Ханты-Мансийского Автономного округа – Югры

«СУРГУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Политехнический институт

Кафедра информатики и вычислительной техники

Курсовая работа

По дисциплине «Методы и средства проектирования информационных

систем»

Выполнил:

студент группы 607-01

Шнайдер В.А.

Проверяет:

Преподаватель кафедры ИВТ

Столбов Д.А.

Сургут, 2024

БУ ВО ХАНТЫ-МАНСИЙСКОГО АВТОНОМНОГО ОКРУГА – ЮГРЫ

«СУРГУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра информатики и вычислительной техники

**Задание выполнение учебной практики, курсового проекта**

**бакалавру гр.\_\_\_607-01\_\_\_**

**направления 09.03.02 «Информационные системы и технологии»**

**Ф.И.О.\_\_\_\_\_\_\_\_\_\_\_\_\_Шнайдер Владислав Алексеевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Тема:** Система мониторинга версий приложений

**Целевая установка:** Разработка системы проверки версий программного обеспечения.

**Исходные данные:** книги, интернет ресурсы.

**Начало НИР:** 08.09.2023.

**Окончание НИР:** 30.12.2023

**Содержание работы**

Курсовая работа содержит 25 страниц, 6 рисунков, 4

таблицы, 7 используемых источников, \_\_ приложения. В неё входят:

введение, цель, задачи, актуальность, описание предметной области, диаграммы, обзор аналогов,

и заключение.

**Отчетный материал**

1. Отчет
2. Презентация

**Подписи:**

Задание получил студент \_\_\_\_\_Шнайдер В.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель Проекта \_\_\_\_\_Столбов Д.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Зав. Кафедрой \_\_\_\_\_Лысенкова С. А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РЕФЕРАТ**

Курсовая работа содержит 25 страниц, 6 рисунков, 4 таблицы, 7 используемых источников, \_\_ приложения.

**Разработка системы проверки версий программного обеспечения, на безопасной и актуальность.**

**Ключевые слова:** Программное обеспечение, мониторинг, сбор информации, версия ПО, информационные системы, уровень угрозы, безопасноть.

**Цель работы:** Разработка системы проверки версий программного обеспечения.

**Основные задачи:**

1. **Разработка пользовательского интерфейса (UI)**: Создание удобного интерфейса для удобного использования пользователем.
2. **Разработка алгоритма поиска актуальных версий программного обеспечения (ПО)**: Создание алгоритма, который будет искать официальные сайты и получать актуальную версию ПО.
3. **Разработка алгоритма поиска уровня угрозы версии ПО**: Создание алгоритма, который будет получать информацию с сайта “Банк угроз” о название продукта, его версии и уровня угрозы.

**Содержание**

[Словарь терминов 5](#_Toc1)

[Введение 6](#_Toc2)

[Основная часть 7](#_Toc3)

[**Постановка Задачи**  7](#_Toc4)

[**Описание предметной области** 8](#_Toc5)

[**Обзор Аналогов** 10](#_Toc6)

[**Требования к системе** 11](#_Toc7)

[**Проектирование Системы** 14](#_Toc8)

[**Разработка информационной системы** 18](#_Toc9)

[Заключение 19](#_Toc10)

[Список использованной литературы 20](#_Toc11)

# **Словарь терминов**

Таблица 1.

|  |  |  |
| --- | --- | --- |
| **Термин** | **Определение** | **Синонимы** |
| Система мониторинга | Система, предназначенная для наблюдения и контроля за определенными явлениями или процессами. | Система наблюдения, система контроля |
| Версия приложения | Конкретная идентифицирующая метка или номер, присвоенный определенной версии программного приложения. | Релиз, обновление |
| Информационная безопасность | Защита информации от несанкционированного доступа, использования, раскрытия, изменения или уничтожения. | Безопасность информации |
| Разработка информационной системы | Процесс создания информационной системы, включающий анализ требований, проектирование, программирование, тестирование и внедрение. | Разработка ИС, разработка ПО |
| Мониторинг версий | Процесс отслеживания изменений и обновлений версий программного приложения. | Контроль версий, отслеживание обновлений |
| Функциональные возможности | Характеристики и возможности системы или приложения, определяющие его функциональность и способность выполнять определенные задачи. | Функциональные характеристики, возможности |
| Безопасность | Состояние защищенности от угроз и рисков, связанных с несанкционированным доступом, вмешательством или повреждением. | Защита, безопасность данных |
| ПО (программное обеспечение) | Комплекс программ, выполняющих определенные функции на компьютере или другом устройстве. | Софт, приложение |
| Информационная система | Система, предназначенная для сбора, хранения, обработки, передачи и использования информации. | Система обработки данных, информационный комплекс |
| Доверенная зона | Область системы, считающаяся безопасной | Безопасная зона, защищенная зона |
| Уровень угрозы | Оценка степени опасности для системы или данных | Степень опасности, риск |

# **Введение**

В наше время пользователи сталкиваются с огромным количеством установленных приложений на своих устройствах — от мобильных телефонов до персональных компьютеров. Столь разнообразные программы представляют собой важные инструменты для работы, обучения, развлечений и многого другого. Однако, с ростом числа установленных приложений возникает необходимость в эффективном управлении версиями этого программного обеспечения.

Именно в этом контексте разрабатывается программа, которая облегчит пользовательскую жизнь, предоставляя возможность управлять просмотром и установкой новых версий приложений. Основная цель — сделать этот процесс простым, удобным и безопасным для всех пользователей. Данное решение поможет пользователям всегда оставаться в курсе актуальных версий и обновлений для своих программ, а также легко устанавливать их.

# **Основная часть**

## **Постановка Задачи**

**Цель ВКР:**

Цель выпускной квалификационной работы (ВКР) - Разработка системы проверки версий программного обеспечения.

**Задачи ВКР:**

1. Исследование существующих приложений для проверки версии ПО.
2. Разработка алгоритмов для поиска актуальных версий ПО и для поиска уровня угрозы на языке программирования Python.
3. Разработка пользовательского интерфейса (UI) с использованием языка программирования C#.
4. Тестирование и отладка для обеспечения корректной работы и удовлетворения поставленных целей.
5. Оценка эффективности.

**Цель информационной системы**

Цель информационной системы - предоставить приложение, позволяющее проверять версии установленного ПО на их актуальность и безопасность для системы.

**Задачи информационной системы**

1. Предоставить полный список установленного ПО и вывести информацию их текущих версий и актуальных.
2. Поддержка интерактивного и привлекательного пользовательского интерфейса для максимального участия, и мотивации пользователей.
3. Постоянное обновление и совершенствование контента и функциональности системы в соответствии с обратной связью от пользователей и образовательных специалистов.

## **Описание предметной области**

В основе предметной области лежит желание представить сложные концепции

программирования и математики в увлекательной форме. Игра воплощает принципы

языков программирования, логических конструкций и алгоритмов, позволяя игрокам

учиться и применять эти знания на практике.

Цель - сделать учебный процесс увлекательным и интерактивным, стимулируя развитие

логического мышления и практических навыков.

**Заинтересованные лица**

• Обычные пользователи.

• Администраторы информационной системы.

**Основные задачи высокого уровня**

Таблица 2.

|  |  |  |  |
| --- | --- | --- | --- |
| **Заинтересованные лица** | **Цель высокого уровня** | **Проблемы, возможности и замечания** | **Текущее решение** |
| Администраторы информационной системы | Управление версиями лицензионных продуктов | Большое количество сотрудников, у которых надо проверить и обновить ПО | Функция обновления встроенные в ПО |
| Пользователи | Получение актуальных версий ПО | Большое количество установленных программ на компьютере пользователя, требующие обновления их версий до актуальной | Функция обновления встроенные в ПО |

**Модель предметной области** представляет собой структурированное описание основных элементов и взаимосвязей

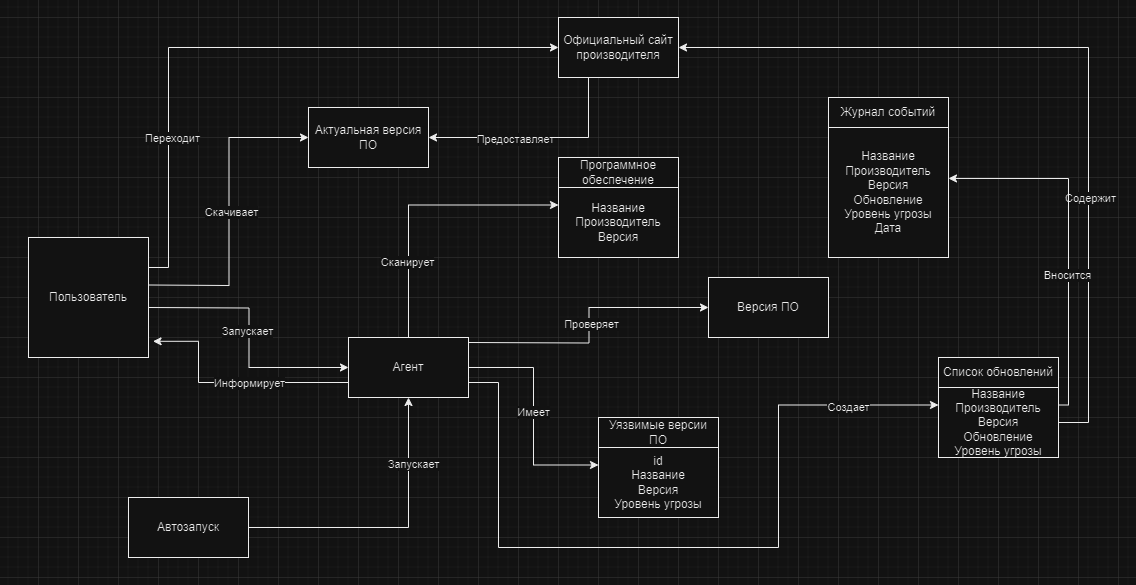


Рис 1 (Концептуальная модель предметной области)

## **Обзор Аналогов**

Таблица 3.

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерии** | **SUMo** | **UpdateStar Premium 4** | **Health-Guard** |
| Простота использования | Просто | Сложно | Просто |
| Ссылка на официальный сайт продукта или авто обновление | Отсутствует | Присутствует | Присутствует |
| Проверка уровня угрозы | Отсутствует | Отсутствует | Присутствует |
| Кроссплатформенность | Не имеется | Не имеется | Имеется |

## **Требования к системе**

**Функциональность**

1. Отслеживание версий ПО:

• Описание: Разработка и реализация механизма для отслеживания и регистрации версий программного обеспечения. • Отношение к прецедентам: Каждая версия ПО связана с прецедентами, определяющими ее особенности, изменения и связанные задачи.

2. Уведомления о новых версиях:

• Описание: Разработка системы уведомлений, которая будет информировать пользователей о доступных обновлениях и новых версиях ПО. • Отношение к прецедентам: Уведомления о новых версиях связаны с прецедентами, определяющими условия и критерии для отправки уведомлений.

3. Хранение истории версий:

• Описание: Разработка механизма для хранения истории всех предыдущих версий ПО, включая информацию о выпущенных обновлениях и изменениях. • Отношение к прецедентам: Хранение истории версий связано с прецедентами, определяющими процесс регистрации и архивирования каждой версии.

4. Оценка уровня угрозы:

Описание: Разработка системы оценки уровня угрозы, связанной с каждой обнаруженной уязвимостью. Отношение к прецедентам: Уровень угрозы связан с прецедентами, определяющими критерии оценки и категоризации (критичная, высокая, средняя, низкая).

**Подключаемые бизнес-правила:**

1. Автоматический сбор информации о версиях:

• Описание: Разработка механизма, которая будет автоматически собирать информацию о версиях ПО из выбранной системы мониторинга, включая номера версий, даты выпуска, описания изменений и другую сопутствующую информацию. • Изменение: Возможность автоматически собирать информацию о версиях ПО.

2. Уведомления о новых версиях и изменениях:

• Описание: Разработка механизма уведомления, позволяющие системе мониторинга версий ПО отправлять уведомления о новых версиях ПО и связанных с ними изменениях разработчикам, администраторам и другим заинтересованным сторонам. • Изменение: Возможность получать уведомления о новых версиях ПО.

3. Отображение и анализ информации о версиях:

• Описание: Разработка интерфейса, позволяющий пользователям просматривать и анализировать информацию о версиях ПО, включая номера версий, даты выпуска, описания изменений и другие связанные данные. • Изменение: Просмотр и анализ информации о версиях ПО.

**Интеграция:**

Интеграция с базами данных уязвимостей (CVE, NVD, OWASP).

Интеграция с системами управления ИТ-инфраструктурой (ITSM).

**Безопасность:**

Обеспечение конфиденциальности информации об уязвимостях.

Защита от несанкционированного доступа к данным.

**Удобство использования:**

Интуитивный пользовательский интерфейс: Описание: Создание удобного и интуитивно понятного пользовательского интерфейса (UI), обеспечивающего простоту навигации и взаимодействия с инструментом Health-Guard. Включает в себя понятные иконки, легкость освоения основных функций и минимизацию необходимости в дополнительных объяснениях. Краткие и понятные инструкции: Описание: Предоставление четких и кратких инструкций для пользователя на протяжении всего процесса мониторинга версий ПО с использованием Health-Guard. Включение подсказок и объяснений новых функций и возможностей инструмента. Быстрый доступ и загрузка: Описание: Оптимизация процесса запуска инструмента и скорости загрузки данных о версиях ПО с помощью Health-Guard. Обеспечение минимального времени ожидания, чтобы пользователи могли быстро получить доступ к актуальной информации о версиях. Фильтрация и сортировка информации по различным критериям (уровень угрозы, тип ПО, дата обнаружения).

**Производительность:**

Обеспечение быстрой и эффективной работы системы сканирования и оценки уязвимостей.

**Ограничения:**

Технические ограничения, связанные с доступностью информации об уязвимостях.

Финансовые и временные ограничения, связанные с разработкой и поддержкой системы.

**Программные интерфейсы:**

Интерфейс пользователя (User Interface, UI): позволяет пользователю взаимодействовать с системой мониторинга версий ПО, отображает информацию о доступных обновлениях и предлагает их установку. Интерфейс программирования приложений (Application Programming Interface, API): может использоваться для получения информации о доступных версиях ПО из внешних источников.

**Средства разработки:**

Для разработки пользовательского интерфейса будет использоваться платформа – VisualStudio с использованием инструмента WinForms.

**Бизнес-правила:**

Правило 1: Автоматическое сканирование ПО на наличие уязвимостей.

Правило 2: Определение уровня угрозы для каждой обнаруженной уязвимости.

Правило 3: Уведомление пользователей об обнаруженных уязвимостях.

Правило 4: Предоставление рекомендаций по исправлению уязвимостей.

Таблица 4.

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя бизнес-правила** | **Описание бизнес-правила** | **Возможность изменения** | **Источник** |
| Правило 1 | Автоматическое сканирование ПО на наличие уязвимостей. | Да | Внутренний |
| Правило 2 | Определение уровня угрозы для каждой обнаруженной уязвимости. | Да | Внутренний |
| Правило 3 | Уведомление пользователей об обнаруженных уязвимостях. | Да | Внутренний |
| Правило 4 | Предоставление рекомендаций по исправлению уязвимостей. | Да | Внутренний |

**Вопросы законодательства:**

Соблюдение требований к защите данных.

Соблюдение авторских прав и лицензионных условий использования ПО.

**Информация из предметной области:**

Текущие тенденции в области ИТ-безопасности.

Популярные практики и стандарты управления уязвимостями.

Требования и ожидания пользователей по поводу защиты от уязвимостей.

**Влияние на другие компоненты:**

Расширение функциональности системы мониторинга версий ПО.

Интеграция с системами ИТ-безопасности.

**Общие технические требования**

**Минимальные системные требования:**

Операционная система: Windows 10 или более поздняя версия;

## **Проектирование Системы**

**Use Case диаграмма (Рис 2)**

Далее необходимо было рассмотреть сценарий использования информационной системы. На рисунке 2 представлен сценарий ключевых аспектов использования информационной системы.

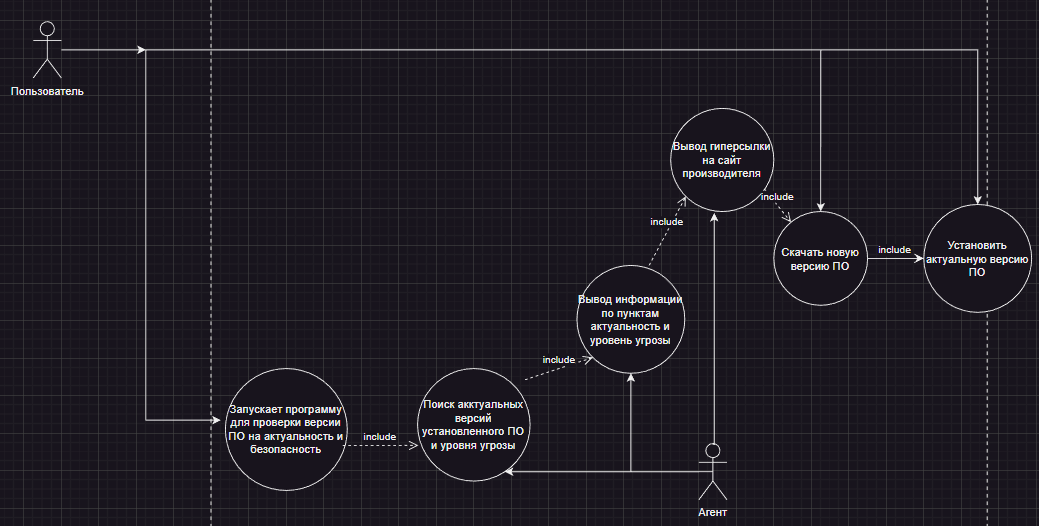


Рис 2 (Use Case диаграмма)

**Анализ прецедентов**

После рассмотрения вариантов использования информационной системы рассмотрим подробно основной успешный сценарий использования информационной системы.

**Название прецедента:**

Текущая версия ПО является уязвимой

**Участники:**

1. Пользователь
2. Программа проверки обновления ПО (часть программного обеспечения)

**Описание:**

Данный прецедент описывает ситуацию, в которой программа проверяет установленное ПО на устройстве и обнаруживает, что версия одного или нескольких приложений являются уязвимыми. Прецедент позволяет пользователю быть уверенным, что установленное ПО является актуальным и безопасным.

**Предусловия:**

1. Пользователь запустил программу проверки версии ПО.

**Основной сценарий:**

1. Пользователь запускает программу проверки версии ПО.
2. Система выводит список ПО(название, производитель, текущая версия, актуальная версия, ссылка на сайт продукта, уровень угрозы) установленное в его системе.
3. Система обнаруживает одну или несколько версий ПО, которые являются уязвимыми и уведомляет об этом пользователя, если ничего не обнаруживается, то перейти к шагу 7.
4. Пользователь принимает во внимание данное уведомление и заходит на официальный сайт разработчика ПО.
5. Пользователь скачивает либо актуальную версию ПО, либо предыдущую версию, чтобы ПО было безопасным.
6. После установки новой версии ПО, система выдаст что версия ПО актуальна и безопасна.
7. Завершение работы программы.

**Постусловия:**

1. Пользователь получает обновленную версию ПО для дальнейшего безопасного использования.

**Альтернативные потоки:**

Нет.

Этот прецедент описывает основной сценарий использования системы по мониторингу версий ПО, где пользователь взаимодействует с программным обеспечением для проверки версии ПО на актуальность и безопасность.

**Диаграммы процессов BPMN**

В процессе проектирования была сформирована BPMN-диаграмма

В ней описан следующий процесс:

1. Пользователь запускает программу проверки версии ПО.
2. Система выводит список ПО(название, производитель, текущая версия, актуальная версия, ссылка на сайт продукта, уровень угрозы) установленное в его системе.
3. Система обнаруживает одну или несколько версий ПО, которые являются уязвимыми и уведомляет об этом пользователя, если ничего не обнаруживается, то перейти к шагу 7.
4. Пользователь принимает во внимание данное уведомление и заходит на официальный сайт разработчика ПО.
5. Пользователь скачивает либо актуальную версию ПО, либо предыдущую версию, чтобы ПО было безопасным.
6. После установки новой версии ПО, система выдаст что версия ПО актуальна и безопасна.
7. Завершение работы программы.

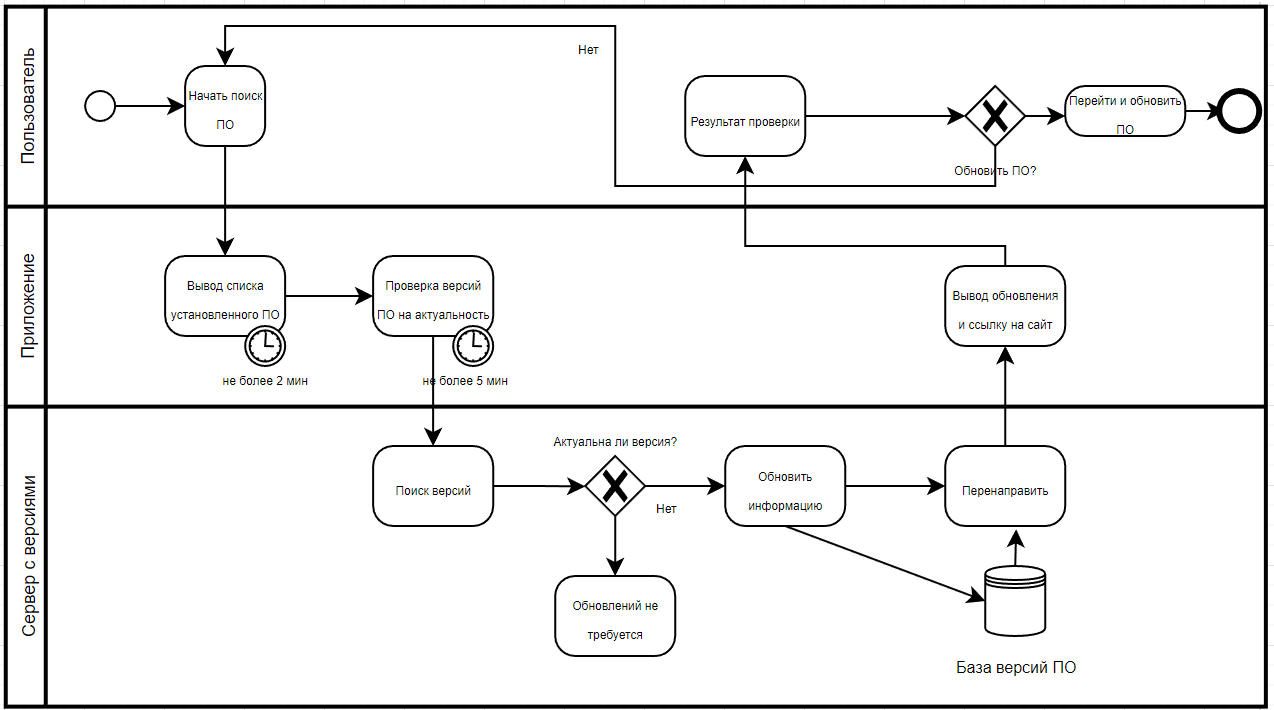


Рис 3. (BPMN диаграмма)

**Диаграмма классов**

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру (поля, методы) и типы отношений (наследование, реализация интерфейсов).

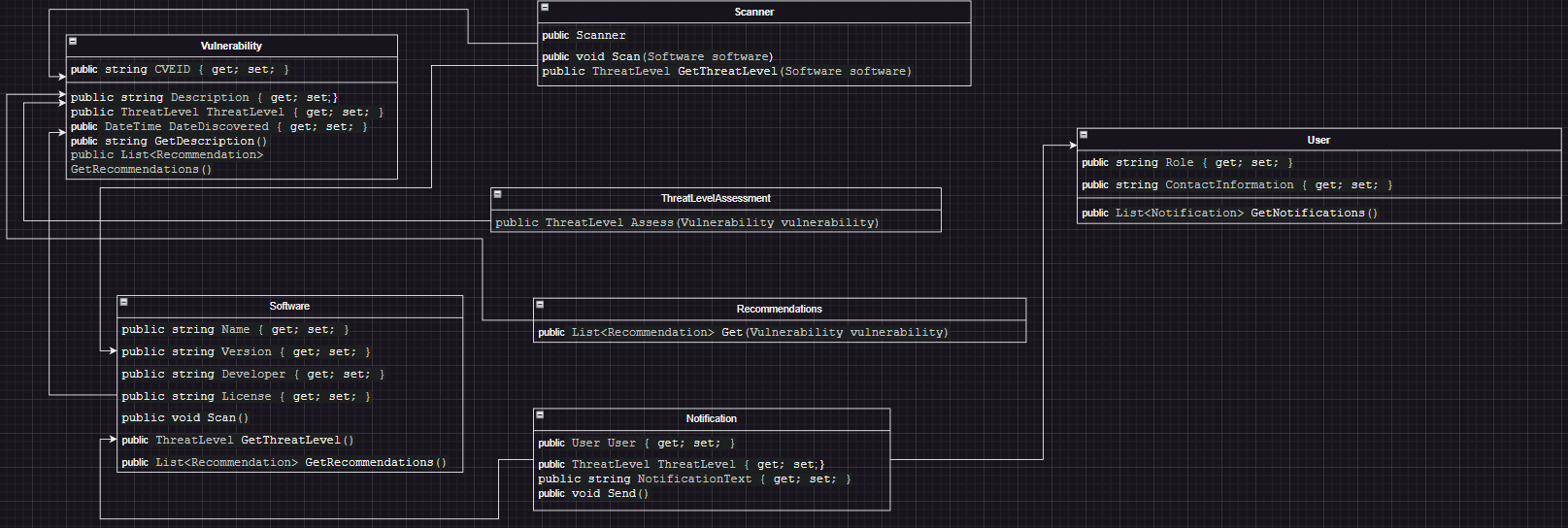


Рис 4. (Диаграмма Классов)

**Диаграмма компонентов**

Диаграмма компонентов — элемент языка моделирования UML, статическая структурная диаграмма, которая показывает разбиение программной системы на структурные компоненты и связи между компонентами. В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и т.д.

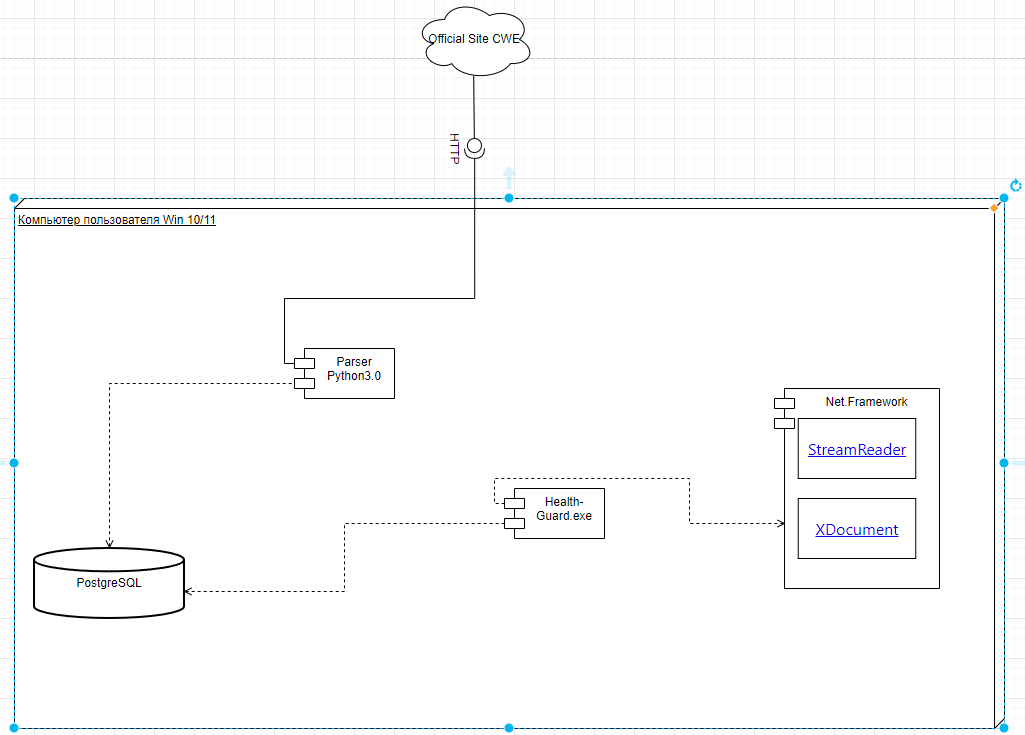


Рис 5 (Диаграмма компонентов)

## **Разработка информационной системы**

План разработки (Backlog)

Backlog – это упорядоченный по приоритету список работ, которые планируется выполнить с учетом знаний, имеющихся на данный момент.

В планах:

* Добавить алгоритм проверки уровня угрозы версии ПО

В процессе:

* Доработка алгоритма поиска актуальных версий приложений
* Функции в параметрах настройках
* Всплывающие окно уведомлений
* Пользовательский интерфейс

**Скриншоты UI**

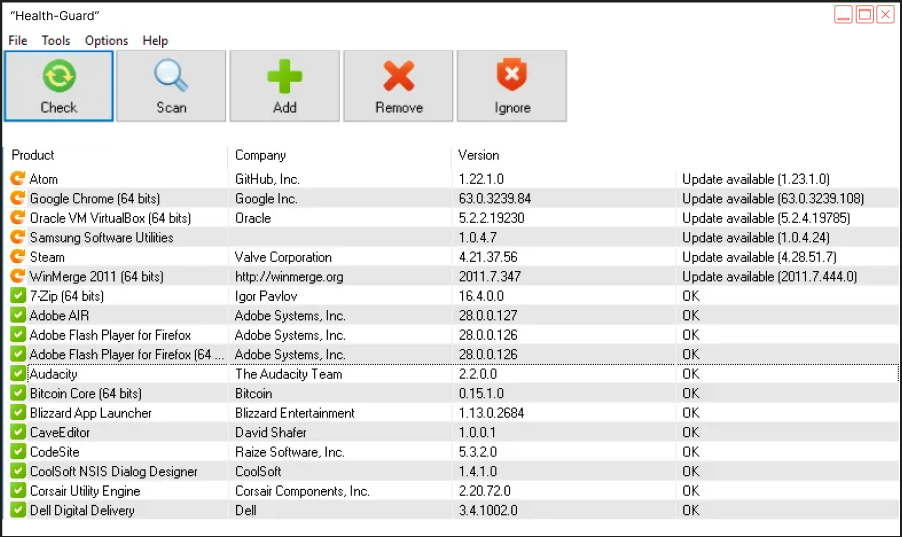


Рис 6 (Прототип UI)

# **Заключение**

# **Список использованной литературы**

1. КонтурНорматив - Приказ ФСТЭК РФ от 11.02.2013 N 17 - <https://normativ.kontur.ru/document?moduleId=1&documentId=344593>

2. ГОСТ 34.601-90 «Информационные технологии. Автоматизированные системы. Стадии создания».

3. Советов Б.Я. Информационная технология: Учебник для вузов по специальности «Автоматизированные системы обработки информации и управления», - М.: Высшая школа, 1994. - 368 с.

4. Автоматизированное рабочее место для статистической обработки данных / В.В. Шураков. - М.: Финансы и статистика, 2002. - 190 с.

5. Стрельцов А.А. Обеспечение информационной безопасности России. Теоретические и методические основы / Под ред. В.А. Садовничего и В.П. Шерстюка. – М.: МЦНМО, 2002.

6. Е.Б. Белов, В.П. Лось Основы информационной безопасности. / Москва, Горячая линия – Телеком, 2006.

7. Вихорев С.В. Информационная Безопасность Предприятий. Москва, 2006.