемесячный платеж: **4 609 ₽**

Ставка: 9.9 %

Переплата: 10 616 ₽

Расчёт произведен по минимальной процентной ставке. Точные условия по кредиту вы получите после подачи пакета документов.

Оформить кредит

Helper functions used in the test:

- Page Object.
 - Task ChangeCreditTerm - changes the loan term from the drop-down list and updates the page.
3. **WhenEnteringInvalidCreditSum** - Entering an amount less or more than available - change to available or error message [parameterized].

Meaning:

The boundary values for the Loan amount are from 100000 to 2000000. You cannot enter the wrong value. If you set the wrong value of the loan amount, the value of the monthly payment remains the default (as for the minimum value of the loan amount).

4. **WhenEnteringFullCreditData** - Entering amount equal available [parameterized].

Meaning:

When you enter the correct Loan, amount included in the range, the value of the monthly payment is set.

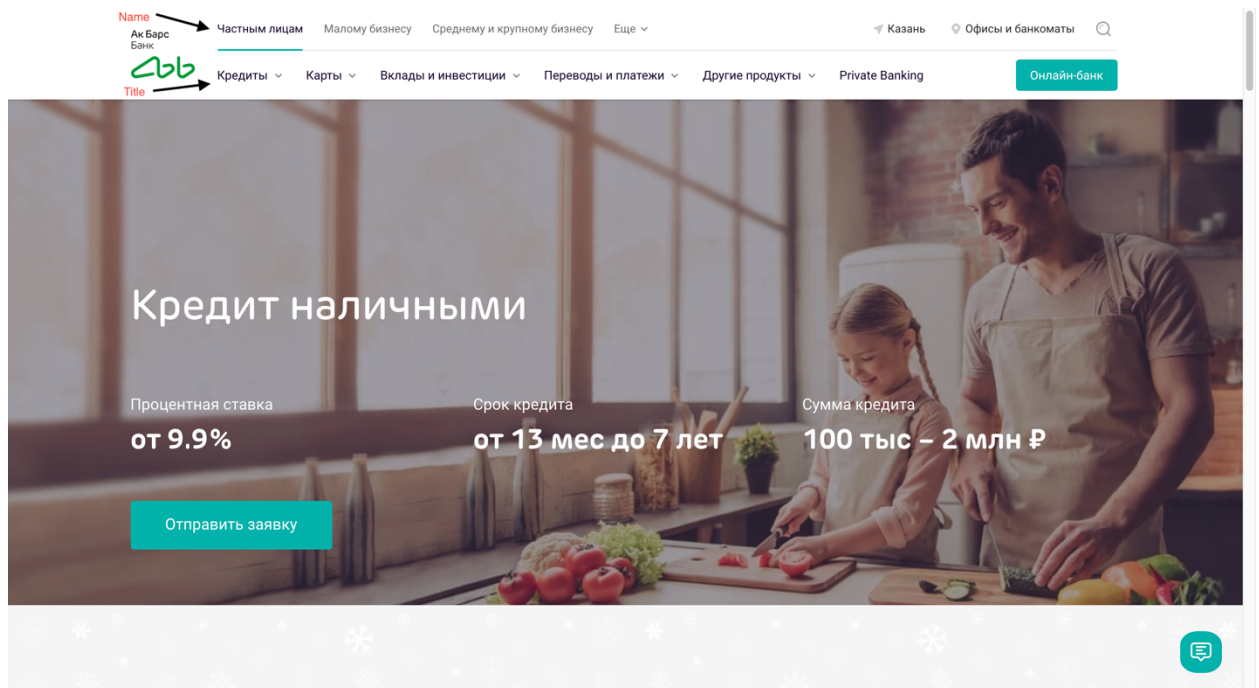
Navigation.

1. **WhenChangingSiteCategory** - Click on the link and check the pageTitle [parameterized].

Meaning:

There is a list of categories (For individuals, Small business, Corporate) of the SiteCategory (models) class, each category has a name and address (name, pageTitle, relativeUrl). When you click on the tab, make sure that we go to the page with the specified pageTitle.

SiteCategory (models): name - the name of the tab, pageTitle - a strip with buttons, relativeUrl - address.



2. **WhenClickingOnLogo** - Click on the bank logo, after which there is a return to the main page [parameterized].

Meaning:

From the list of categories (For individuals, Small business, Corporate), we can go to the main page of the AkBars Bank website by clicking on the bank logo.

3. **WhenClickingOnOnlineBanking** - Click on the online bank - opening the login page

Meaning:

Click on the «Online Banking» button from the «Individuals» tab and make sure that the transition to the address «<https://online.akbars.ru/>» is completed.

Debit Card Application.

1. **WhenEnteringEmptyRequiredValue** - Entering an empty value in the required field - an error appears

Helper functions used in the test:

Page Object CardOrderForm - the page for applying for a new card is located at <https://www.akbars.ru//individuals/cards/generation>.

Контактная информация

Фамилия	Имя	Отчество
---------	-----	----------

Дата рождения

Номер вашего телефона	Электронная почта
-----------------------	-------------------

Task EnterIntoRequiredField - insert in the PageObject in the field the name of an empty value.

Question RequiredFieldErrorMessage - pulls text from an error message.

Фамилия	!

Заполните поле

2. WhenConfirmingCardOrderStepWithValidData Sending - a step with the correct data - go to the next step.

Helper functions used in the test:

- Page Object CardOrderForm.
- Task StartWith - go to the desired page of the site (the main address of the bank is stored in the configuration class + the desired tab).
- Task ConfirmCardFirstStep.withFilledFields () - fills all fields from Page Object with correct data.
- Question SecondStep.isDisplayed () - displays the confirmation code confirmationCode () on the Page Object page.

3. WhenConfirmingCardOrderStepWithInvalidData - Sending a step with incorrect data - there is no transition to the next step.

Helper functions used in the test:

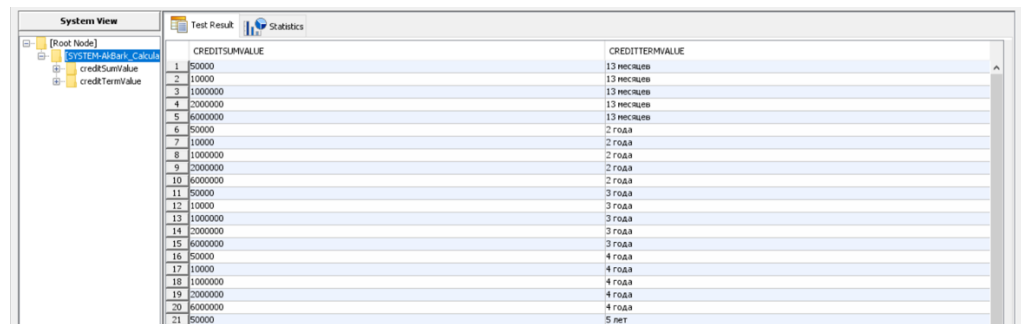
- Page Object CardOrderForm.
- Task StartWith - go to the desired page of the site (the main address of the bank is stored in the configuration class + the desired tab).
- Task ConfirmCardFirstStep.withNotFilledFields () - filling data with a skip Name.
- Fills all fields from Page Object with correct data.
- Question SecondStep.isNotDisplayed () - does not display confirmation code. confirmationCode () on the Page Object page.

CSV files.

1. CreditCalculator

Data inputs: CreditSumValue, CreditTermValue, ExpectedCost

Used in: WhenChangingCreditSum, WhenChangingCreditTerm, WhenEnteringInvalidCreditSum



	CREDITSUMVALUE	CREDITTERMVALUE
1	50000	13 requires
2	10000	13 requires
3	1000000	13 requires
4	2000000	13 requires
5	6000000	13 requires
6	50000	2 roaa
7	10000	2 roaa
8	1000000	2 roaa
9	2000000	2 roaa
10	6000000	2 roaa
11	50000	3 roaa
12	10000	3 roaa
13	1000000	3 roaa
14	2000000	3 roaa
15	6000000	3 roaa
16	50000	4 roaa
17	10000	4 roaa
18	1000000	4 roaa
19	2000000	4 roaa
20	6000000	4 roaa
21	50000	5 net

2. Credit – values.

Data inputs: Credit sum, Credit term, Expected cost.

3. firstNameFieldError

Used in: WhenEnteringEmptyRequiredValue

4. Cardorder

2-way:

correct = SecondStep.isDisplayed ()

incorrect = SecondStep.isNotDisplayed ()

3-way:

mixed is not applicable, since there is no correlation between the input data

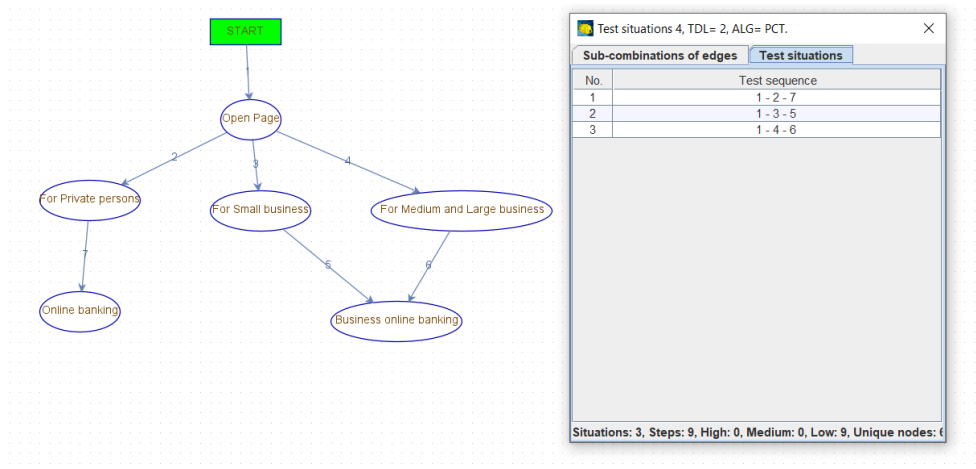
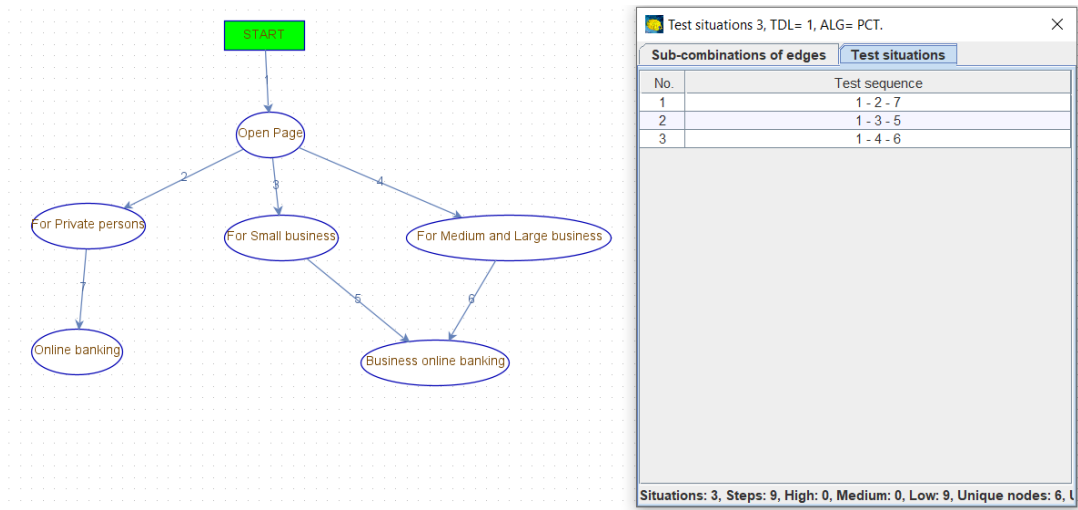
Used in:

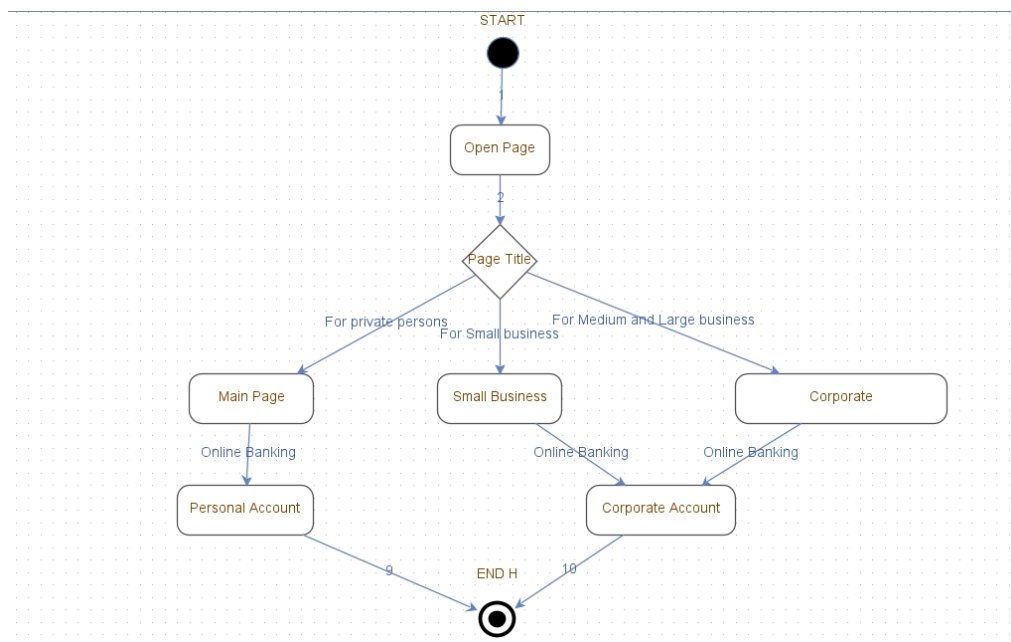
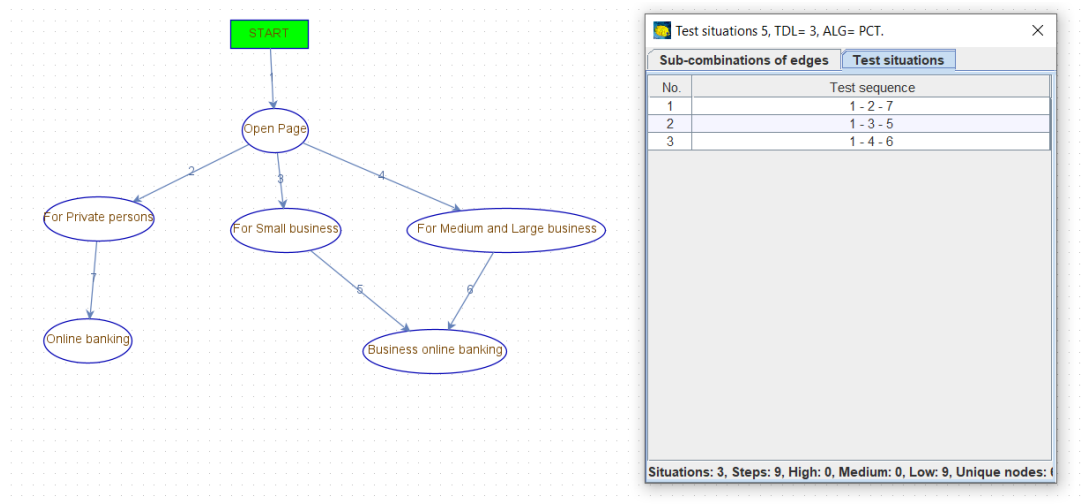
WhenConfirmingCardOrderStepWithInvalidData,
WhenConfirmingCardOrderStepWithValidData

TDL

Test-case 1 – Navigation

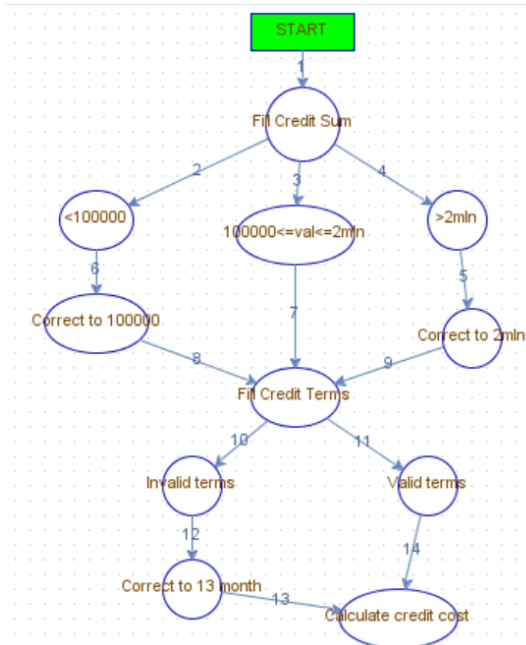
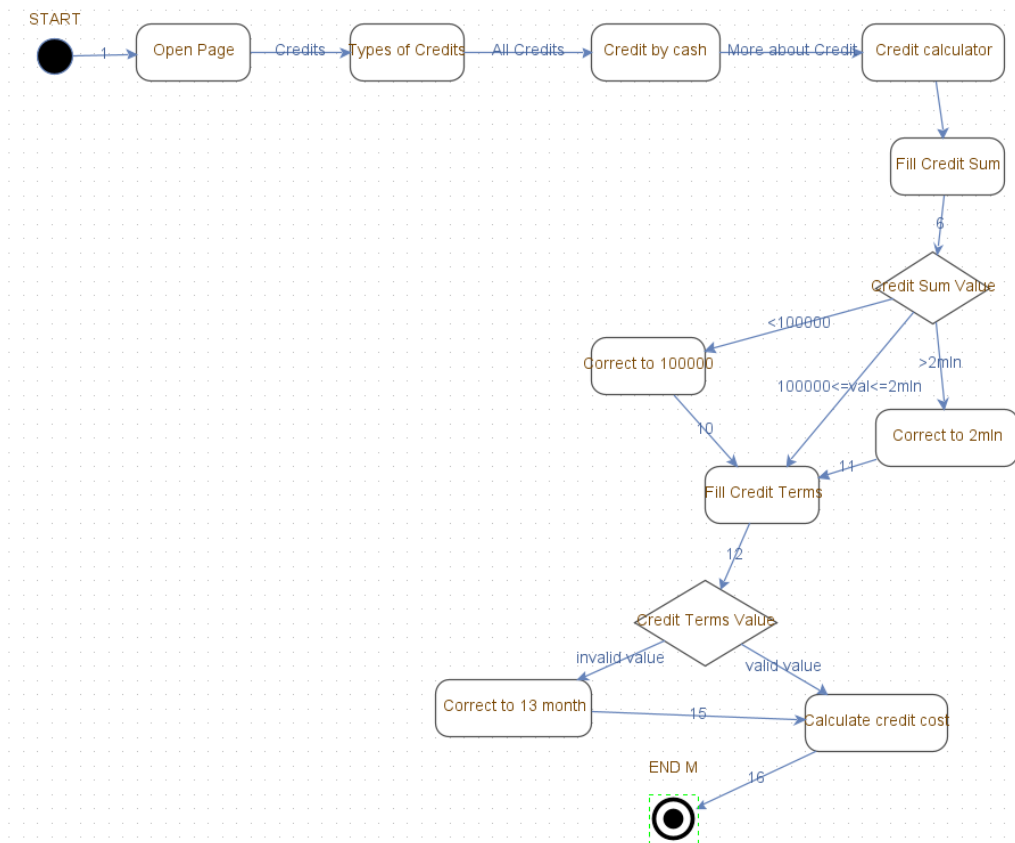
Start - Selecting a tab (For individuals, Small businesses) - clicking on Online Banking.





Test-case 2 - Credit Calculator

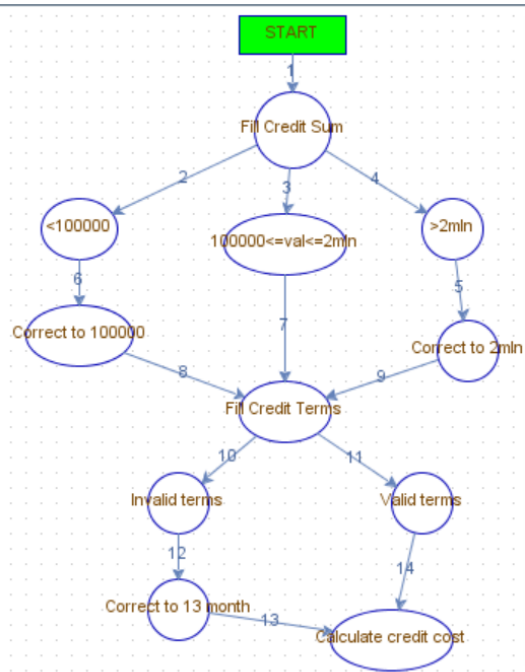
Open page - Loans - All loans - Fill in the loan amount (less than 100,000, in the acceptable interval, more than 2 million - if incorrect, correct by default) - Fill in the loan term (acceptable value, invalid value - if incorrect, correct by default) - Calculate the monthly payment - Open the application form.



Test situations 2, TDL= 1, ALG= PCT.

Sub-combinations of edges		Test situations
No.	Test sequence	
1	1 - 2 - 6 - 8 - 10 - 12 - 13	
2	1 - 3 - 7 - 11 - 14	
3	1 - 4 - 5 - 9 - 10 - 12 - 13	

Situations: 3, Steps: 19, High: 0, Medium: 0, Low: 19, Unique nodes: 11, Unique edges: 19



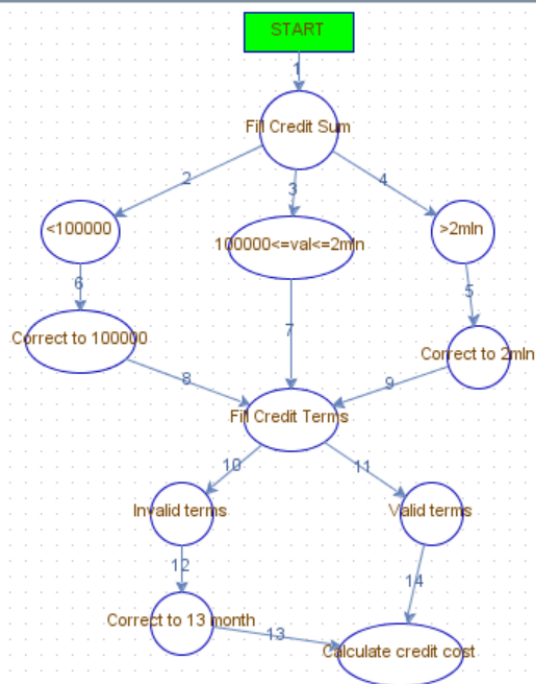
Test situations 4, TDL= 2, ALG= PCT.

Sub-combinations of edges

Test situations

No.	Test sequence
1	1 - 2 - 6 - 8 - 10 - 12 - 13
2	1 - 3 - 7 - 10 - 12 - 13
3	1 - 4 - 5 - 9 - 10 - 12 - 13
4	1 - 3 - 7 - 11 - 14
5	1 - 2 - 6 - 8 - 11 - 14
6	1 - 4 - 5 - 9 - 11 - 14

Situations: 6, Steps: 37, High: 0, Medium: 0, Low: 37, Unique nodes: 11, Unique ed:



Test situations 7, TDL= 3, ALG= PCT.

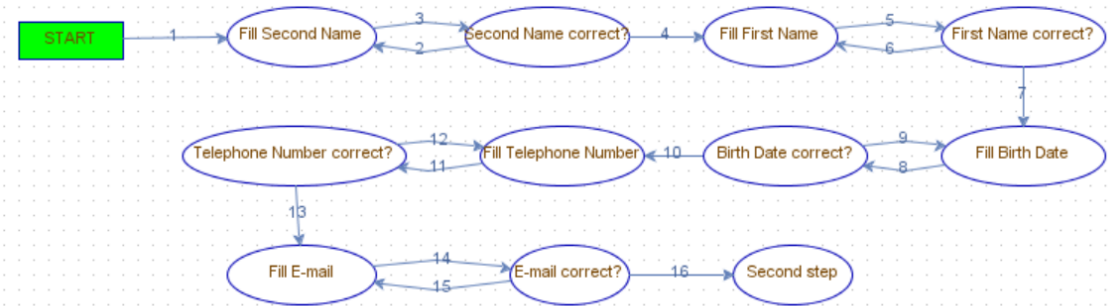
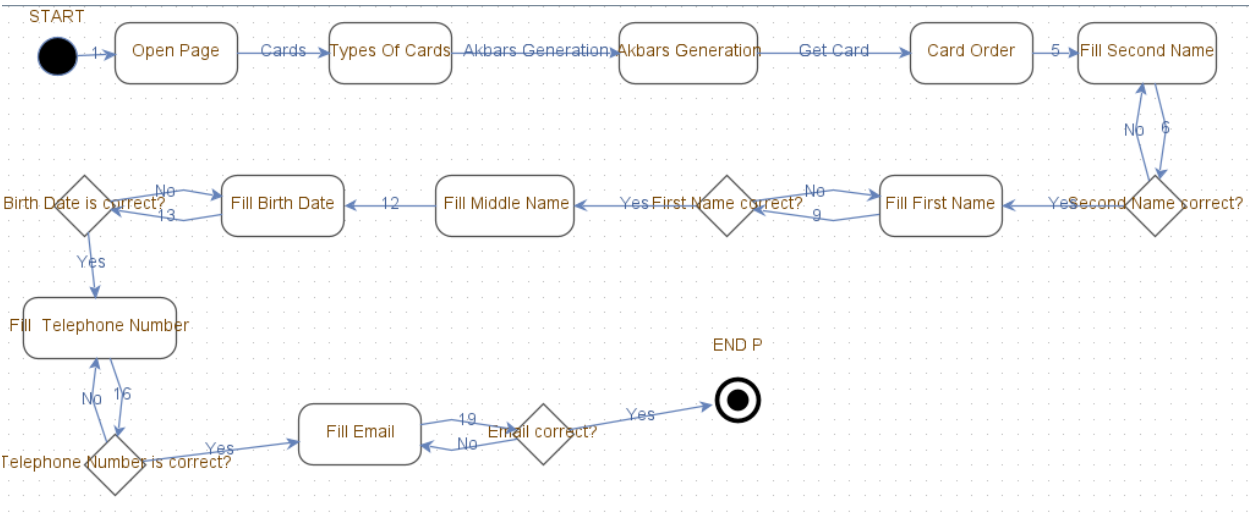
Sub-combinations of edges

Test situations

No.	Test sequence
1	1 - 2 - 6 - 8 - 10 - 12 - 13
2	1 - 3 - 7 - 10 - 12 - 13
3	1 - 4 - 5 - 9 - 10 - 12 - 13
4	1 - 3 - 7 - 11 - 14
5	1 - 2 - 6 - 8 - 11 - 14
6	1 - 4 - 5 - 9 - 11 - 14

Situations: 6, Steps: 37, High: 0, Medium: 0, Low: 37, Unique nodes: 11, Uni

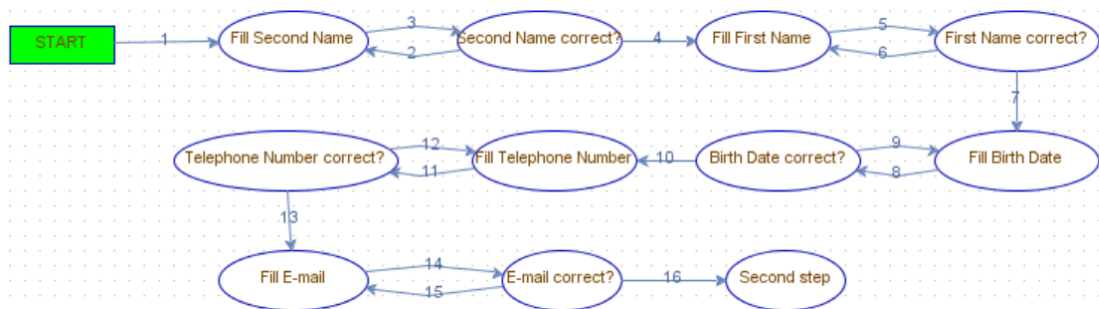
Test-case 3 - Credit Card ordering



Test situations 4, TDL= 1, ALG= PCT.

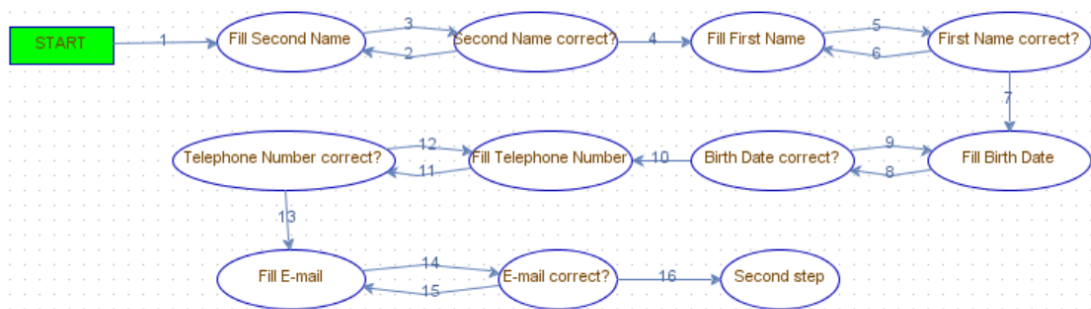
No.	Test sequence
1	1 - 3 - 2 - 3 - 4 - 5 - 6 - 5 - 7 - 8 - 9 - 8 - 10 - 11 - 12 - 11 - 13 - 14 - 15 - 14 - 16

Situations: 1, Steps: 21, High: 0, Medium: 0, Low: 21, Unique nodes: 11, Unique edges: 16



Test situations 10, TDL= 2, ALG= PCT.

Sub-combinations of edges		Test situations
No.	Test sequence	
1	1 - 3 - 2 - 3 - 4 - 5 - 6 - 5 - 7 - 8 - 9 - 8 - 10 - 11 - 12 - 11 - 13 - 14 - 15 - 14 - 16	
Situations: 1, Steps: 21, High: 0, Medium: 0, Low: 21, Unique nodes: 11, Unique edges: 16		



Test situations 12, TDL= 3, ALG= PCT.

Sub-combinations of edges		Test situations
No.	Test sequence	
1	1 - 3 - 2 - 3 - 2 - 3 - 4 - 5 - 6 - 5 - 6 - 5 - 7 - 8 - 9 - 8 - 9 - 8 - 10 - 11 - 12 - 11 - 12 - 11 - 13 - 14 - 15 - 14 - 15 - 14 - 16	
2	1 - 3 - 4 - 5 - 7 - 8 - 10 - 11 - 13 - 14 - 16	
Situations: 2, Steps: 42, High: 0, Medium: 0, Low: 42, Unique nodes: 11, Unique edges: 16		

