

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ”

Факультет программной инженерии и компьютерной техники
(название факультета)
Кафедра информатики и прикладной математики
(название кафедры)
Направление подготовки (специальность) 09.04.01

О Т Ч Ё Т

о производственной (НИР) **практике**
(название практики)

Тема задания «Разработка веб-интерфейса для проведения геодезических изысканий с помощью устройств Emlid Reach и Emlid Reach RS»

Студент Кузнецов Андрей Андреевич **Группа №** Р4215
(Фамилия, Имя, Отчество)

Руководитель практики от организации Фёдоров Е.М., ИП Николаев Д.А., руководитель от-
дела разработки программного обеспечения
(Фамилия И.О., должность и место работы)

Руководитель практики от университета Исаев И.В., ассистент
(Фамилия И.О., должность)

Практика пройдена с оценкой _____

Подписи членов комиссии

_____ (подпись)	_____ (Фамилия И.О.)
_____ (подпись)	_____ (Фамилия И.О.)
_____ (подпись)	_____ (Фамилия И.О.)

Дата « _____ » _____ 20 ____ г.

Санкт-Петербург
2018 г.

СОДЕРЖАНИЕ

1 ОБЩИЕ СВЕДЕНИЯ	3
2 ХОД РАБОТЫ	3
2.1 Этап 1 – Знакомство с платформой разработки	3
2.2 Этап 2 – Постановка задачи	4
2.2.1 Требования к модулю	4
2.3 Этап 3 – Разработка модуля	5
2.4 Этап 4 – Тестирование модуля	7
2.4.1 Модульное тестирование	7
2.4.2 Функциональное тестирование	9
2.4.3 Полевые испытания устройств	10
2.5 Этап 5 – Оформление пользовательской документации	10

1 ОБЩИЕ СВЕДЕНИЯ

С 5 февраля по 29 апреля 2018 года обучающийся проходил производственную практику в ИП Николаев Денис Александрович. На практику было дано задание по разработке программного модуля веб-приложения для управления ГНСС-приёмником, работающим под управлением программного обеспечения, основанного на программном комплексе высокоточного позиционирования RTKLIB.

Разрабатываемый модуль предназначен для осуществления геодезических изысканий с помощью веб-интерфейса устройств Emlid Reach и Emlid Reach RS.

В процессе прохождения практики были изучены следующие электронные источники и литература:

- документация программного комплекса RTKLIB;
- документация устройств Emlid Reach и Emlid Reach RS;
- техническое задание на разработку программного модуля.

2 ХОД РАБОТЫ

2.1 Этап 1 – Знакомство с платформой разработки

В рамках данной практики платформой для разработки являлись устройства компании Emlid: ГНСС-модуль Reach и ГНСС-приёмник Reach RS. Данные устройства работают под управлением программного обеспечения, основанного на программном комплексе высокоточного позиционирования RTKLIB. Работа пользователя с данными продуктами осуществляется через веб-приложение, доступ к которому можно получить с помощью любого устройства, на котором установлен современный веб-браузер.

Веб-клиент рассматриваемых устройств написан с использованием языков программирования Python и JavaScript.

Несмотря на то, что Reach и Reach RS созданы на базе одного и того же вычислительного модуля и используют одинаковые приёмники u-blox, имеется ряд существенных различий в аппаратном обеспечении данных устройств. Различия Reach и Reach RS, которые необходимо учесть при создании веб-приложения, указаны в таблице 2.1.

Таблица 2.1 – Различия Reach и Reach RS

Техническая/функциональная особенность	Reach	Reach RS
Встроенная батарея	нет	да
Встроенная антенна	нет	да
Встроенное радио	нет	да
Физическая кнопка на корпусе	нет	да
Возможность управления фотокамерой	да	нет

2.2 Этап 2 – Постановка задачи

Основной задачей производственной практики являлось создание программного компонента, необходимого для проведения геодезических изысканий с помощью вышеупомянутых ГНСС-приёмников.

Также ставится задача встраивания рассматриваемого программного модуля в существующее веб-приложение, через которое осуществляется вся работа с приёмником.

2.2.1 Требования к модулю

Модуль должен реализовывать инструменты для проведения геодезических изысканий и предоставлять пользователю следующие возможности:

- создание и удаление проектов;
- экспорт проектов в различных форматах;
- сбор точек;
- просмотр данных проекта на интерактивной карте.

2.2.1.1 Создание проекта

При создании нового проекта пользователь должен:

- указать название проекта;
- указать имя автора проекта;
- добавить описание проекта (опционально);
- ввести высоту антенны по умолчанию;
- создать до трёх правил для автоматического сбора точек.

Правила для автоматического сбора точек представляют собой условия, при которых устройство автоматически будет принимать результаты усреднения координат. Параметры правил:

- статус решения (*Single* – автономная позиция, *Float* – «плавающее решение» или *Fix* – «фиксированное» решение);
- минимальное время сбора (от одной секунды до одного часа);
- максимальное допустимое значение погрешности измерений;
- максимальное допустимое значение DOP (*dilution of precision* с англ. – «снижение точности»).

2.2.1.2 Экспорт проекта

Экспорт проектов должен быть доступен в следующих форматах:

- GeoJSON;
- ESRI Shapefile;
- DXF;
- CSV;
- DroneDeploy CSV.

2.3 Этап 3 – Разработка модуля

С помощью разрабатываемого модуля приложения будет происходить основная часть работы с устройством. Используя интерфейс модуля, пользо-

ватель может производить геодезические изыскания – сбор точек на местности с разделением их на проекты.

В соответствии с заявленными требованиями был разработан модуль приложения, основанный на сущностях и прецедентах, описанных на рисунках 2.1 и 2.2 соответственно.

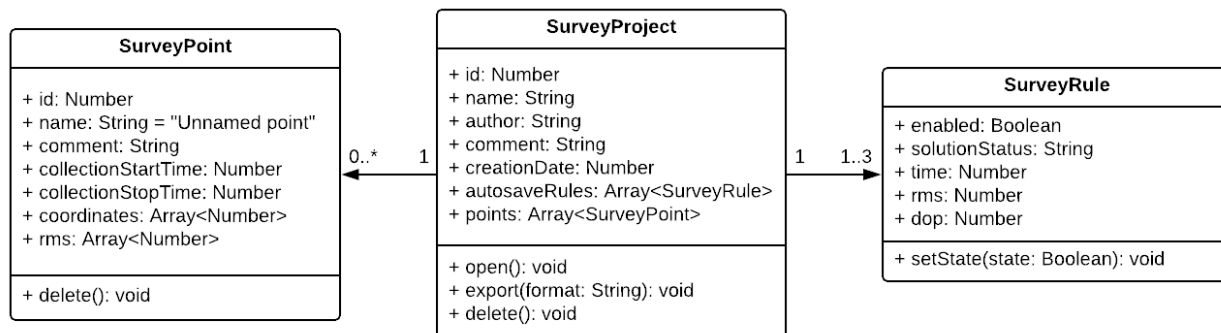


Рисунок 2.1 – Диаграмма классов модуля

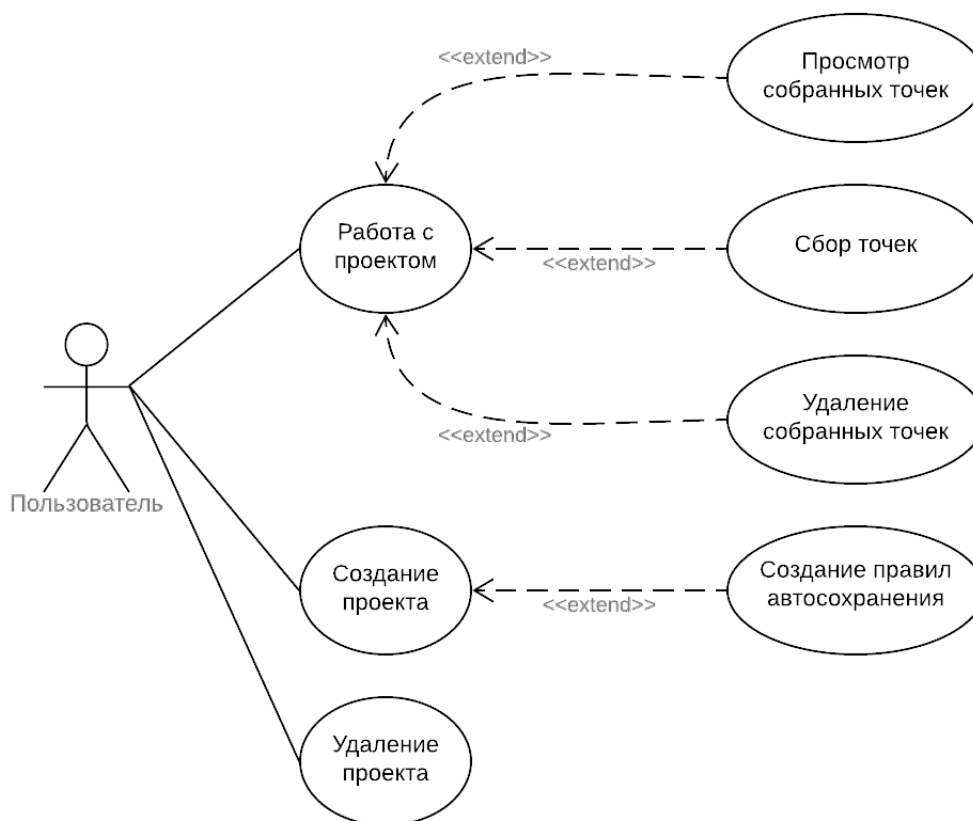


Рисунок 2.2 – Диаграмма прецедентов модуля

Модуль состоит из множества отдельных представлений, переключа-
ясь между которыми, пользователь осуществляет разнообразные действия

с собранными данными. Одной из основных задач при разработке данного модуля было чёткое описание возможных состояний и условий перехода между ними. На рисунке 2.3 представлена диаграмма состояний модуля.

На данный момент рассматриваемый модуль предоставляет пользователю инструменты только для сбора точек и их организации (см. рис. 2.2).

В модуле содержится следующий алгоритм сбора точек:

- а) Получение очередной тройки координат;
- б) Расчёт выборочного среднего (англ. *sample mean*) для координат собираемой точки;
- в) Вычисление несмещённой оценки дисперсии (англ. *unbiased sample variance*) каждой из трёх координат;
- г) Вычисление среднеквадратичной ошибки среднего арифметического (англ. *root-mean-square error, RMSE*) каждой из трёх координат;
- д) Вычисление горизонтальной среднеквадратичной ошибки среднего арифметического.

Для вычисления значения в) используется метод Велфорда.

2.4 Этап 4 – Тестирование модуля

2.4.1 Модульное тестирование

В ходе разработки исходный код приложения был покрыт модульными тестами. Для написания тестов были использованы следующие инструменты:

- **Karma** – утилита для запуска JavaScript-тестов;
- **Mocha** – тестовый фреймворк для JavaScript-приложений;
- **Istanbul** – утилита для анализа покрытия исходного кода JavaScript-приложений модульными тестами.

В листинге 2.1 представлен пример объявления тестовых случаев (англ. *test case*) и тестовых наборов (англ. *test suite*).

Листинг 2.1 – Тесты Mocha

```
1 describe('Toggle.vue', () => {
2   describe('Render behavior', () => {
3     beforeEach(() => { ... });
4
5     it('Should render correct title', (done) => { ... });
6
7     it('Should render correct color', (done) => { ... });
8   });
9
10  describe('Click behavior', () => {
11    beforeEach(() => { ... });
12
13    it('Should toggle active class on click', (done) => { ... });
14
15    it('Should not toggle active class on click when disabled', (done) => {
16      ...
17    });
18  });
19 });
```

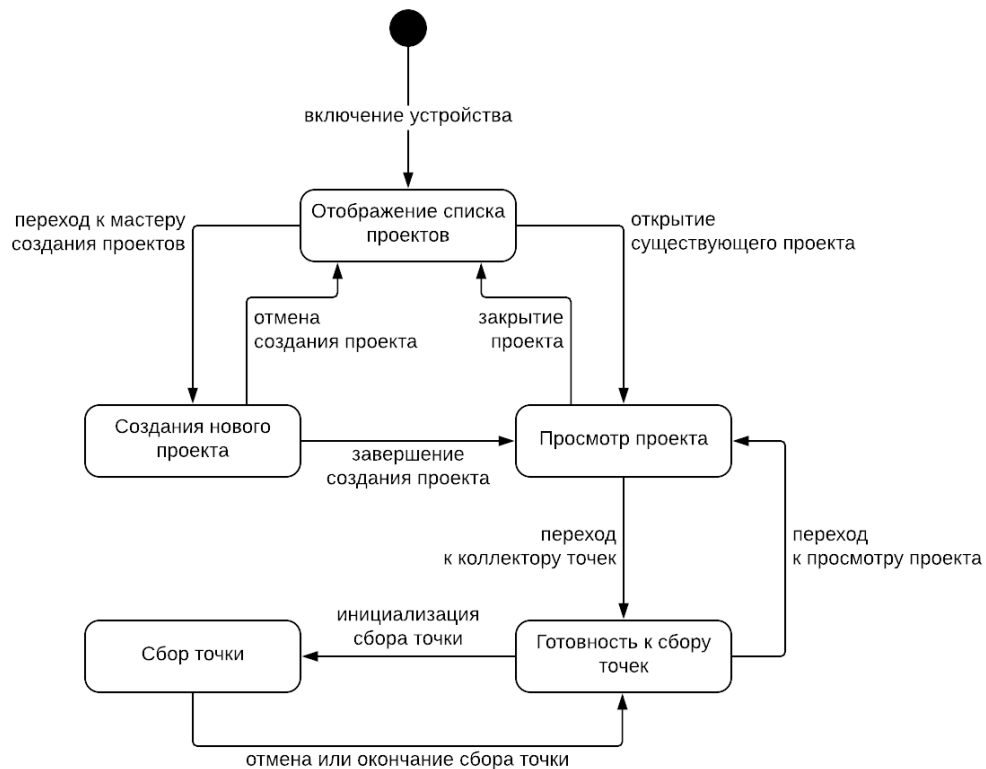


Рисунок 2.3 – Диаграмма состояний модуля

На рисунке 2.4 продемонстрирован пример вывода Karma, работающей с Istanbul.

```
$ npm run test

...

Executed 148 of 154 (skipped 6) SUCCESS (6.701 secs / 5.888 secs)
TOTAL: 148 SUCCESS

===== Coverage summary =====
Statements : 58.99% ( 1109/1880 )
Branches   : 51.66% ( 606/1173 )
Functions  : 47.92% ( 115/240 )
Lines      : 59.58% ( 1088/1826 )
=====
```

Рисунок 2.4 – Запуск тестов с помощью утилиты Karma

2.4.2 Функциональное тестирование

Функциональное тестирование приложения проводилось в ручном и автоматизированном режимах. Для автоматизации тестов были использованы язык программирования Python и библиотека Selenium, позволяющая автоматизировать управление веб-браузерами (см. листинг 2.2).

Благодаря использованию тестового фреймворка pytest, предназначенного для написания тестов на Python, был автоматизирован процесс тестирования приложения в нескольких браузерах. Вызов графического сервера Xvfb из кода Python-тестов позволил добавить возможность проверки приложения на различных разрешениях экрана.

Листинг 2.2 – Тестирование с помощью pytest и Selenium

```
1 from selenium.webdriver.support import expected_conditions as EC
2 from selenium.webdriver.support.wait import WebDriverWait
3
4 from tools.locators import Locators as locators
5 from tools.screenshot import screenshot
6
7
```

```
8 @screenshot
9 def test_check_battery_icon(browser, device):
10     if device == 'Reach':
11         WebDriverWait(browser, 0).until_not(
12             EC.visibility_of_any_elements_located(locators.battery_icon)
13         )
14     elif device == 'ReachRS':
15         WebDriverWait(browser, 0).until(
16             EC.visibility_of_any_elements_located(locators.battery_icon)
17         )
```

2.4.3 Полевые испытания устройств

Разработанный веб-интерфейс был протестирован при работе устройств под открытым небом. Был произведён ручной сбор точек на местности, а также осуществлена проверка работы приложения при использовании правил автоматического сбора точек.

2.5 Этап 5 – Оформление пользовательской документации

После проведения всех необходимых проверок разработанный модуль был добавлен в очередной стабильный выпуск приложения. К новому модулю была написана подробная пользовательская документация, доступная на страницах официального сайта компании Emlid.