

Министерство транспорта Российской Федерации  
Федеральное государственное автономное образовательное  
учреждение высшего образования

«Российский университет транспорта (МИИТ)» (РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

## **Лабораторная работа №4**

### **«Программная инженерия»**

На тему:

«Линейный и разветвляющийся вычислительные процессы»

Выполнил: ст. гр. ТУУ-171

Дудкин А.В.

Вариант №4

Проверил: доц. Сафронов А. И.

Москва – 2025

## Содержание

Цель работы .....	2
Описание задачи.....	2
Содержательная часть .....	4
Технология локального подключения фреймворка Vue.js .....	4
Таблица соответствия переменных .....	7
Параметрическая подстановка.....	10
Методы и события.....	12
Параметрическая связка .....	17
Условная отрисовка/шаблоны .....	17
Вывод.....	24

## Цель работы

Закрепить навыки работы с фреймворком Vue.js. Закрепить навыки работы с параметрической подстановкой (Директива `v-bind`), методами и событиями (Директива `v-on`), параметрической связкой (Директива `v-model`) и с условной отрисовкой шаблонов (Директива `v-if`).

## Описание задачи

Разработать четыре простых локальных одностраничных *web*-приложение (*LSPWA*) под управлением фреймворка *Vue.js* на языке *JavaScript*.

### Параметрическая подстановка (`v-bind`)

Настроить при помощи параметрической подстановки тег рамки (фрейма) `<iframe></iframe>`. Параметрически выводить во фрейм сайт-визитку, созданный в Задании 2. Параметрически убрать границы (`border`) и прокрутку (`scroll`) фрейма. Внутреннее содержимое фрейма параметрически сдвинуть на одинаковую величину в 50 пикселей. Ширину и высоту фрейма принять 300 и 400 пикселей, соответственно.

### Методы и события (`v-on`)

Предусмотреть блок `<div></div>`, в котором по двойному щелчку кнопкой мыши появляется надпись «Мой текст». Двойной щелчок кнопкой мыши при маскировании клавишей «Shift» последовательно реализует смену состояний выравнивания текста в блоке `<div></div>` сначала по центру, затем снова по левому краю.

### **Параметрическая связка (v-model)**

Блок `<div></div>` фиксированных размеров 500 на 500 пикселей по ширине и высоте центрирован в браузерном окне. Предусмотреть изменение посредством нажатия на стрелки курсора на клавиатуре («□», «□», «□», «□») выравнивание текста в блоке `<div></div>` последовательно как по горизонтали (слева, посередине, справа), так и по вертикали (сверху, посередине, снизу). Учесть тот факт, что для выравнивания содержимого по вертикали в блоке `<div></div>` отсутствуют специальные перечислимые значения и параметр управления. Реализовать соответствующее выравнивание следует средствами настройки стилевого оформления тега. Переход через границы блока `<div></div>` не предусматривать. При размещении текста у границ блока – не реагировать на нажатие стрелок курсора на клавиатуре.

### **Условная отрисовка / шаблоны (v-if)**

Подготовить форму авторизации пользователя по логину и паролю. Предусмотреть несколько режимов входа. С параметрами «test» – «test» можно только посмотреть на заложенные в программу Фамилию, Имя, Отчество разработчика, а также название и номер учебной группы, которые по умолчанию не заложены в программу. В режиме «admin» – «admin» иметь возможность отредактировать Фамилию, Имя и Отчество автора. В режиме «user» – «user» иметь возможность ввода названия и номера учебной группы. Иметь возможность выхода к форме авторизации из каждого режима.

## Содержательная часть

### Технология локального подключения фреймворка Vue.js

Технология локального подключения фреймворка Vue.js заключается в переносе содержимого фреймворка Vue.js в локальный файл сценария (.js). Для этого необходимо:

1. Перейти к официальной документации [Vue.js](https://vuejs.org/) и нужно нажать на кнопку Install.
2. Перейти к разделу подключения фреймворка (см. рисунок 1)



Рисунок 1 – Подключение фреймворка через CDN (Content Delivery Network)

3. Перейти по ссылке, по которой подключается фреймворк к приложению (см. рисунок 2)

```

/**
 * vue v3.5.16
 * (c) 2018-present Yuxi (Evan) You and Vue contributors
 * @license MIT
 */
var Vue = (function (exports) {
  'use strict';

  /*! #__NO_SIDE_EFFECTS__ */
  /* __NO_SIDE_EFFECTS__ */
  function makeMap(str) {
    const map = /* @__PURE__ */ Object.create(null);
    for (const key of str.split(",")) map[key] = 1;
    return (val) => val in map;
  }

  const EMPTY_OBJ = Object.freeze({});
  const EMPTY_ARR = Object.freeze([]);
  const NOOP = () => {};
  const NO = () => false;
  const isOn = (key) => key.charCodeAt(0) === 111 && key.charCodeAt(1) === 110 && // uppercase letter
    (key.charCodeAt(2) > 122 || key.charCodeAt(2) < 97);
  const isModelListener = (key) => key.startsWith("onUpdate:");
  const extend = Object.assign;
  const remove = (arr, el) => {
    const i = arr.indexOf(el);
    if (i > -1) {
      arr.splice(i, 1);
    }
  };
  const hasOwnProperty$1 = Object.prototype.hasOwnProperty;
  const hasOwn = (val, key) => hasOwnProperty$1.call(val, key);
  const isArray = Array.isArray;
  const isMap = (val) => toString(val) === "[object Map]";
  const isSet = (val) => toString(val) === "[object Set]";
  const isDate = (val) => toString(val) === "[object Date]";
  const isRegExp = (val) => toString(val) === "[object RegExp]";
  const isFunction = (val) => typeof val === "function";
  const isString = (val) => typeof val === "string";
  const isSymbol = (val) => typeof val === "symbol";
  const isObject = (val) => val !== null && typeof val === "object";
  const isPromise = (val) => {
    return (isObject(val) || isFunction(val)) && isFunction(val.then) && isFunction(val.catch);
  };
  const objectToString = Object.prototype.toString;
  const toString = (val) => objectToString.call(val);

```

Рисунок 2 – Страница с содержимым фреймворка

4. Скопировать содержимое страницы в отдельный файл сценария (см. рисунок 3)

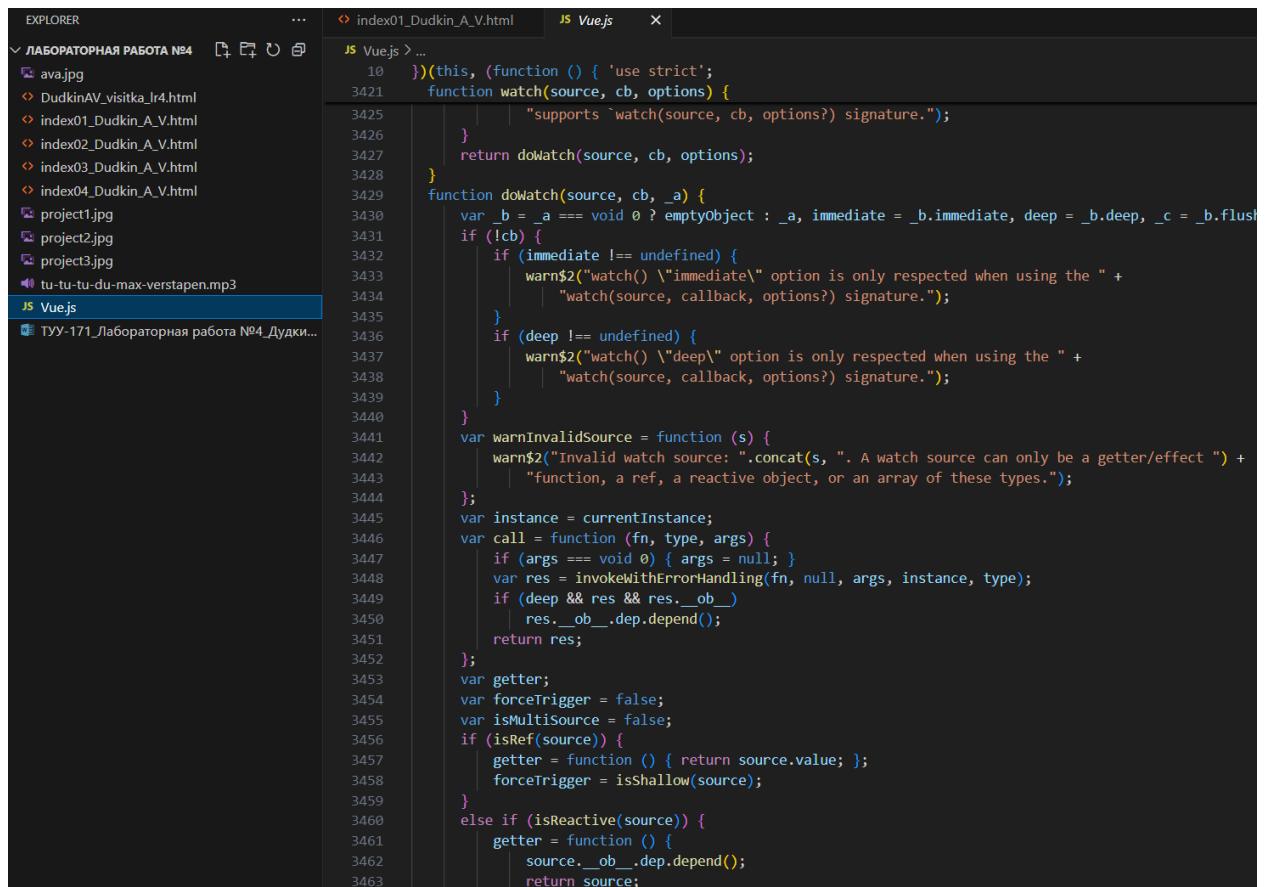


Рисунок 3 – Локальный файл сценария фреймворка Vue.js

- В заголовке разметки подключить файл сценария с фреймворком (см. рисунок 4)



Рисунок 4 – Локальное подключение фреймворка Vue.js

В данном практическом задании был аналогично подключен фреймворк Vue.js 2 версии.

## Таблица соответствия переменных

В таблице 1 представлены переменные, используемые в веб-приложениях.

Таблица 1. Переменные, используемые в веб-приложениях

Веб-приложение	Методы & Переменные	Описание
Параметрическая подстановка	siteVistPath	Строковая переменная, содержащая путь к HTML-файлу визитки (DudkinAV_visitka_lr4.html), который загружается в <iframe>.
	iframeStyle	Объект, содержащий CSS-стили для <iframe>, включая ширину (width), высоту (height), отсутствие границы (border: 'none'), скрывание переполнения (overflow: 'hidden') и смещение (transform: 'translate(50px, 50px)').
Методы и события	displayText	Строковая переменная, хранящая текст, который отображается внутри блока <div>. Изначально пустая, но меняется при двойном клике.
	isCentered	Логический флаг (true/false), определяющий, выровнен ли текст по центру (center) или слева (left). Меняется при двойном клике с зажатой клавишей <b>Shift</b> .
	width	Ширина блока <div> (по умолчанию '300px').
	height	Высота блока <div> (по умолчанию '300px').
	border	Стиль границы блока <div> (по умолчанию '2px solid black').
	textAlign() (вычисляемое свойство)	Срабатывает при двойном клике на блок <div>. Устанавливает displayText = 'Мой текст'. Если клик выполнен с зажатым <b>Shift</b> , переключает isCentered.



Таблица 1. Продолжение

Веб-приложение	Методы & Переменные	Описание
Методы и события	handleDoubleClick(event)	Срабатывает при двойном клике на блок <div>. Устанавливает displayText = 'Мой текст'. Если клик выполнен с зажатым <b>Shift</b> , переключает isCentered.
Параметрическая связка	displayText	Текст, который отображается в div-элементе
	horizontalPosition	Числовая переменная (0-2), определяющая горизонтальное выравнивание текста (0 - left, 1 - center, 2 - right)
	verticalPosition	Числовая переменная (0-2), определяющая вертикальное положение текста (0 - top, 1 - middle, 2 - bottom)
	horizontalAlign	Геттер/сеттер, преобразующий horizontalPosition в CSS-значение (left/center/right)
	verticalPadding	Геттер/сеттер, преобразующий verticalPosition в значение padding-top (0px/200px/400px)
	textStyle	Вычисляемый объект стилей для текста, включающий выравнивание и отступы
	handleKeydown(event)	Метод, обрабатывающий нажатия клавиш стрелок для изменения позиционирования текста

Таблица 1. Продолжение

Веб-приложение	Методы & Переменные	Описание
Условная отрисовка / шаблоны	currentMode	Текущий режим работы приложения: 'auth', 'test', 'admin', 'user'
	login	Логин пользователя (вводится в форме авторизации)
	password	Пароль пользователя (вводится в форме авторизации)
	error	Сообщение об ошибке при неверной авторизации
	lastName	Фамилия разработчика (по умолчанию: "Дудкин")
	firstName	Имя разработчика (по умолчанию: "Александр")
	middleName	Отчество разработчика (по умолчанию: "Владимирович")
	group	Группа разработчика (по умолчанию: "ТУУ-171")
	checkAuth()	Проверяет введенные login и password, переключает currentMode в зависимости от данных
	logout()	Сбрасывает currentMode в 'auth' (выход из аккаунта)

## Параметрическая подстановка

Листинг веб-приложения (задание 1):

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Задание 1</title>
  <script src="Vue.js"></script>
</head>
<body>
  <div id="app">
    <iframe
      :src="siteVistPath"
      :style="iframeStyle">
    </iframe>
    <!--Директиву v-bind можно записать сокращённо как :-->
  </div>

  <script>
    var myVue = new Vue({
      el: '#app',
      data: {
        siteVistPath: 'DudkinAV_visitka_lr4.html',
        iframeStyle: {
          width: '300px',
          height: '400px',
          border: 'none',
          overflow: 'hidden',
          transform: 'translate(50px, 50px)'
        }
      }
    });
  </script>
</body>
</html>
```

На рисунке 5 представлено отображение приложения в браузере.

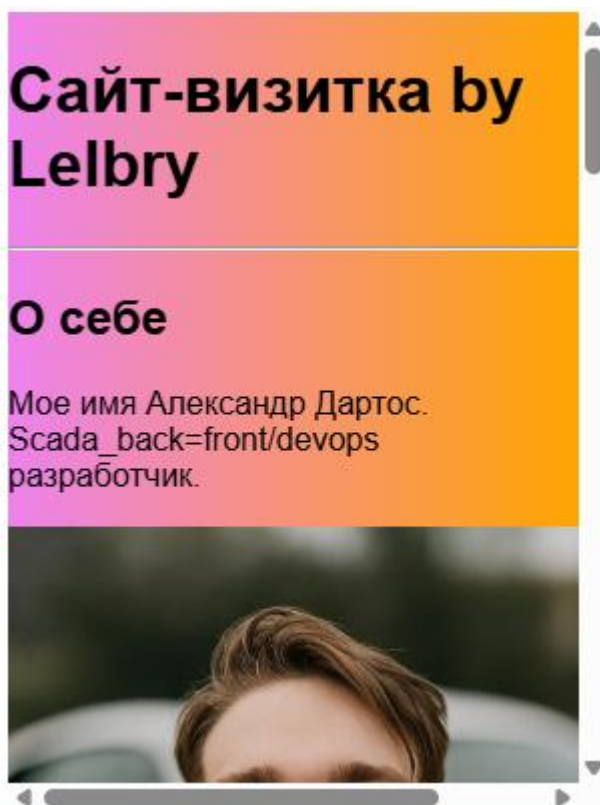


Рисунок 5 – Отображение веб-приложения (задание 1)

BPMN схема представлена на рисунке 6.

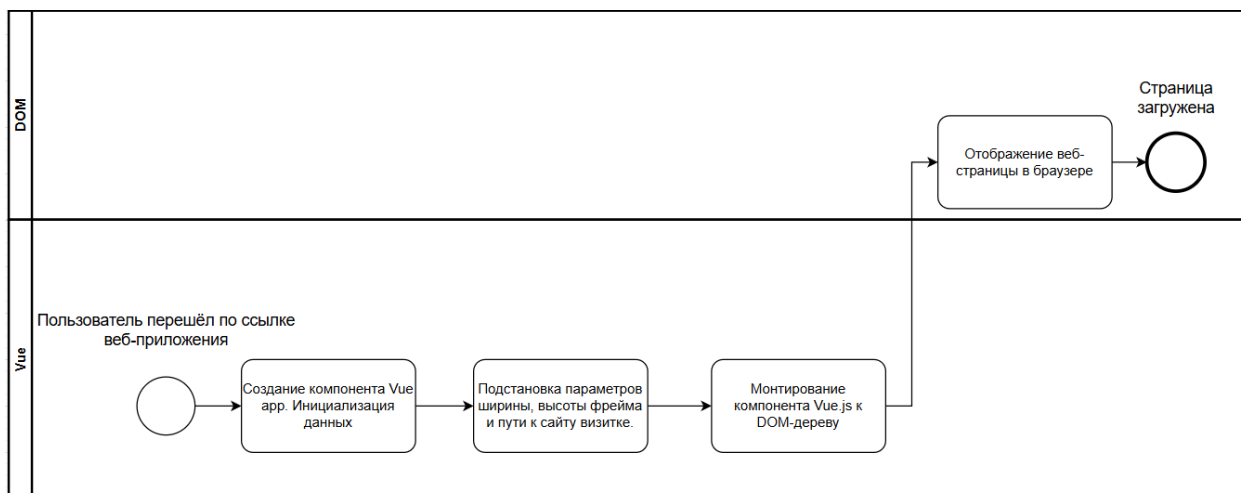


Рисунок 6 – BPMN схема логики работы веб-приложения 1 задания

## Методы и события

Листинг веб-приложения задания 2:

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Задание 2</title>
  <script src="Vue.js"></script>
</head>
<body>
  <div id="app">
    <div :style="{ textAlign: textAlign, width: width, height: height, border: border }"
      @dblclick="handleDoubleClick">
      {{ displayText }}
    </div>
    <!--Директиву v-on можно записать сокращённо как @-->
  </div>

  <script>
    new Vue({
      el: '#app',
      data: {
        displayText: '', // Отображаемый текст
        isCentered: false, // Логический флажок выравнивания текста по центру
        width: '300px', // Ширина блока div
        height: '300px', // Высота блока div
        border: '2px solid black', // Граница для div
      },
      computed: {
        textAlign() {
          return this.isCentered ? 'center' : 'left';
        } // Выравнивание текста
      },
      methods: {
        handleDoubleClick(event) {
          this.displayText = 'Мой текст';

          if (event.shiftKey) {
            this.isCentered = !this.isCentered;
          }
        }
      }
    })
  </script>
</body>
</html>
```

```
    }  
  });  
</script>  
</body>  
</html>
```

На рисунке 7 представлено отображение веб-приложения в браузере в исходном состоянии.

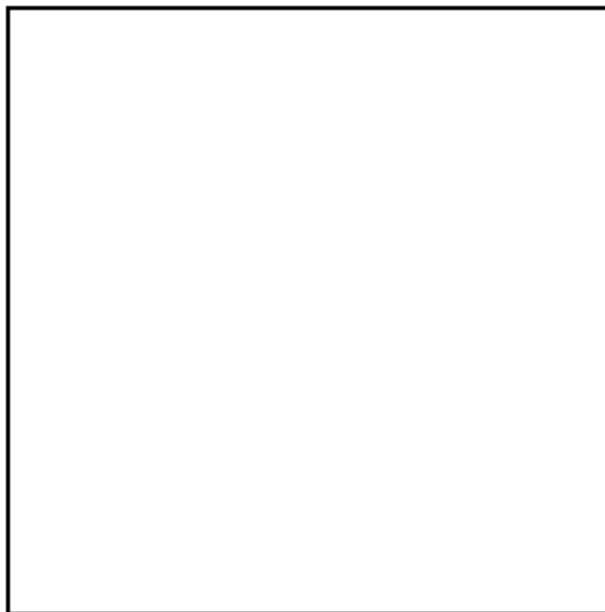


Рисунок 7 – Отображение веб-приложения 2 задания в исходном состоянии

На рисунке 8 представлено отображение веб-приложения в браузере после двойного нажатия мыши по фрейму.

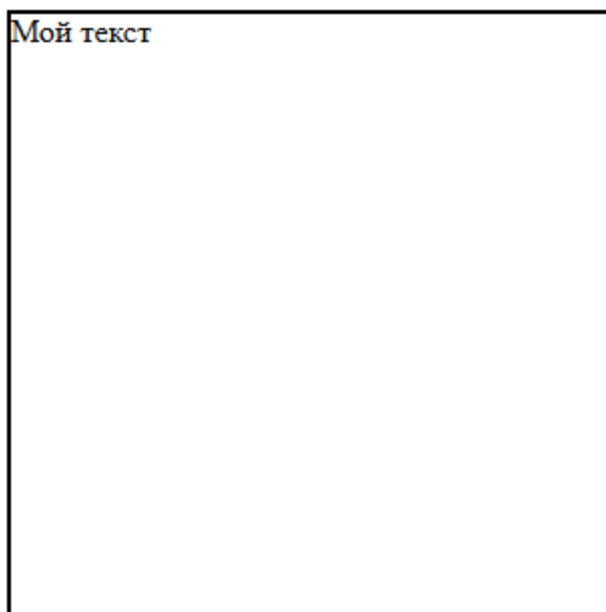


Рисунок 8 – Отображение веб-приложения 2 задания после 2го клика  
мышы

На рисунке 9 представлено отображение веб-приложения shift+2ой клик мышы. Что влечёт за собой смену состояний выравнивания текста сначала по центру, затем снова по левому краю.

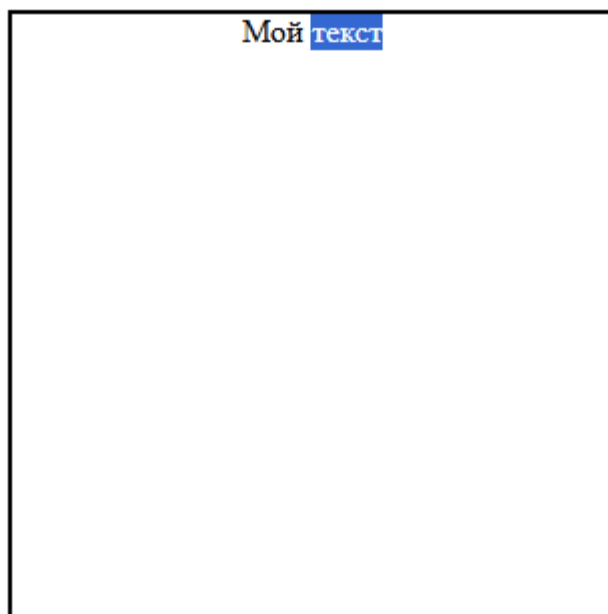


Рисунок 9 – Отображение веб-приложения 2 задания со скрытыми ползунками фрейма

Первый фрагмент BPMN схемы логики работы веб-приложения 2-го задания представлен на рисунке 10.

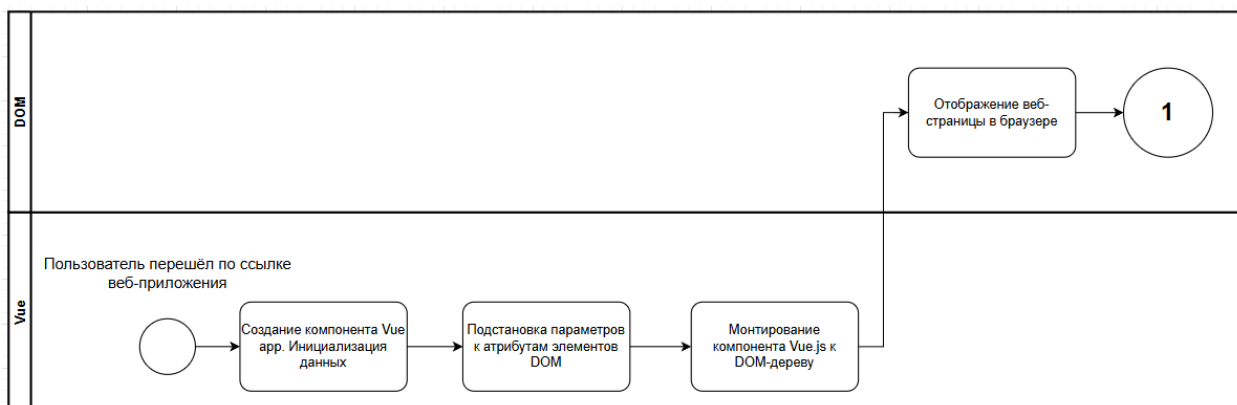


Рисунок 10 – Первый фрагмент BPMN схемы логики работы веб-приложения 2 задания

Второй фрагмент (продолжение) BPMN схемы логики работы веб-приложения 2-го задания представлен на рисунке 11.



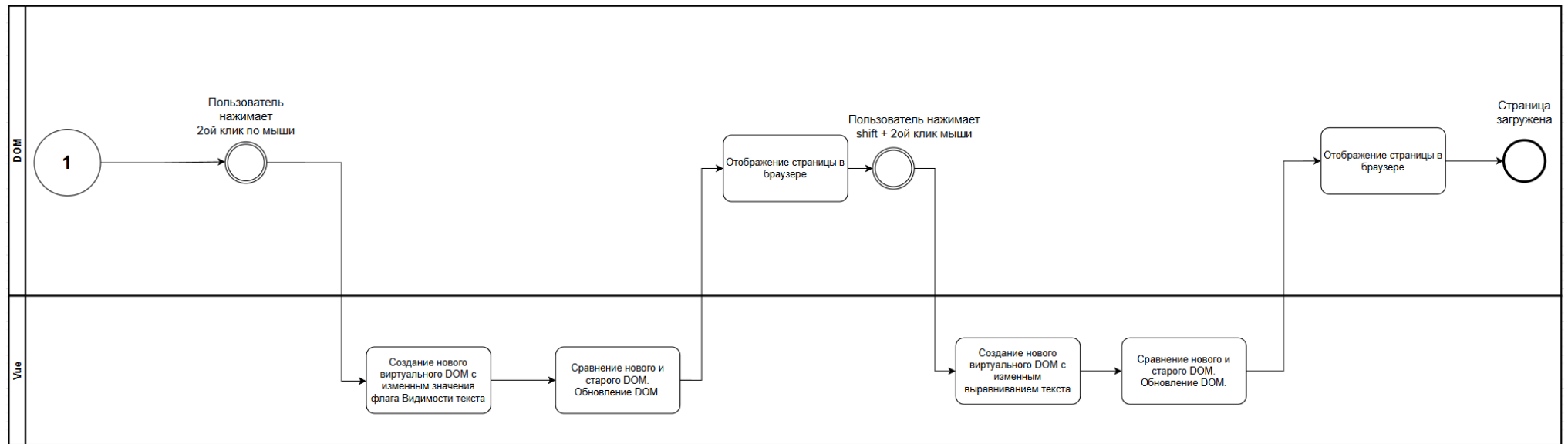


Рисунок 11 – Второй фрагмент BPMN схемы логики работы веб-приложения 2 задания

## Параметрическая связка

Листинг веб-приложения 3 задания:

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Задание 3</title>
  <script src="Vue.js"></script>
</head>
<body>
  <div id="app" @keydown="handleKeydown" tabindex="0">
    <div :style="textStyle">
      {{ displayText }}
    </div>
  </div>

  <script>
    new Vue({
      el: '#app',
      data: {
        displayText: 'Текст для выравнивания',
        horizontalPosition: 0, // 0: left, 1: center, 2: right
        verticalPosition: 0 // 0: top, 1: middle, 2: bottom
      },
      computed: {
        horizontalAlign: {
          get() {
            return ['left', 'center', 'right'][this.horizontalPosition];
          },
          set(value) {
            this.horizontalPosition = ['left', 'center', 'right'].indexOf(value);
          }
        },
        verticalPadding: {
          get() {
            return ['0px', '200px', '400px'][this.verticalPosition];
          },
          set(value) {
            this.verticalPosition = ['0px', '200px', '400px'].indexOf(value);
          }
        },
        textStyle() {
```

```

        return {
            textAlign: this.horizontalAlign,
            paddingTop: this.verticalPadding,
            width: '100%',
            height: '100px',
            boxSizing: 'border-box'
        };
    },
    methods: {
        handleKeydown(event) {
            // Управление выравниванием по горизонтали
            if (event.key === 'ArrowLeft') {
                if (this.horizontalPosition > 0) {
                    this.horizontalPosition--;
                }
            } else if (event.key === 'ArrowRight') {
                if (this.horizontalPosition < 2) {
                    this.horizontalPosition++;
                }
            } else if (event.key === 'ArrowUp') {
                if (this.verticalPosition > 0) {
                    this.verticalPosition--;
                }
            } else if (event.key === 'ArrowDown') {
                if (this.verticalPosition < 2) {
                    this.verticalPosition++;
                }
            }
        }
    }
});
</script>
<style>
    body {
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
        margin: 0;
    }

    #app {
        width: 500px;

```

```
height: 500px;
border: 1px solid #000;
display: flex;
justify-content: flex-start;
align-items: flex-start;
position: relative;
}
</style>
</body>
</html>
```

На рисунке 12 представлено отображение в браузере веб-приложения 3 задания в исходном состоянии.

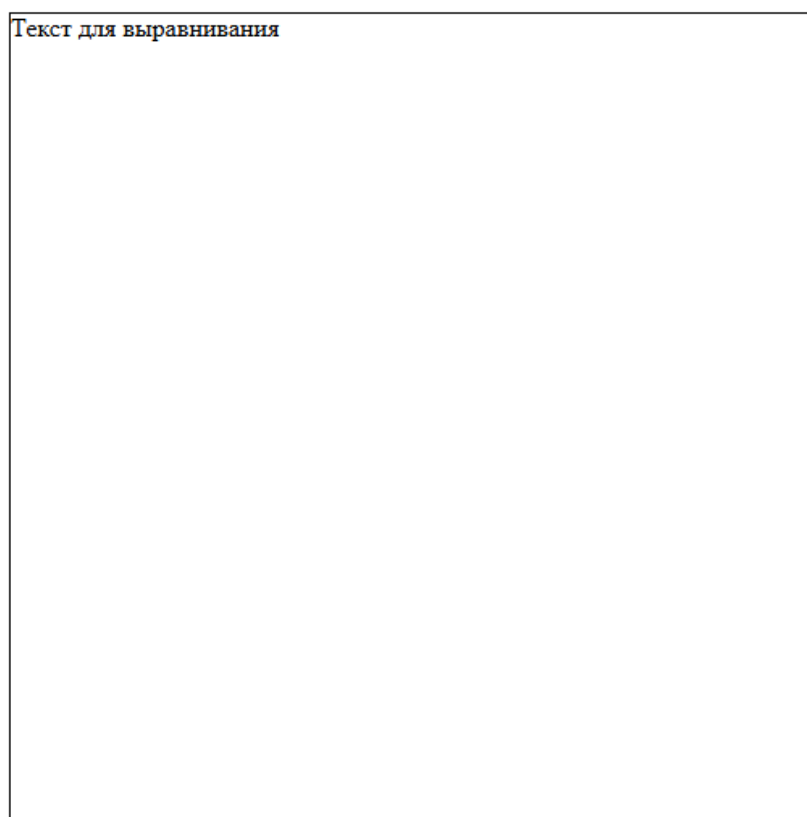


Рисунок 12 – Отображение веб-приложения 3 задания в исходном состоянии

На рисунке 13 представлено отображение в браузере веб-приложения 3 задания при нажатии на стрелочку вправо «→».

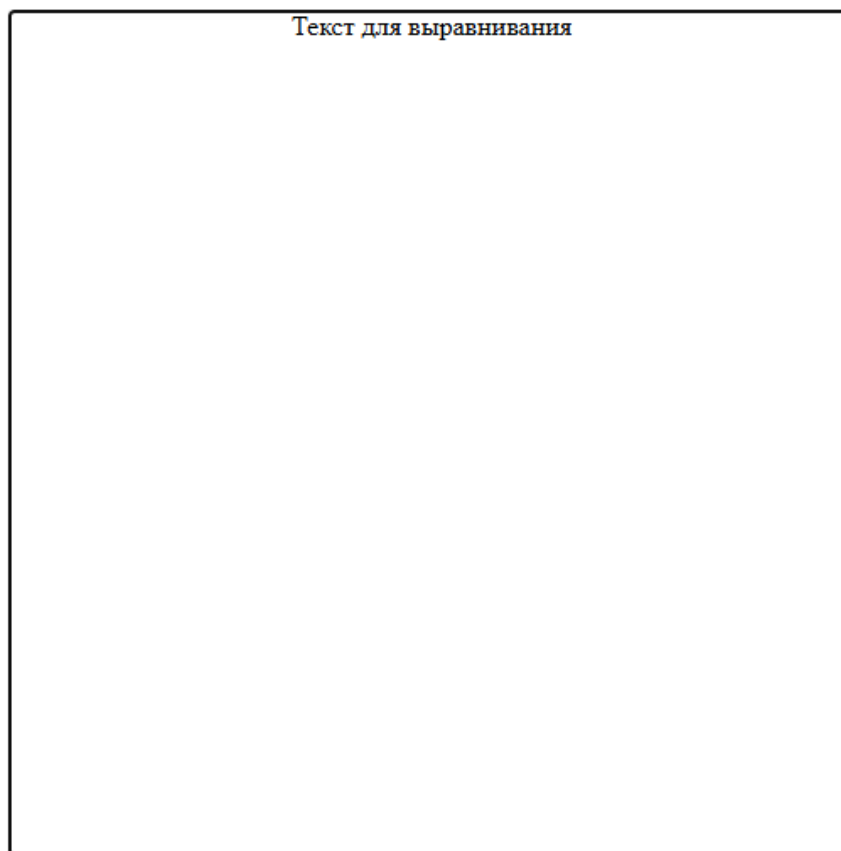


Рисунок 13 – Отображение веб-приложения 3 задания при нажатии на стрелочку вправо «→»

На рисунке 14 представлено отображение в браузере веб-приложения 3 задания при нажатии на стрелочку вправо «↓». Оставшиеся стрелки курсора выполняют соответствующие функции их направлению.

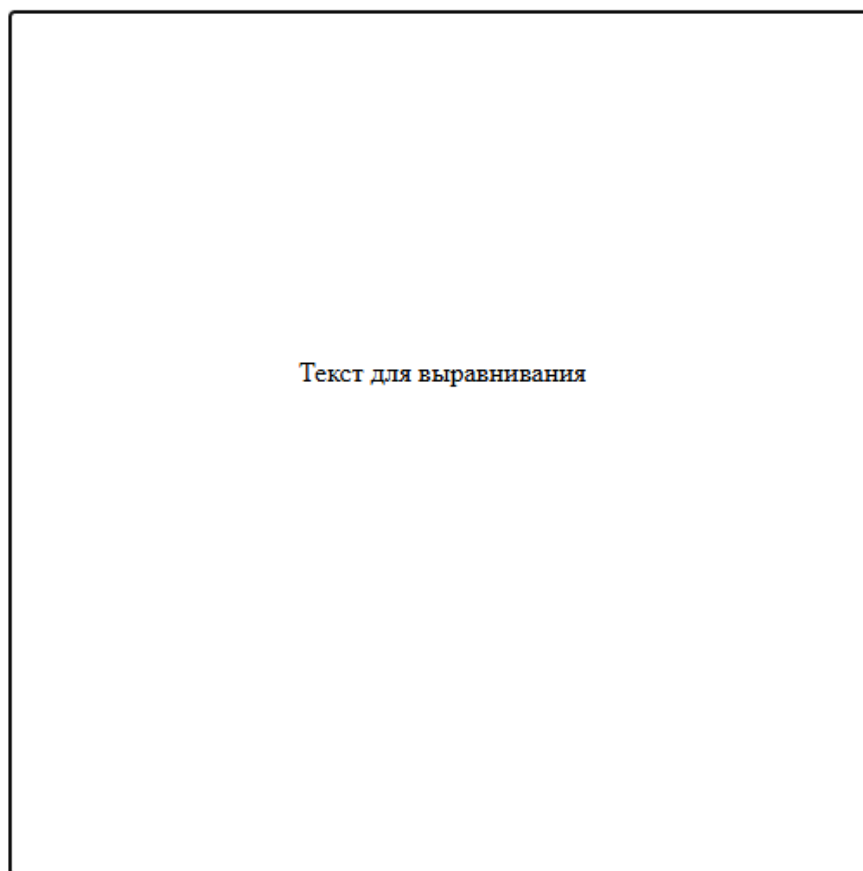


Рисунок 14 – Отображение веб-приложения 3 задания при нажатии на стрелочку вправо «→»

Первый фрагмент BPMN схемы логики работы веб-приложения 3 задания аналогичен с первым фрагментом BPMN схемой 2 задания (см. рисунок 10). Второй фрагмент BPMN схемы логики работы веб-приложения 3 задания представлен на рисунке 15.

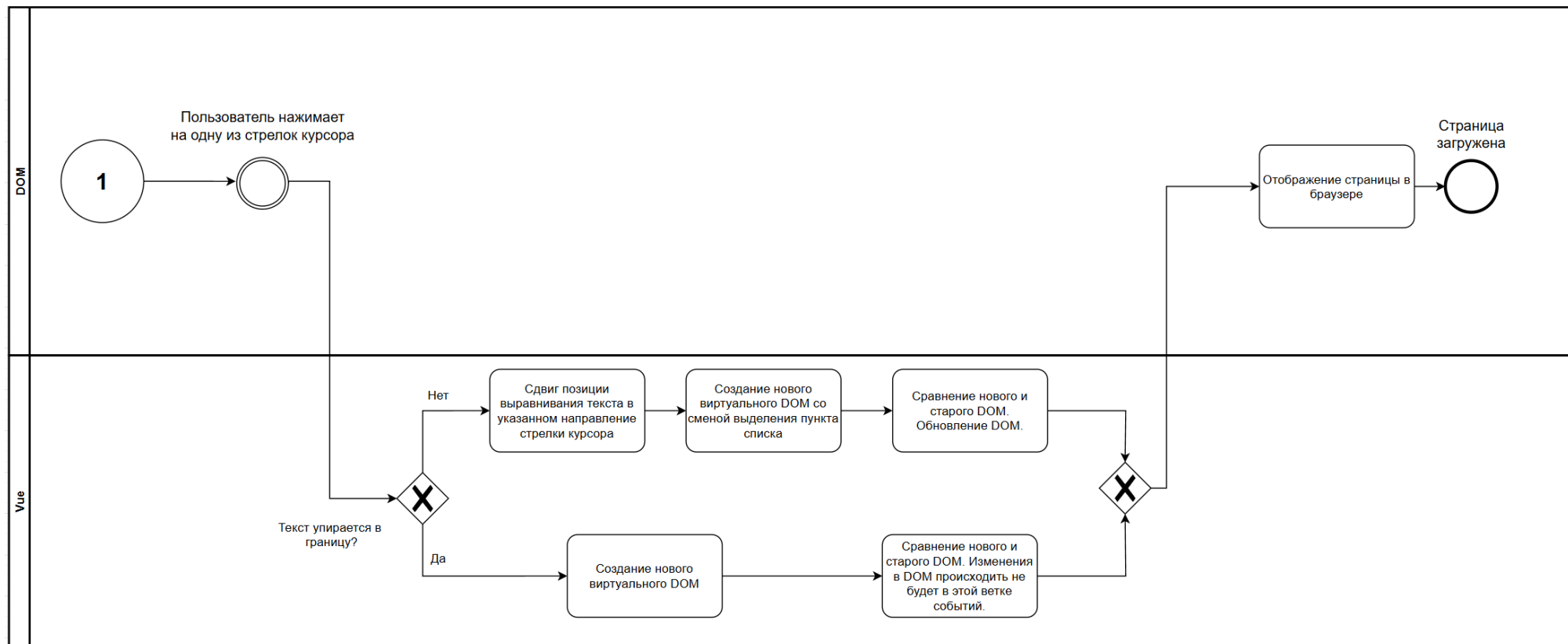


Рисунок 15 – Второй фрагмент BPMN схемы логики работы веб-приложения 3 задания

## Условная отрисовка/шаблоны

Первый фрагмент листинга веб-приложения 4 задания:

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Задание 4</title>
  <script src="Vue.js"></script>
</head>
<body>
  <div id="app">
    <!-- Форма авторизации -->
    <template v-if="currentMode === 'auth'" style="text-align: center; margin-top: 100px;">
      <h2>Форма авторизации</h2>
      <div>
        <label>Логин: </label>
        <input type="text" v-model="login">
      </div>
      <div style="margin-top: 10px;">
        <label>Пароль: </label>
        <input type="password" v-model="password">
      </div>
      <button style="margin-top: 20px;" @click="checkAuth">Войти</button>
      <p v-if="error" style="color: red;">{{ error }}</p>
    </template>

    <!-- Режим test -->
    <template v-if="currentMode === 'test'" style="text-align: center; margin-top: 100px;">
      <h2>Информация о разработчике</h2>
      <p>Фамилия: {{ lastName }}</p>
      <p>Имя: {{ firstName }}</p>
      <p>Отчество: {{ middleName }}</p>
      <p>Группа: {{ group }}</p>
      <button style="margin-top: 20px;" @click="logout">Выйти</button>
    </template>

    <!-- Режим admin -->
    <template v-if="currentMode === 'admin'" style="text-align: center; margin-top: 100px;">
      <h2>Редактирование информации</h2>
      <div>
        <label>Фамилия: </label>
        <input type="text" v-model="lastName">
      </div>
    </template>
  </div>
</body>
</html>
```



```

<div style="margin-top: 10px;">
  <label>Имя: </label>
  <input type="text" v-model="firstName">
</div>
<div style="margin-top: 10px;">
  <label>Отчество: </label>
  <input type="text" v-model="middleName">
</div>
<button style="margin-top: 20px;" @click="logout">Выйти</button>
</template>

<!-- Режим user -->
<template v-if="currentMode === 'user'" style="text-align: center; margin-top: 100px;">
  <h2>Ввод информации о группе</h2>
  <div>
    <label>Название и номер группы: </label>
    <input type="text" v-model="group">
  </div>
  <button style="margin-top: 20px;" @click="logout">Выйти</button>
</template>
</div>

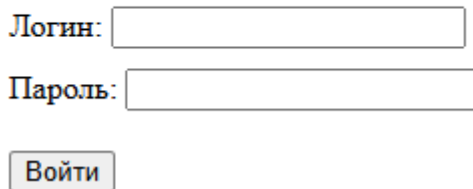
<script>
new Vue({
  el: '#app',
  data: {
    currentMode: 'auth',
    login: "",
    password: "",
    error: "",
    lastName: 'Дудкин',
    firstName: 'Александр',
    middleName: 'Владимирович',
    group: 'ТУУ-171'
  },
  methods: {
    checkAuth() {
      if (this.login === 'test' && this.password === 'test') {
        this.currentMode = 'test';
        this.error = "";
      } else if (this.login === 'admin' && this.password === 'admin') {
        this.currentMode = 'admin';
        this.error = "";
      } else if (this.login === 'user' && this.password === 'user') {

```

```
        this.currentMode = 'user';
        this.error = "";
    } else {
        this.error = 'Неверный логин или пароль';
    }
    this.login = "";
    this.password = "";
},
logout() {
    this.currentMode = 'auth';
}
}
});
</script>
</body>
</html>
```

На рисунке 16 представлено отображение в браузере веб-приложения 4 задания в исходном состоянии.

### **Форма авторизации**



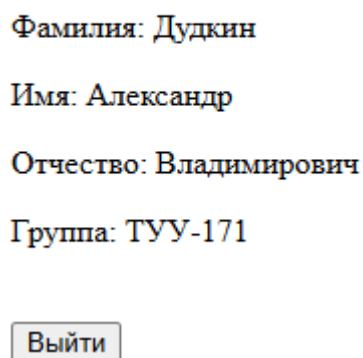
Логин:

Пароль:

Рисунок 16 – Отображение веб-приложения 4 задания в исходном состоянии

На рисунке 17 представлено отображение в браузере веб-приложения при вводе данных test/test для логина и пароля соответственно. Получаем полную картину данных по ФИО и номеру группы разработчика.

### **Информация о разработчике**



Фамилия: Дудкин

Имя: Александр

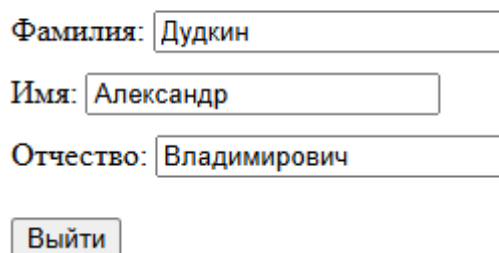
Отчество: Владимирович

Группа: ТУУ-171

Рисунок 17 – Отображение веб-приложения 4 задания с информацией о разработчике

На рисунке 18 представлено отображение в браузере веб-приложения при вводе данных admin/admin для логина и пароля соответственно (в исходном состоянии рис.16). Получаем окно с возможностью редактирования ФИО разработчика.

## Редактирование информации



Фамилия:

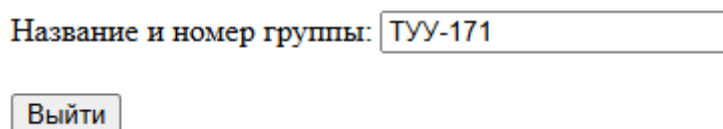
Имя:

Отчество:

Рисунок 18 – Отображение веб-приложения 4 задания с окном ввода ФИО разработчика

На рисунке 19 представлено отображение в браузере веб-приложения при вводе данных user/user для логина и пароля соответственно (в исходном состоянии рис.16). Получаем окно с возможностью редактирования номера группы разработчика.

## Ввод информации о группе



Название и номер группы:

Рисунок 19 – Отображение веб-приложения 4 задания с окном редактирования номера группы разработчика

Первый фрагмент BPMN схемы логики работы веб-приложения 3 задания аналогичен с первым фрагментом BPMN схемой 2 задания (см. рисунок 10). Второй, третий фрагменты BPMN схемы логики работы веб-приложения 4 задания представлены на рисунках 20 и 21 соответственно.

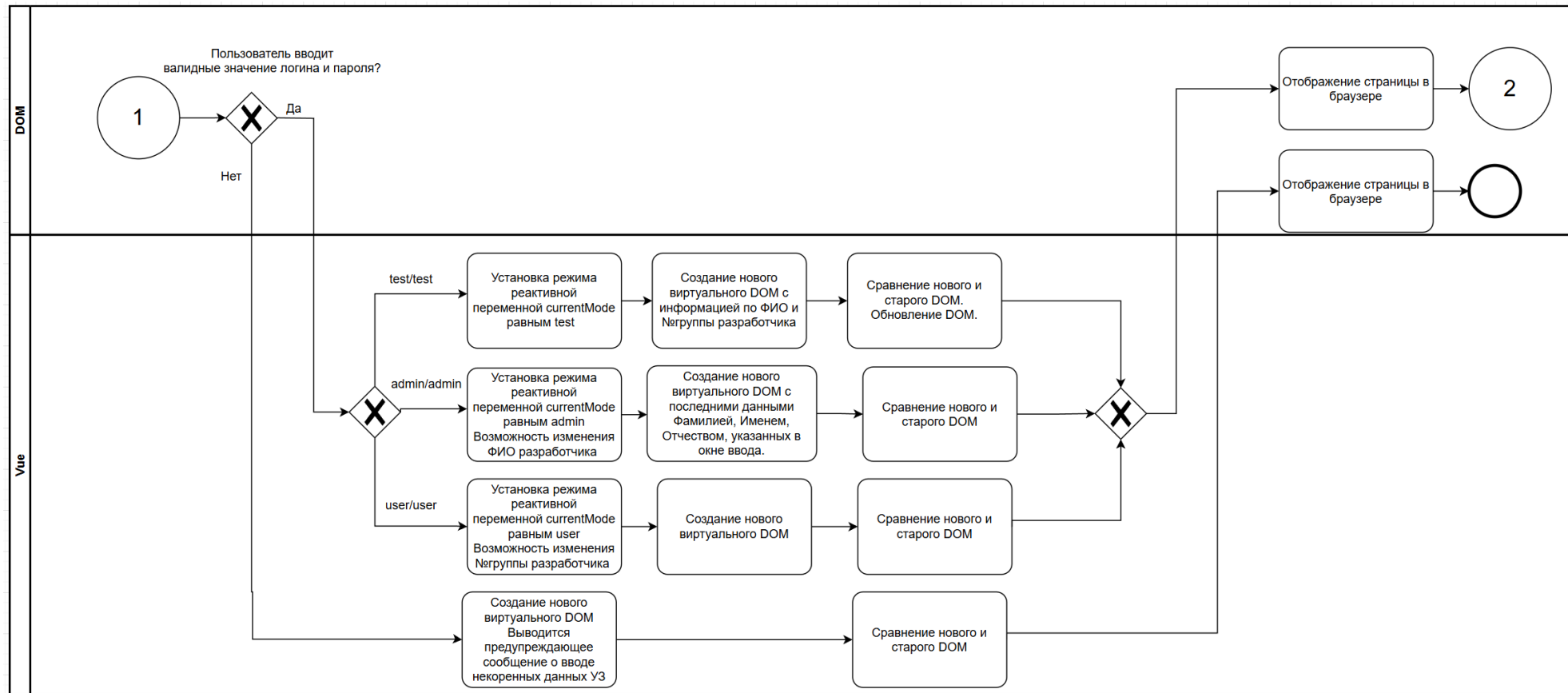


Рисунок 20 – Второй фрагмент BPMN схемы логики работы веб-приложения 4 задания

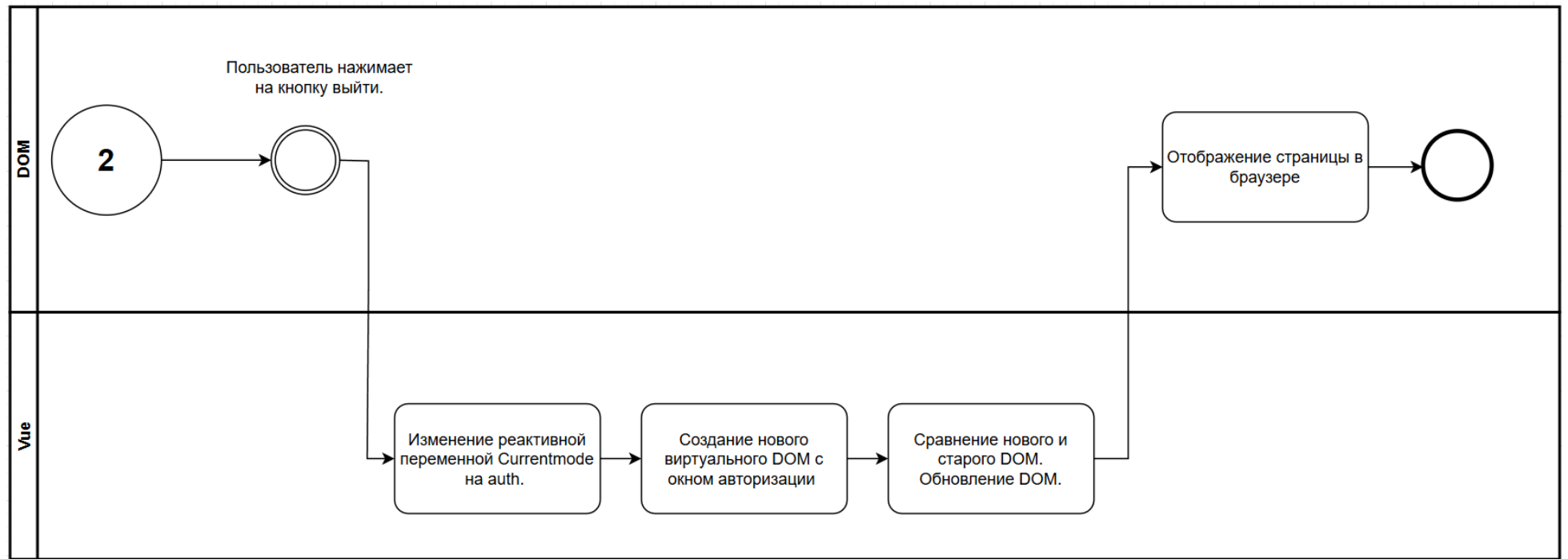


Рисунок 21 – Третий фрагмент BPMN схемы логики работы веб-приложения 4 задания кнопки Выйти

## **Вывод**

Получены навыки работы с фреймворком Vue.js, с директивами v-bind, v-on, v-model и v-if. Получены дополнительные знания о взаимодействии DOM и фреймворка Vue.js.