

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Государственное образовательное учреждение высшего профессионального образования  
САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО  
Институт компьютерных наук и кибербезопасности  
Высшая школа программной инженерии

**ПРОЕКТ ИНТЕРНЕТ-СЕРВИСА НАБЛЮДЕНИЯ ЗА ПОГОДОЙ**  
**Функциональные спецификации**

по дисциплине «Технологии разработки качественного программного обеспечения»

Группа 5130904/20105

Команда:

Вакама Юджения Ибимина

Гайнетдинов Искандар Райханович

Гурьянов Кирилл Евгеньевич

Эйсмонт Михаил Михайлович

Преподаватель: Маслаков Алексей Павлович

Санкт-Петербург 2025 г.

# ОГЛАВЛЕНИЕ

1.1 Назначение.....	3
1.2 Функции и возможности.....	3
1.2.1 Сбор и обработка данных о погоде.....	3
1.2.1.1 Выбор источников данных.....	3
1.2.1.2 Получение данных.....	5
1.2.1.3 Суммирование данных.....	6
1.2.2 Веб-интерфейс.....	6
1.2.2.1 Отображение погоды.....	6
1.2.2.2 Выбор населенного пункта.....	7
1.2.2.3 Внешний вид, персонализация и совместимость.....	7
1.2.3 Telegram-бот.....	8
1.2.4 Рекомендации по одежде и аксессуарам.....	9
1.2.5 Учет пользователей.....	9
1.2.6 Администрирование.....	10
1.3 Нефункциональные требования.....	10
1.4 Технологии.....	10

## 1.1 Назначение

Разрабатываемый продукт предназначен для предоставления информации о погоде в выбранном населенном пункте. Сервис агрегирует данные из нескольких источников, суммирует их, визуализирует и предоставляет пользователю рекомендации по одежде и аксессуарам.

Доступ к сервису будет осуществляться через веб-интерфейс и Telegram-бота.

Задачи:

1. Сбор данных о погоде из нескольких источников.
2. Суммирование данных с учетом приоритетов источников.
3. Предоставление прогноза погоды на текущий день и на неделю вперед.
4. Формирование персонализированных рекомендаций по одежде и аксессуарам.
5. Обеспечение доступа к сервису через веб-интерфейс и Telegram-бота.

## 1.2 Функции и возможности

### 1.2.1 Сбор и обработка данных о погоде

#### 1.2.1.1 Выбор источников данных

Идентификатор	Требование
F_DC_Src	Администратор должен иметь возможность добавлять, удалять и редактировать источники данных о погоде, а также изменять настройки извлечения данных для каждого источника посредством конфигурационных файлов источников данных. Поддерживаемые типы источников данных о погоде: веб-сайт (парсинг) и REST API (JSON или XML).
F_DC_Src_Parsing	Парсинг веб-сайтов: поддержка различных структур HTML. Администратор должен иметь возможность указать: <ul style="list-style-type: none"><li>• URL-адрес веб-страницы с данными (может содержать подстановочные знаки под параметры, указывающие на населенный пункт и/или дату и время);</li><li>• CSS-селекторы (в соответствии с синтаксисом селекторов CSS3) для извлечения данных из полей вывода (элементов страницы, содержащих искомые данные) о следующих показателях, доступных на веб-сайте:<ul style="list-style-type: none"><li>• текущая температура;</li><li>• ощущаемая температура;</li><li>• скорость ветра;</li></ul></li></ul>

	<ul style="list-style-type: none"> <li>● направление ветра;</li> <li>● влажность;</li> <li>● давление;</li> <li>● осадки (тип и количество);</li> <li>● облачность.</li> </ul> <p>Система извлекает данные из полей вывода (элементов страницы, содержащие искомые данные), найденных в коде полученной веб-страницы по указанным CSS-селекторам.</p> <p>Данные могут предоставляться на 24 часа и на 3 дня (либо другой срок, предоставляемый источниками данных, до 7 дней).</p> <p>Указываемые CSS-селекторы должны поддерживаться библиотекой BeautifulSoup версии не ниже 4.11.</p>
F_DC_Src_REST	<p>Поддержка REST API. Администратор должен иметь возможность указать:</p> <ul style="list-style-type: none"> <li>● URL адрес API (может содержать подстановочные знаки под параметры, указывающие на населенный пункт и/или дату и время);</li> <li>● параметры HTTP-запроса для получения данных: <ul style="list-style-type: none"> <li>● строки дополнительных заголовков HTTP-запроса (опционально; может содержать подстановочные знаки под параметры, указывающие на населенный пункт и/или дату и время);</li> <li>● тип запроса: GET, POST или произвольный;</li> <li>● содержимое тела запроса (не допускается при типе запроса GET; опционально; может содержать подстановочные знаки под параметры, указывающие на населенный пункт и/или дату и время);</li> </ul> </li> <li>● тип структуры данных: JSON или XML;</li> <li>● пути к полям данных в структуре JSON или XML ответа (указанным в F_DC_Src_Parsing), предоставляемым источниками данных.</li> </ul> <p>Программа извлекает данные из полей JSON или XML ответа по указанным путям к полям JSON или XML. Имена полей указываются с учетом регистра символов.</p>
F_DC_Src_REST_Path_JSON	<p>Путь к каждому полю JSON указывается в виде последовательности имен строковых ключей объектов (в двойных верхних кавычках “лапках”) и/или числовых индексов массивов (без кавычек), разделенных угловой скобкой вправо “&gt;”: объект может содержать вложенные объекты и/или массивы.</p> <p>Примеры путей:</p> <pre> "field1"&gt;"fieldA"&gt;0&gt;"myFieldB1" 12&gt;"myField1"&gt;"myFieldA"&gt;3 </pre>

F_DC_Src_REST_Path_XML	<p>Путь к каждому полю XML указывается в виде последовательности имен тегов и в конце – атрибута. Имена тегов заключаются в двойные верхние кавычки “лапки”, разделяются угловой скобкой вправо “&gt;”. Если данные находятся не в атрибуте, а между открывающим и закрывающим тегом, то последняя угловая скобка должна быть последним символом строки, указывающей путь к полю.</p> <p>Примеры путей:  <code>"tag1"&gt;"tag-a"&gt;"my-tag-b1"&gt;</code>  <code>"tag1"&gt;"tag-b"&gt;"my-tag-d1"&gt;"attribute-1"</code></p>
F_DC_Src_PS	<p>Система должна предоставлять пользователям возможность выбора из источников данных, известных системе (в результате добавления администратором конфигурационных файлов получения данных), и указания приоритета для каждого источника данных. Приоритет представляет собой целочисленное значение в диапазоне от 1 до 10. Значение приоритета по умолчанию – 5.</p> <p>Статусы выбора источников данных и приоритеты сохраняются в системе для каждого пользователя, могут быть изменены пользователем.</p>
F_DC_Src_5	<p>Должна быть возможность проверки настроек парсинга и API (проверка успешности подключения и извлечения данных).</p> <p>Система производит тестовый запрос для получения данных согласно выбранному файлу конфигурации источника данных.</p> <p>В случае успеха система сообщает об успешном соединении и получении данных. В случае ошибки система осуществляет действия, указанные в пункте F_DC_Get_3.</p>

### 1.2.1.2 Получение данных

F_DC_Get_1	Система должна автоматически получать данные о погоде из выбранных источников с заданной периодичностью (например, каждые 30 минут). Периодичность задается администратором в файле конфигурации системы.
F_DC_Get_2	Должна быть возможность ручного обновления данных по требованию администратора. При этом период автоматического получения данных не сбрасывается.
F_DC_Get_3	Система должна обрабатывать ошибки при получении данных (такие как недоступность сайта, изменение структуры HTML, ошибки API) и логгировать их в файл журнала.

### 1.2.1.3 Суммирование данных

F_DC_Sum_1	Система должна суммировать данные о погоде из разных источников с учетом выбранных пользователем приоритетов.
F_DC_Sum_2	При наличии противоречивых данных, приоритет отдается источнику с более высоким приоритетом.
F_DC_Sum_3	Должны быть предусмотрены алгоритмы усреднения числовых значений (температура, скорость ветра, осадки).
F_DC_Sum_4	Для нечисловых значений (например, тип осадков) должен использоваться алгоритм выбора наиболее часто встречающегося значения.

## 1.2.2 Веб-интерфейс

### 1.2.2.1 Отображение погоды

F_WS_WD_1	<p>Отображение доступных показателей текущей погоды в выбранном городе, перечисленных в F_DC_Src_Parsing.</p> <p>Графическое представление погодных условий в виде иконок, приблизительно соответствующих полученным погодным показателям.</p> <p>Для осадков:</p> <ul style="list-style-type: none"><li>● солнце/полумесяц (ясно);</li><li>● солнце/полумесяц с маленьким облаком рядом (облачно);</li><li>● облако (пасмурно, без осадков);</li><li>● облако с каплями дождя (дождь);</li><li>● облако со снежинками (снег).</li></ul> <p>Иконка скорости ветра накладывается поверх иконки осадков к нижнему левому углу изображения:</p> <ul style="list-style-type: none"><li>● 2 кривые линии (слабый ветер);</li><li>● 4 кривые линии (сильный ветер).</li></ul>
F_WS_WD_2	<p>Отображение доступных данных прогноза погоды на 3 дня или другой предоставляемый источниками данных период времени (не более 7 дней):</p> <ul style="list-style-type: none"><li>● минимальная и максимальная температура на каждый день;</li><li>● вероятность осадков;</li><li>● иконки погодных условий.</li></ul>
F_WS_WD_3	<p>График температуры на сутки для выбранного населенного пункта:</p> <ul style="list-style-type: none"><li>● на 24 часа;</li><li>● на 3 дня (или другой временной период до 7 дней, данные о котором предоставляет используемый погодный API).</li></ul> <p>Интервал точек графика на 24 часа – 1 час или больший, установленный в системе, до 6 часов. Интервал точек графика на более продолжительный период («на 3 дня») – 6 часов или больший интервал, предоставляемый погодным API, до 24 часов.</p>

F_WS_WD_4	<p>Вывод информации об источниках, из которых получены данные:</p> <ul style="list-style-type: none"> <li>• название источника;</li> <li>• выбранный приоритет;</li> <li>• один или два показателя из перечисленных в F_DC_Src_Parsing с возможностью переключения между ними (кнопками со стрелками влево и вправо в интерфейсе);</li> <li>• логотип источника, если доступен, иначе – иконка по умолчанию, предусмотренная внутри системы.</li> </ul> <p>В интерфейсе для вывода информации об источниках должно быть предусмотрено модальное окно с возможностью закрытия по кнопке с подписью “X” или «Закрыть».</p>
-----------	--

#### 1.2.2.2 Выбор населенного пункта

F_WS_SC_Menu_1	Поиск города (населенного пункта) по названию. Возможность выбора населенного пункта из списка, предоставляемого системой. Соответствующее меню выбора отображается по нажатию на элемент интерфейса с подписью «Выбрать город».
F_WS_SC_Menu_2	Меню выбора населенного пункта должно быть выполнено в виде модальной формы (отображаемой поверх интерфейса приложения) с прокручиваемым списком населенных пунктов с возможностью выбора и кнопками: «Выбрать» и «Отменить», либо в виде выпадающего списка с выбором одного из пунктов по нажатию на него (подобно стандартному HTML-тегу “select”).
F_WS_SC_Menu_3	Должна быть возможность закрытия меню без выбора какого-либо пункта (т. е. отмены); если меню выполнено в виде модальной формы, то закрытие должно осуществляться по нажатию соответствующей кнопки с подписью “X” или «Закрыть».
F_WS_SC_Common_1	Возможность выбора города по умолчанию с сохранением в Cookies либо в хранилище Local Storage в браузере. Город по умолчанию выбирается при открытии веб-интерфейса; если город по умолчанию не выбран, то выбирается город Москва.
F_WS_SC_Common_2	Сохранение списка избранных городов (населенных пунктов).

#### 1.2.2.3 Внешний вид, персонализация и совместимость

F_WS_APC_MU	<p>Возможность выбора пользователем единиц измерения:</p> <ul style="list-style-type: none"> <li>• температура: Цельсий, Фаренгейт, Кельвин;</li> <li>• скорость ветра: м/с, км/ч, мили/ч;</li> <li>• давление: мм рт. ст., гПа.</li> </ul>
-------------	---

F_WS_APC_CS	<p>Возможность смены цветовой темы интерфейса. Веб-интерфейс предусматривает несколько цветовых схем интерфейса:</p> <ul style="list-style-type: none"> <li>• светлая;</li> <li>• темная.</li> </ul> <p>Цветовые схемы добавляются администратором и хранятся внутри системы в виде конфигурационных файлов с HEX-представлением цветов для групп и/или типов элементов интерфейса, а также отдельных элементов: цвет текста, цвет фона, цвет обводки (опционально). Допускаются дополнительные цветовые схемы помимо перечисленных в данном пункте.</p>
F_WS_APC_L	Отображаемый текст интерфейса должен быть на русском языке. Данное ограничение не применяется строго к списку населенных пунктов.
F_WS_APC_B	Сайт должен поддерживаться веб-браузерами современных версий (3-летней или меньшей давности): Mozilla Firefox и браузерами на основе движка Chromium.

### 1.2.3 Telegram-бот

F_TGB_Commands	<p>Команды бота:</p> <p><b>/start</b> – приветствие и краткое описание возможностей бота.</p> <p><b>/weather [город]</b> – получение информации о текущей погоде в указанном городе (населенном пункте) либо в городе по умолчанию.</p> <p><b>/forecast [город]</b> – получение прогноза погоды на неделю (или другой предоставляемый источниками период не больше недели) в указанном городе либо в городе по умолчанию.</p> <p><b>/city</b> – список доступных в системе городов (населенных пунктов).</p> <p><b>/city &lt;город&gt;</b> – выбор города по умолчанию для пользователя.</p> <p><b>/source</b> – список доступных в системе источников данных с идентификатором (id), статусом выбора и приоритетом.</p> <p><b>/source &lt;id&gt; &lt;приоритет&gt;</b> – выбор источника по идентификатору из списка доступных вариантов с установкой приоритета (0 – снять выбор, 1..10 – выбрать с соответствующим целочисленным приоритетом от 1 до 10).</p> <p><b>/help</b> – получение справки по командам бота.</p>
F_TGB_Display_1	Отображение погоды: форматированный текстовый вывод текущей погоды и прогноза на неделю (аналогично веб-интерфейсу, но без графиков), использование emoji для визуализации погодных условий (осадки, скорость ветра).
F_TGB_Display_2	Возможность получения информации о погоде как в городе по умолчанию, так и в выбранном городе, даже если он не установлен как город по умолчанию.



F_TGB_Save	Сохранение города по умолчанию для каждого пользователя с возможностью смены города по умолчанию.
------------	---

#### 1.2.4 Рекомендации по одежде и аксессуарам

F_R_ML	В системе должна применяться модель машинного обучения, обученная на данных о том, какую одежду и аксессуары носили люди в различных погодных условиях.
F_R_ML_Data	Данные для обучения модели должны включать: <ul style="list-style-type: none"> <li>• температуру;</li> <li>• влажность;</li> <li>• осадки;</li> <li>• скорость ветра;</li> <li>• время суток.</li> </ul>
F_R_ML_Result_Format	Модель должна быть способна выдавать рекомендации по: <ul style="list-style-type: none"> <li>• верхней одежде (куртка, пальто, плащ);</li> <li>• головному убору (шапка, шляпа, кепка);</li> <li>• обуви (ботинки, кроссовки, сапоги);</li> <li>• аксессуарам (зонт, солнцезащитные очки, перчатки, шарф).</li> </ul>
F_R_ML_Result_Data	Система должна использовать текущие погодные данные и обученную модель машинного обучения для формирования рекомендаций. Рекомендации должны быть представлены в виде понятного текста. Рекомендации выдаются автоматически вместе с прогнозом погоды на текущий день.

#### 1.2.5 Учет пользователей

F_U_Data	В системе хранятся данные пользователей: <ul style="list-style-type: none"> <li>• тип пользователя: “веб-версия” или “Telegram-бот”;</li> <li>• идентификатор (в соответствии с F_U_Data_Id);</li> <li>• текстовый комментарий: <ul style="list-style-type: none"> <li>• веб-версия: User-Agent браузера;</li> <li>• Telegram-бот: никнейм в Telegram;</li> </ul> </li> <li>• дата и время первого использования системы;</li> <li>• дата и время последнего использования системы.</li> </ul>
F_U_Data_Id	В системе хранятся идентификаторы пользователей системы. В веб-версии идентификатор представляет собой псевдослучайное число, генерируемое при первом заходе пользователя на сайт; оно сохраняется внутри системы и в браузере пользователя (в Cookies либо

	<p>в хранилище Local Storage). Идентификаторы веб-версии действительны 6 месяцев с момента последнего использования системы.</p> <p>В Telegram-боте идентификатор соответствует id чата бота с пользователем в мессенджере Telegram.</p>
--	--

### 1.2.6 Администрирование

F_A_Log_Stat	<p>Администратор должен иметь возможность просматривать логи ошибок (файл журнала) и статистику использования сервиса.</p> <p>Система сохраняет статистику об использовании сервиса за каждые сутки:</p> <ul style="list-style-type: none"> <li>• количество запросов данных о погоде от пользователей веб-версии;</li> <li>• количество запросов данных о погоде от пользователей Telegram-бота.</li> </ul>
F_A_Config_Update	Администратор должен иметь возможность вносить изменения в настройки системы (конфигурационные файлы).
F_A_Config_Apply	<p>В системе должна быть возможность по требованию администратора обновлять настройки системы по актуальному содержимому файлов конфигурации без обязательного перезапуска сервиса.</p> <p>Данное требование не обязательно для:</p> <ul style="list-style-type: none"> <li>• настраиваемых параметров взаимодействия с СУБД;</li> <li>• IP-адреса и порта вещания сервера.</li> </ul> <p>Вышеперечисленные параметры могут быть изменены после перезапуска сервера.</p>

### 1.3 Нефункциональные требования

NF_P_DAR	Предусмотреть защиту от несанкционированного доступа к данным (разграничение прав доступа).
NF_P_Attacks	Защита от XSS-атак и SQL-инъекций.
NF_A_UX	Интерфейс веб-версии и Telegram-бота должен быть интуитивно понятным и удобным для пользователя.

### 1.4 Технологии

Языки программирования: Python.

Веб-фреймворк: Flask, Django или другой подходящий фреймворк.

Библиотеки для парсинга: BeautifulSoup.

Библиотеки для работы с API: Requests.

Библиотеки для машинного обучения: Scikit-learn, TensorFlow, PyTorch и/или другие.

База данных: PostgreSQL или MySQL.

Telegram Bot API: Python Telegram Bot.

Библиотеки для фронтенда: Vue.js, React.js и/или другие подходящие, либо без использования библиотек для ускорения построения фронтенда.