Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
“Петрозаводский государственный университет”

Институт математики и информационных технологий
Кафедра информатики и математического обеспечения

Фролова Марина Игоревна

Отчет по дисциплине «Верификация программного обеспечения»

Система автоматизированного распределения заявлений, поступающих в суд “CourtCase”

Направление 09.04.02 — Информационные системы и технологии Программа «Управление данными»

Преподаватель: к.ф.–м.н., доцент К.А. Кулаков

Петрозаводск
2022

Содержание

Объект тестирования

Объектом тестирования является система автоматизированного распределения заявлений, поступающих в суд, CourtCase. Краткий принцип работы заключается в следующем: система принимает на вход заявление от пользователя, который для этого заполняет поля соответствующей формы и отправляет дело на рассмотрение. Далее система обрабатывает поданное заявление, и данному делу назначается судья. В процессе подбора судьи для рассмотрения дела рассматриваются далее перечисленные факторы. Во-первых, коэффициент плановой нагрузки судьи, который представляет собой произведение коэффициента сложности дел на их количество. Во-вторых, учитывается доступность судьи на время рассмотрения дела, поскольку выбранный сотрудник может находиться в отпуске или на больничном и, соответственно, он не должен получить данное дело на рассмотрение в этот период. В-третьих, учитывается принадлежность судьи к выбранному заявителем суду.

Система разработана при помощи фреймворка React Native на языке TypeScript.

Функциональность объекта тестирования

Общий функционал приложения CourtCase:

1. Авторизация пользователя
2. Выход из системы
3. Создание заявления (путем заполнения формы)
4. Отправка заполненной формы заявления
5. Назначение судьи для рассмотрения поданного заявления
6. Просмотр отправленных заявлений и актуальной информации о статусе заявления и назначенном судье, по клику на соответствующую кнопку перенаправления

Интерфейс приложения

На рисунках ниже представлены макеты, отражающие интерфейс системы.

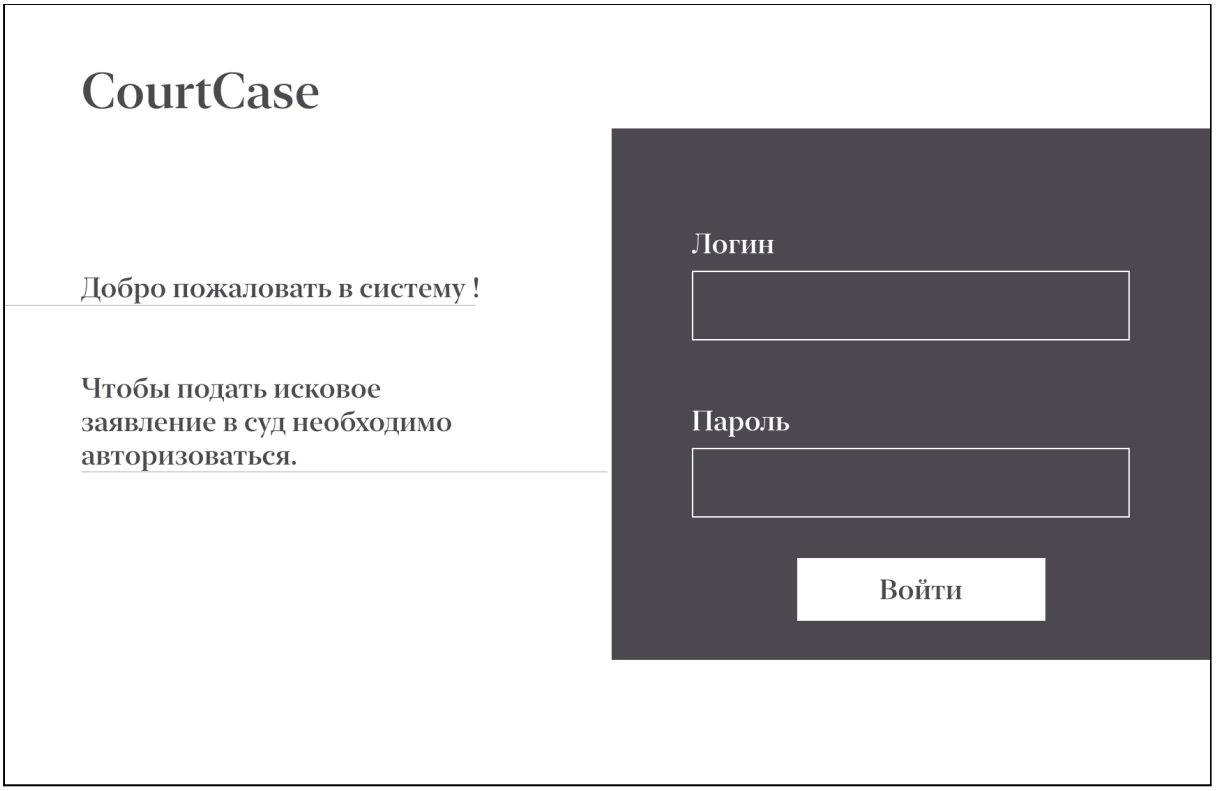

Рис. 1 Форма входа в систему

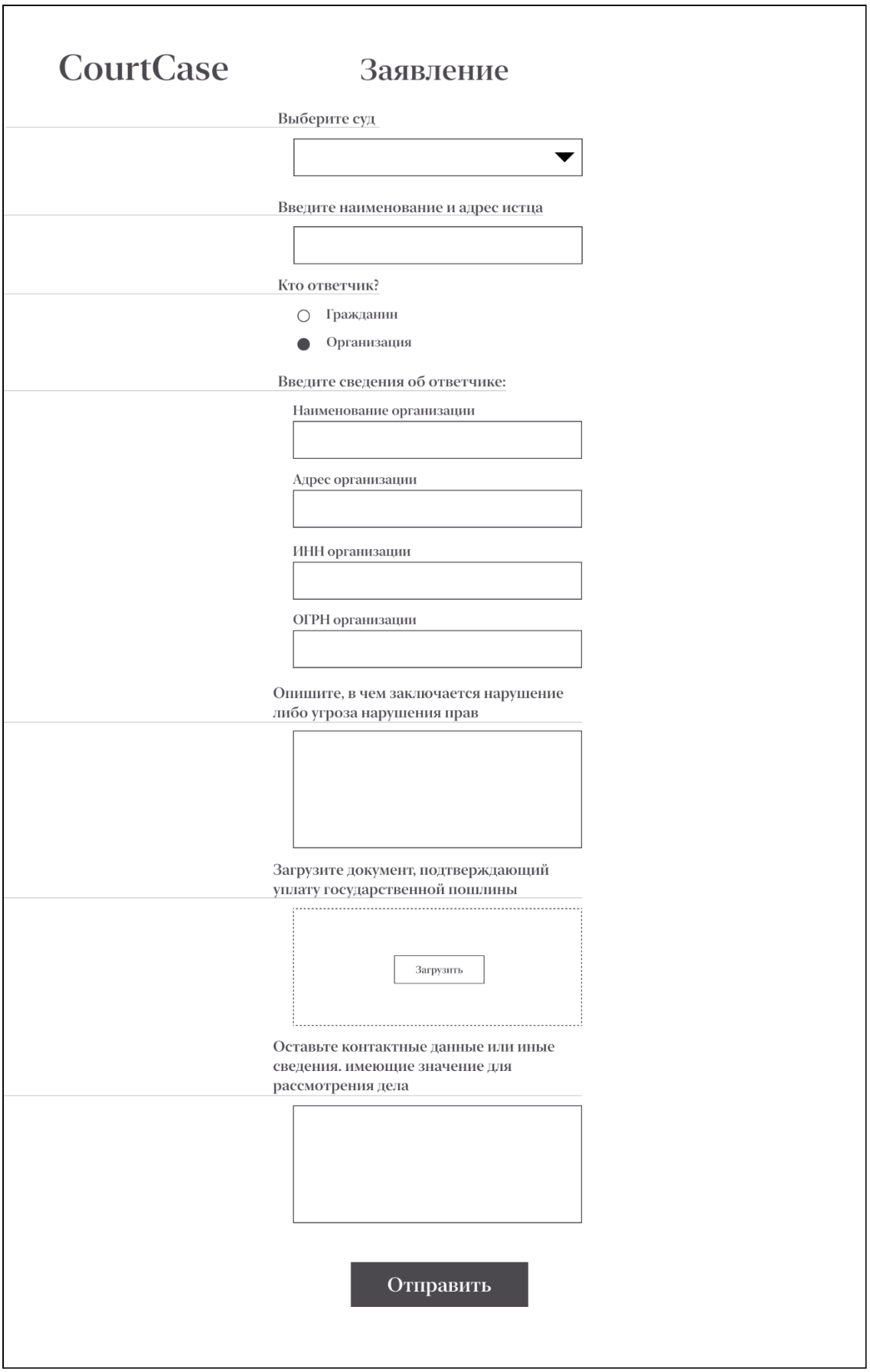

Рис. 2 Форма заявления для авторизованного пользователя

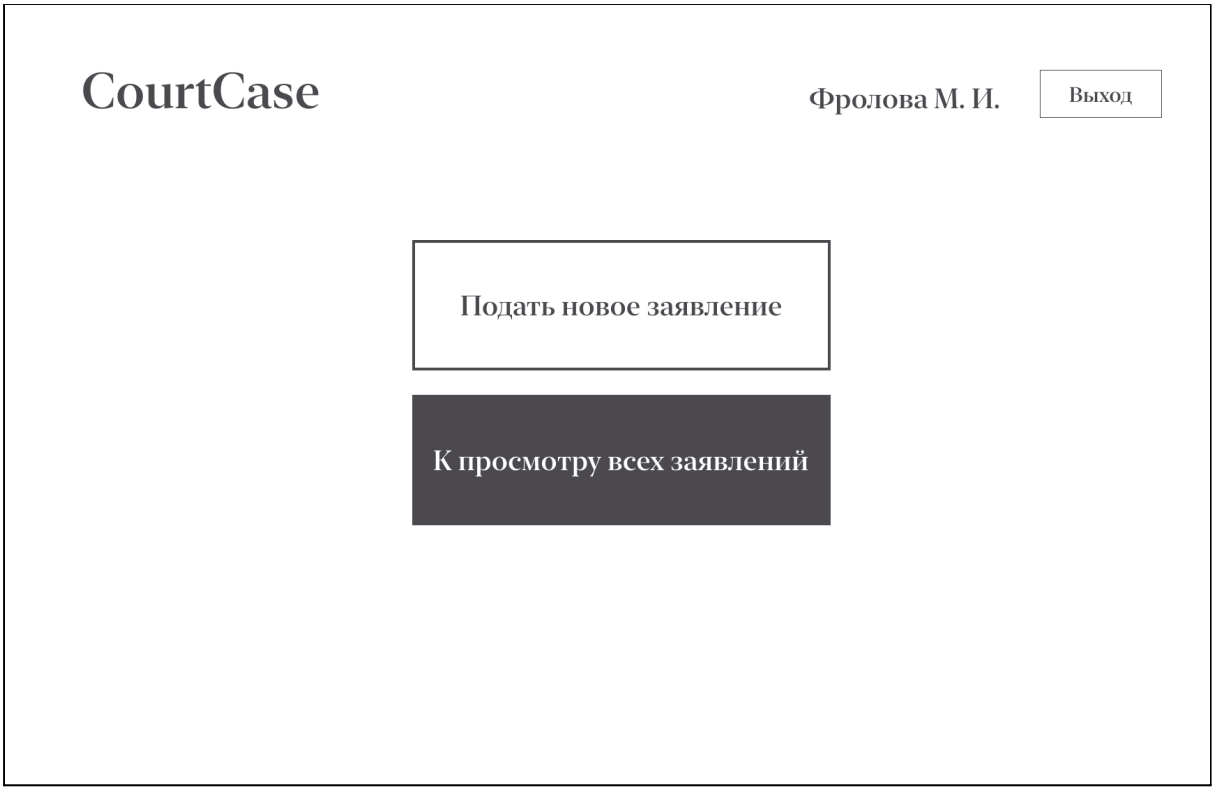

Рис. 3 Страница с кнопками для перехода к списку всех заявлений, которые отправил данный пользователь, и к созданию нового заявления. Также присутствует кнопка выхода из системы

Архитектура объекта тестирования

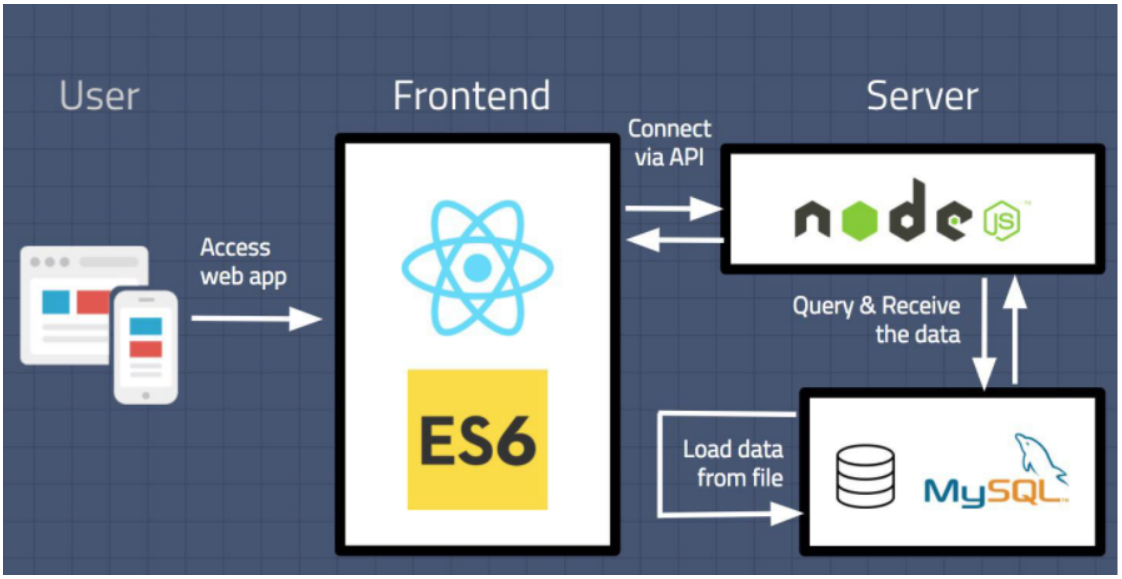

Рис. 5 Архитектура стандартного React Native приложения

Элементы системы в вышеописанной схеме:

Пользователь – человек, взаимодействующий с сервером посредством браузера и HTTP-запросов.

Web-сервер, хранящий в себе исходные коды и обеспечивающий их выполнение.

База данных – объект, используемый для хранения информации, обособлен от Web-сервера.

Пользователь заходит на веб-сайт и видит интерфейс, представленный на рисунках 1-3. Когда пользователь запрашивает данные, интерфейс соединяется с конечными точками API, созданными NodeJS запросить данные. NodeJS будет запрашивать данные из MySQLбазы данных, а затем отправить результат обратно в веб-интерфейс. MySQL база данных будет загружать данные из текстового файла в базу данных в течение каждого фиксированного периода времени.

Рассматриваемая система CourtCase состоит из связанных между собой модулей (компонентов):

- FormInput
- SingleSelect
- TextInput
- Dropdown
- Button
- AttachmentsInput
- JudgePicker

Блочное тестирование

В качестве инструмента тестирования был выбран фреймворк Jest, с помощью которого очень удобно провести не только блочное тестирование, но и интеграционное.

В данном случае, тестируется, например, модуль Dropdown - это компонент, который входит в состав формы заявления и который содержит в себе список судов. На рисунке 6 представлена реализация этого теста. В константе mockedOptions содержится список тестовых значений (option 1, option 2, option 3). Таким же образом можно протестировать компоненты: SingleSelect - компонент, отвечающий за реализацию единственного выбора из условий, TextInput - компонент для ввода текста, используется в формах (форма входа и форма заявления), Attachment Input - компонент для загрузки файлов.

```
import React from 'react';
import renderer from 'react-test-renderer';
import {DropdownComponent} from '../DropdownComponent';

const mockedOptions = ['option 1', 'option 2', 'option 3'];
test( name: 'Dropdown Component renders correctly', fn: () => {
    const tree = renderer
        .create(
            <DropdownComponent
                items={mockedOptions}
                labelExtractor={jest.fn()}
                defaultValue={''}
            />,
        )
        .toJSON();
    expect(tree).toMatchSnapshot();
});
```

Рис. 6 Пример тестирования компонента Dropdown

```
test( name: 'TextInputComponent on change function is called on text change', fn: () => {
    const mockFnChange = jest.fn();

    const {getByA11yLabel} = render(
        <TextInputComponent type={InputFormat.Text} onChange={mockFnChange} defaultValue={''} />,
    );

    const answerInputs = getByA11yLabel( matcher: 'Input answer');

    fireEvent.changeText(answerInputs, data: 'test 1');
    expect(mockFnChange).toBeCalledWith( args: 'test 1');
});
```

Рис. 7 Пример тестирования компонента TextInput

```
test( name: 'TextInputComponent renders correctly', fn: () => {
    const tree = renderer
        .create(
            <TextInputComponent
                type={InputFormat.Text}
                onChange={jest.fn()}
                defaultValue={'test default value'}
            />,
        )
        .toJSON();
    expect(tree).toMatchSnapshot();
});
```

Рис. 7.1 Пример тестирования компонента TextInput (рендеринг компонента)

№	Б1
Цель теста	Проверка корректной работы компонента
Тип теста	Позитивный
Объект теста	Компонент Button
Входные параметры	-

Ожидаемый результат	Корректное выполнение события onClick при нажатии на кнопку в зависимости от типа (если кнопка “Войти”, значит, по нажатию - переход на следующий экран с формой, если “Отправить” - отправка заполненной формы заявления и переход на следующий экран, если “Подать новое заявление” - переход на экран с формой для заполнения заявления, если “К просмотру всех заявлений”, то переход на экран, где отображен список заявлений от пользователя, если “Загрузить” - появление модального окна для загрузки файлов)

№	Б2
Цель теста	Проверка правильности рендринга
Тип теста	Позитивный
Объект теста	Компонент Button
Входные параметры	-
Ожидаемый результат	Вывод сообщения об успешном сравнении снимков (актуального и ожидаемого)

№	Б3
Цель теста	Проверка корректной работы компонента
Тип теста	Позитивный

Объект теста	Компонент SingleSelect
Входные параметры	-
Ожидаемый результат	При выборе одного условия включается соответствующий radiobutton, при последующем выборе противоположного условия, прежний radiobutton должен отключиться, новый - включиться

№	Б4
Цель теста	Проверка правильности рендринга
Тип теста	Позитивный
Объект теста	Компонент SingleSelect
Входные параметры	-
Ожидаемый результат	Вывод сообщения об успешном сравнении снимков (актуального и ожидаемого)

№	Б5
Цель теста	Проверка корректной работы компонента
Тип теста	Позитивный

Объект теста	Компонент TextInput
Входные параметры	test default value - значение по умолчанию
Ожидаемый результат	Беспрепятственный ввод и отображение введенного текста, возможность ввести многострочный текст (multiline = true). После ввода какого-то текста, значение по умолчанию отображаться больше не должно

№	Б6
Цель теста	Проверка правильности рендринга
Тип теста	Позитивный
Объект теста	Компонент TextInput
Входные параметры	-
Ожидаемый результат	Вывод сообщения об успешном сравнении снимков (актуального и ожидаемого)

№	Б7
Цель теста	Проверка корректной работы компонента

Тип теста	Позитивный
Объект теста	Компонент Dropdown
Входные параметры	[option1, option2, option3]
Ожидаемый результат	При нажатии на стрелочку (dropdown-arrow) появляется выпадающий список с наименованием доступных для выбора судов. При выборе одного из вариантов, он сохраняется в поле ввода

№	Б8
Цель теста	Проверка правильности рендринга
Тип теста	Позитивный
Объект теста	Компонент Dropdown
Входные параметры	[option1, option2, option3]
Ожидаемый результат	Вывод сообщения об успешном сравнении снимков (актуального и ожидаемого)

№	Б9
Цель теста	Проверка корректной работы компонента

Тип теста	Позитивный
Объект теста	Компонент AttachmentsInput
Входные параметры	test.png
Ожидаемый результат	Успешная загрузка выбранного файла

№	Б10
Цель теста	Проверка правильности рендринга
Тип теста	Позитивный
Объект теста	Компонент AttachmentsInput
Входные параметры	test.png
Ожидаемый результат	Вывод сообщения об успешном сравнении снимков (актуального и ожидаемого)

Интеграционное тестирование

Иногда модули по отдельности работают корректно, но их взаимодействие происходит не так, как ожидалось. Интеграционные тесты работают с целыми процессами, проверяя правильность взаимодействия отдельных модулей и побочные эффекты.

Учитывая, что смысл и главная цель системы CourtCase заключается в распределении заявления, отправленного пользователем, судье в соответствии с факторами (коэффициент плановой нагрузки, суд и доступность судьи в данный период времени), то, получается, функция подбора судьи JudgePicker напрямую

зависит от результата выбора пользователем суда (при заполнении формы первым делом пользователь через Dropdown выбирает суд). Также наблюдается зависимость компонента формы FormInput от наполняющих его модулей: SingleSelect, Dropdown и AttachmentsInpu.

№	И1
Цель теста	Проверка работоспособности взаимодействия FormInput с TextInput, Dropdown, AttachmentsInput и SingleSelect
Тип теста	Позитивный
Объект теста	Компоненты FormInput, TextInput, Dropdown, AttachmentsInput и SingleSelect, Button
Входные параметры	Корректное заполнение всех полей формы, загрузка тестовой картинки. Клик по кнопке Button
Ожидаемый результат	В результате отправки формы по нажатию на кнопку корректно сформировано заявление, данные не утеряны

№	И2
Цель теста	Проверка правильности рендринга
Тип теста	Позитивный
Объект теста	Компоненты FormInput, TextInput, Dropdown, AttachmentsInput и SingleSelect
Входные параметры	-

Ожидаемый результат	Вывод сообщения об успешном сравнении снимков (актуального и ожидаемого)

№	Из
Цель теста	Проверка работоспособности взаимодействия результата выбора в компоненте Dropdown и функции JudgePicker
Тип теста	Позитивный
Объект теста	Функция JudgePicker и компонент Dropdown
Входные параметры	В компоненте Dropdown выбран определённый суд
Ожидаемый результат	В результате работы функции JudgePicker для рассмотрения дела был определен судья, относящийся к суду, выбранному в компоненте Dropdown на экране с формой

Аттестационное тестирование

При аттестационном тестировании происходит тестирование системы полностью. В частности, в данном случае можно применить скриншотное тестирование. Скриншотным тестированием мы проверяем регрессии пользовательского интерфейса. Сперва мы делаем скриншот-эталон, с которым потом сравниваем интерфейс после изменений. Если скриншоты совпадают, значит UI остался тем же, и никаких ошибок при изменении кода мы не допустили. Если же скриншоты отличаются, мы что-то упустили из виду. Также немаловажно применить end-2-end тестирование, такие тесты имитируют, как пользователь будет работать с программой. Обычно такие тесты описывают сценарий для теста в терминах действий пользователя, в случае системы CourtCase необходимо проверить работу форм входа и заполнения заявлений, работу всех кнопок на предмет кликабельности и выполнения действия по клику и т. п.

№	A1
Описание	Проверяются функциональные требования: Вход в систему
Тип теста	Позитивный
Сценарий аттестационного тестирования	1. Запустить приложение 2. Открыть стартовую страницу 3. Заполнить поля формы входа в систему корректными данными 4. Нажать на кнопку “Войти”
Ожидаемый результат	Вход в программу - переход на экран с формой для заполнения заявления

№	A2
Описание	Проверяются функциональные требования: Вход в систему
Тип теста	Негативный
Сценарий аттестационного тестирования	1. Запустить приложение 2. Открыть стартовую страницу 3. Заполнить поля формы входа в систему корректными данными 4. Нажать на кнопку “Войти”
Ожидаемый результат	Вход в программу - переход на экран с формой для заполнения заявления

№	A3
Описание	Проверяются функциональные требования: Создание заявления
Тип теста	Позитивный
Сценарий аттестационного тестирования	1. Запустить приложение 2. Открыть стартовую страницу 3. Заполнить поля формы входа в систему корректными данными 4. Нажать на кнопку “Войти” 5. Перейти на страницу с формой для заполнения заявления 6. Заполнить все поля формы 7. Подгрузить изображение или pdf файл 8. Нажать на кнопку “Отправить”

Ожидаемый результат	Успешное создание и отправка заявления в суд, переход на следующую страницу

№	А4
Описание	Проверяются функциональные требования: Просмотр отправленных заявления и актуальной информации о статусе заявления и назначенном судье
Тип теста	Позитивный
Сценарий аттестационного тестирования	1. Запустить приложение 2. Открыть стартовую страницу 3. Заполнить поля формы входа в систему корректными данными 4. Нажать на кнопку “Войти” 5. Перейти на страницу с формой для заполнения заявления 6. Заполнить все поля формы 7. Подгрузить изображение или pdf файл 8. Нажать на кнопку “Отправить” 9. Перейти на страницу с кнопками “Подать новое заявление” и “К просмотру всех заявлений” 10. Нажать на кнопку “К просмотру всех заявлений” 11. Перейти на страницу со списком всех заявлений, созданных пользователей (у каждого заявления сгенерирован и назначен судья)
Ожидаемый результат	Просмотр списка отправленных пользователем заявлений и актуальные сведения об этих заявлениях

№	А5
Описание	Проверяются функциональные требования: Просмотр отправленных заявления и актуальной информации о статусе заявления и назначенном судье
Тип теста	Позитивный
Сценарий аттестационного тестирования	1. Запустить приложение 2. Открыть стартовую страницу

	3. Заполнить поля формы входа в систему корректными данными 4. Нажать на кнопку “Войти” 5. Перейти на страницу с формой для заполнения заявления 6. Заполнить все поля формы 7. Подгрузить изображение или pdf файл 8. Нажать на кнопку “Отправить” 9. Перейти на страницу с кнопками “Подать новое заявление” и “К просмотру всех заявлений” 10. Нажать на кнопку “Подать новое заявление” 11. Перейти на страницу с формой для заполнения заявления 12. Заполнить все поля формы 13. Подгрузить изображение или pdf файл 14. Нажать на кнопку “Отправить” 15. Перейти на страницу с кнопками “Подать новое заявление” и “К просмотру всех заявлений”
Ожидаемый результат	После успешного создания заявления есть возможность создать еще одно заявление

№	A5
Описание	Проверяются функциональные требования: Выход из системы
Тип теста	Позитивный
Сценарий аттестационного тестирования	1. Запустить приложение 2. Открыть стартовую страницу 3. Заполнить поля формы входа в систему корректными данными 4. Нажать на кнопку “Войти” 5. Перейти на страницу с формой для заполнения заявления 6. Заполнить все поля формы 7. Подгрузить изображение или pdf файл 8. Нажать на кнопку “Отправить” 9. Перейти на следующий экран (с кнопкой для перехода к общему списку заявлений) 10. Нажать на кнопку “Выйти” 11. Перейти на стартовую страницу

Ожидаемый результат	Успешный выход из системы, возвращение на стартовую страницу с формой для входа в систему

Нагрузочное тестирование

Для проведения нагрузочного тестирования инструмент Jest уже не подойдет, в данном случае нужно использовать такой инструмент как K6, разработанный для этих целей. С помощью K6 можно реализовать проверку, например, как это показано на рисунке 8 (код взят из официальной документации к K6)

```
1  import { group, check } from 'k6';
2  import http from 'k6/http';
3
4  const id = 5;
5
6  // reconsider this type of code
7  group('get post', function () {
8    http.get(`http://example.com/posts/${id}`);
9  });
10 group('list posts', function () {
11   const res = http.get(`http://example.com/posts`);
12   check(res, {
13     'is status 200': (r) => r.status === 200,
14   });
15 });
```

Рис. 8 Пример проверки (check метод) с помощью инструмента K6