Ханты-Мансийского Автономного округа – Югры

«СУРГУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Политехнический институт

Кафедра информатики и вычислительной техники

Реферат

по дисциплине «Система контроля версий приложений»

Выполнил:

студент группы 607-01

Шнайдер В.А.

Проверяет:

Преподаватель кафедры ИВТ

Столбов Д.А.

Сургут, 2023

**Содержание**

[Видение 3](#_Toc154193158)

[Дополнительная спецификация 9](#_Toc154193159)

[Use case-диаграмма 12](#_Toc154193160)

[Описание прецедента 13](#_Toc154193161)

[Диаграмма компонентов 16](#_Toc154193162)

[Концептуальная модель 17](#_Toc154193163)

[Диаграмма классов 18](#_Toc154193164)

[Прототип UI 19](#_Toc154193165)

# Видение

**Краткое наименование:** Health-Guard

**Полное наименование:** Система контроля версий приложений.

**Введение**

Health-Guard - это информационная система, разработанная для эффективного контроля и управления версиями приложений. Она предназначена для обеспечения актуальности приложений, и в случае обнаружения устаревших версий предлагает их обновление. Данная система разработана с учетом потребностей организаций и пользователей, которые хотят быть в курсе последних версий программного обеспечения.

**Возможности**

Health-Guard предлагает следующие возможности:

1. Мониторинг версий приложений в режиме реального времени.
2. Генерация отчетов о доступных обновлениях.
3. Уведомления и оповещения о неактуальных версиях приложений.
4. Централизованное управление версиями приложений.
5. Интеграция с другими системами контроля версий.
6. Проверка приложений на наличие уязвимых версий.
7. Возможность добавление автоматического сканирования программ.

**Проблема**

Быть в курсе последних версий приложений является важным аспектом для организаций и пользователей. Однако, отслеживание и контроль версий может быть сложной задачей, особенно с учетом быстрого развития и обновления программного обеспечения. “Health-Guard” решает эту проблему, предоставляя эффективный механизм контроля версий приложений.

**Заинтересованные лица**

|  |  |  |
| --- | --- | --- |
| **Роль** | **Цель** | **Текущее решение** |
| Администраторы информационной системы | Управление версиями приложений | SUMo App-Version Control |
| Пользователи | Получение актуальных версий | UpdateStar Premium 4 |

**Конкуренты (Анализ рынка)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерии** | **SUMo** | **UpdateStar Premium 4** | **Health-Guard** |
| Широкий функционал | Ограниченный | Ограниченный | Широкий |
| Интуитивный интерфейс | Да | Да | Да |
| Централизованное управление версиями | Нет | Да | Да |
| Поддержка различных платформ | Да | Да | Да |
| Стоимость | Бесплатно | Платно | Гибкая модель |
| Нагрузка на систему | Средняя | Средняя | Оптимизированная |
| Обновление | Ручное | Ручное | Автоматическое |

**Расчет рынка**

**Заработок**

Продажа лицензионных ключей активации является основным источником дохода для "Health-Guard". Организации, заинтересованные в обеспечении информационной безопасности, могут приобрести ключи, что позволит вашей компании генерировать стабильный поток дохода. Ключи могут предоставляться в различных форматах, включая одноразовые, временные или многопользовательские лицензии, в зависимости от потребностей и бюджетов клиентов.

**Перспективы решения**

1. "Health-Guard" обладает конкурентными преимуществами, такими как специализация на требованиях 17-го приказа ФСТЭК, надежность и поддержка, адаптация к изменениям, гибкая модель монетизации, простота внедрения и использования, а также централизованное управление политиками безопасности.
2. С увеличением требований к информационной безопасности и обязательств государственных учреждений соблюдать требования 17-го приказа ФСТЭК, рынок продуктов, обеспечивающих повышение уровня защиты информационных систем, будет расти.
3. Предоставление гибкой модели монетизации и различных форматов лицензий позволит эффективно масштабировать бизнес и привлекать широкий спектр клиентов.

Эти факторы указывают на перспективы роста и успеха решения "Health-Guard" на рынке информационной безопасности.

Словарь терминов

|  |  |  |
| --- | --- | --- |
| **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 |

# Дополнительная спецификация

**Введение**

Проект "Health-Guard" направлен на создание инструмента для мониторинга версий программного обеспечения. Данная дополнительная спецификация описывает различные аспекты функциональности, безопасности, удобства использования и другие важные аспекты проекта.

**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.

**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.

Безопасность: Обеспечение безопасности данных о версиях ПО и защита от несанкционированного доступа.

**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.

**Надежность**

Обеспечение стабильной работы инструмента мониторинга версий ПО Health-Guard, предотвращение сбоев и ошибок.

**Производительность**

Оптимизация кода и ресурсов для обеспечения плавной и быстрой работы инструмента мониторинга версий ПО Health-Guard.

**Возможности поддержки**

Мониторинг версий ПО:

Описание: Разработка игровой системы с модульной структурой, позволяющей легко внедрять новые функции и изменения без серьезных изменений в основной кодовой базе. Это обеспечивает гибкость и удобство при внесении изменений. Разработка модуля, отвечающего за отслеживание и хранение информации о версиях программного обеспечения. Создание системы автоматического обновления, которая будет проверять наличие новых версий и предлагать их установку. Реализация функциональности отображения текущей установленной версии ПО в пользовательском интерфейсе.

Конфигурирование

Описание: Возможность выбора источника информации о новых версиях (например, официальный сайт разработчика, FTP-сервер, API). Настройка периода проверки наличия новых версий. Возможность включения или отключения автоматического обновления.

**Ограничения**

Технические ограничения:

Описание: Учитывать требования к аппаратному обеспечению, необходимые для работы мониторинга версий ПО. Обеспечить оптимальную производительность при работе с большим количеством программных компонентов и версий.

Финансовые и временные ограничения:

Описание: Учитывать бюджетные ограничения при разработке и поддержке системы мониторинга версий ПО. Планировать сроки разработки и обновлений системы с учетом временных ограничений.

**Программные интерфейсы**

Важные 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.

Аппаратные средства:

Серверы или хостинг-провайдеры для хранения информации о доступных версиях ПО и обновлений. Определение частоты проверки наличия новых версий и обновлений. Установление процедур уведомления пользователя о доступных обновлениях. Определение правил автоматического обновления и возможности пользовательского контроля над этим процессом. Разработка механизмов обработки и регистрации ошибок при обновлении.

**Средства разработки**

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

**Бизнес-правила**

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя бизнес-правила** | **Описание бизнес-правила** | **Возможность изменения** | **Источник** |
| Правило 1 | Разработка механизма для автоматического обнаружения новых версий ПО | Да | Внутренний |
| Правило 2 | Отправка уведомлений пользователям о наличии новых версий ПО | Да | Внутренний |
| Правило 3 | Отслеживание изменений и исправлений в каждой новой версии ПО | Нет | Внутренний |
| Правило 4 | Распределение и установка обновлений ПО на целевые устройства | Да | Внутренний |

**Вопросы законодательства**

При реализации системы мониторинга версий ПО необходимо учитывать соответствие следующим вопросам законодательства: Защита данных пользователей и соблюдение приватности. Соблюдение авторских прав и лицензионных условий использования ПО. Соответствие требованиям безопасности и защите от вредоносных воздействий.

**Информация из предметной области**

Для разработки системы мониторинга версий ПО полезно обладать следующей информацией из предметной области: Текущие тенденции в развитии и поставках программного обеспечения. Популярные практики и стандарты отслеживания и управления версиями ПО. Требования и ожидания пользователей по поводу мониторинга и обновлений ПО. Лицензионные условия различных ПО и требования к их использованию.

# Use case-диаграмма

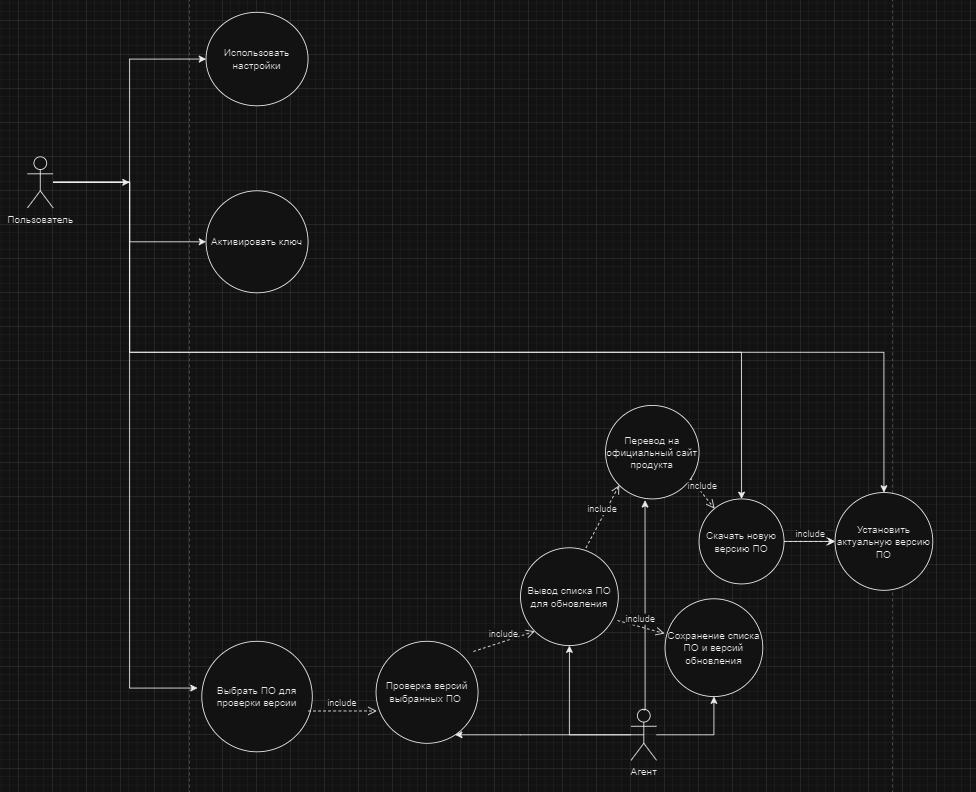


Рисунок 1. Usecase – 1 прицидент

# Описание прецедента

**Название прецедента:**

Обновление версии ПО до актуальной.

**Участники:**

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

**Описание:**

Данный прецедент описывает ситуацию, в которой пользователь запускает программу для проверки установленного ПО на наличие актуальной версии. Прецедент позволяет пользователю быть уверенным, что установленное ПО актуально.

**Предусловия:**

1. Пользователь запустил программу проверки версии ПО.

**Основной сценарий:**

1. Пользователь запускает программу проверки версии ПО.
2. Система выводит список ПО(название, производитель, версия, обновление) установленное в его системе.
3. Система сохраняет отчет о том какое ПО имеет обновление, уровень угрозы и дата создание отчета, после чего сохраняет его в xml файле.
4. Пользователь выбирает то ПО, что ему нужно обновить.
5. Пользователь нажимает обновить и программа перенаправлет его на официальный сайт разработчика ПО.
6. Пользователь скачивает новую версию ПО и устанавливает его.
7. После установки новой версии ПО, система выдаст что новой версии нету.

**Постусловия:**

1. Пользователь получает обновленнвую версию ПО для дальнейшего безопасного использования.
2. Пользователь может обновить и отсальное ПО, что выдает ему система или обновить только то, что нужно.

**Альтернативные потоки:**

Нет.

Этот прецедент описывает основной сценарий использования системы по мониторингу версий ПО, где пользователь взаимодействует с программным обеспечением для обновления своего ПО.

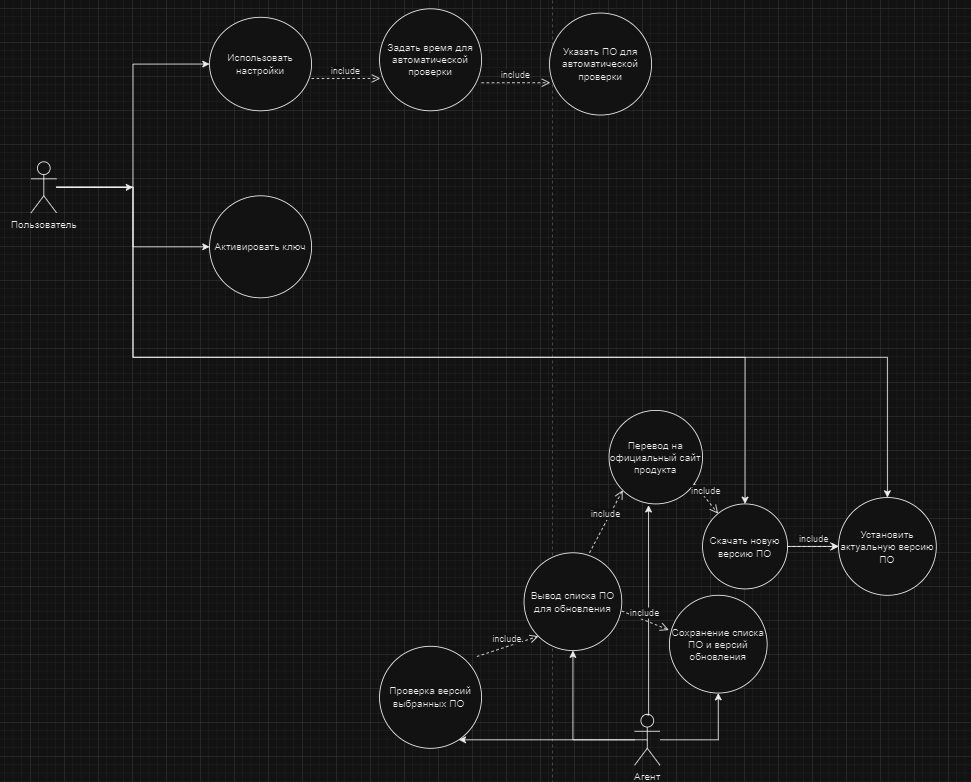


Рисунок 2. Usecase – 2 прицидент

**Название прецедента:**

Автоматическое обновление версии ПО до актуальной.

**Участники:**

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

**Описание:**

Данный прецедент описывает ситуацию, в которой пользователь добавляет время проверки обновления выбранного списка ПО на проверку их актуальных версий для дальнейшей установки и уведомляет об этом пользователя.

**Предусловия:**

1. Пользователь запустил программу проверки версии ПО.

**Основной сценарий:**

1. Пользователь запускает программу проверки версии ПО.
2. Пользователь заходит в настройки приложения и добавляет автоматическую проверку версий выбранного ПО в назначенное время.
3. Система в назначенное время выводит увидомление о том, что имеется актуальная версия ПО(если обновлений нету, то ничего не выводит).
4. Система сохраняет отчет о том какое ПО имеет обновление, уровень угрозы и дата создание отчета, после чего сохраняет его в xml файле.
5. Пользователь нажимает на уведомление и попадая на окно со списком обновлений.
6. Пользователь нажимает обновить, и программа перенаправляет его на официальный сайт разработчика ПО.
7. Пользователь скачивает новую версию ПО и устанавливает его.
8. После установки новой версии ПО, система выдаст что новой версии нету.

**Постусловия:**

1. Пользователь получает обновленную версию выбранного ПО без лишней затраты времени на поиск ПО для обновления.

**Альтернативные потоки:**

Нет.

Этот прецедент описывает побочный сценарий использования системы по мониторингу версий ПО, где пользователь выбирает нужные ему приложения и в определенное время система автоматически проверяет их и выводит уведомления, что их можно обновить.

# Диаграмма компонентов

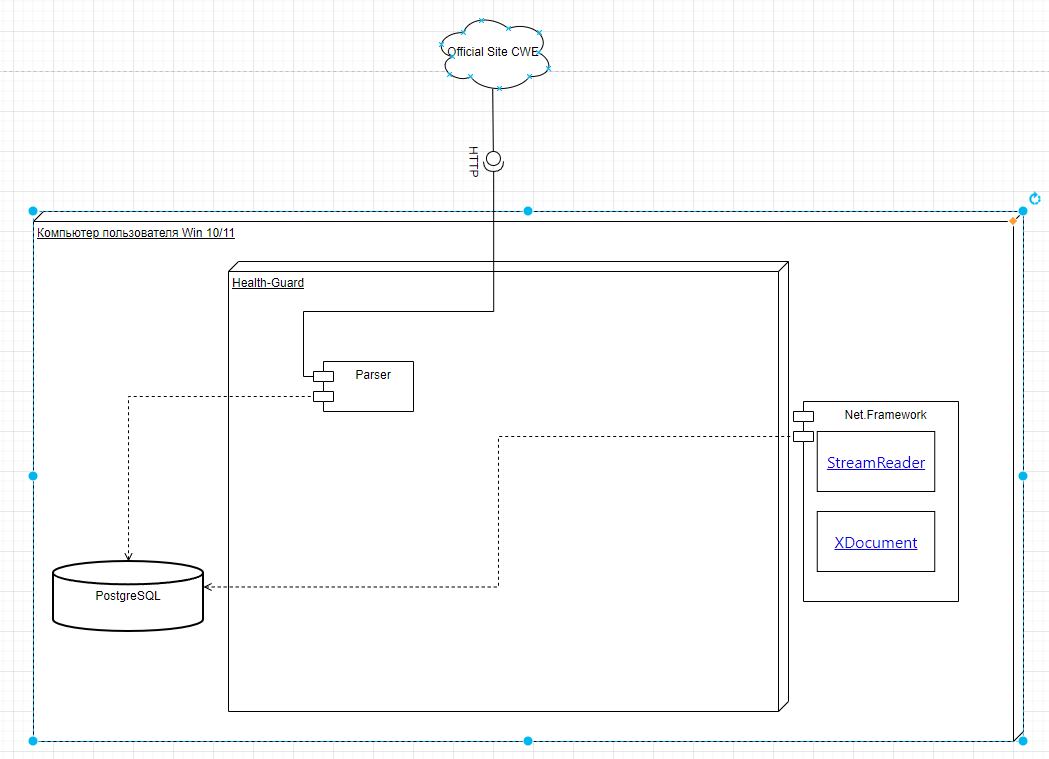


Рис.3. Диаграмма компонентов

# Концептуальная модель

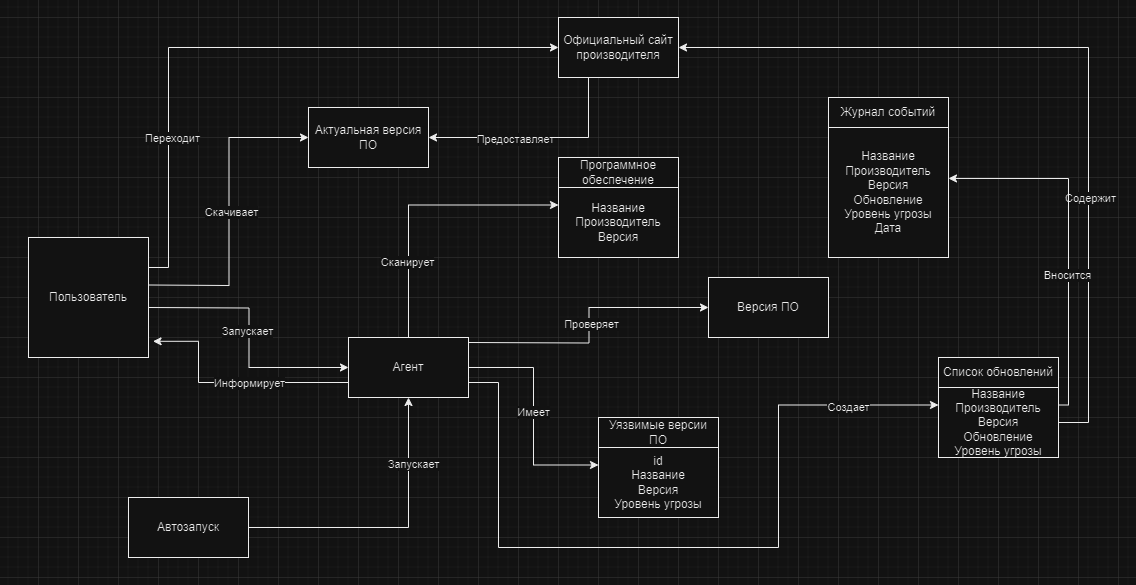
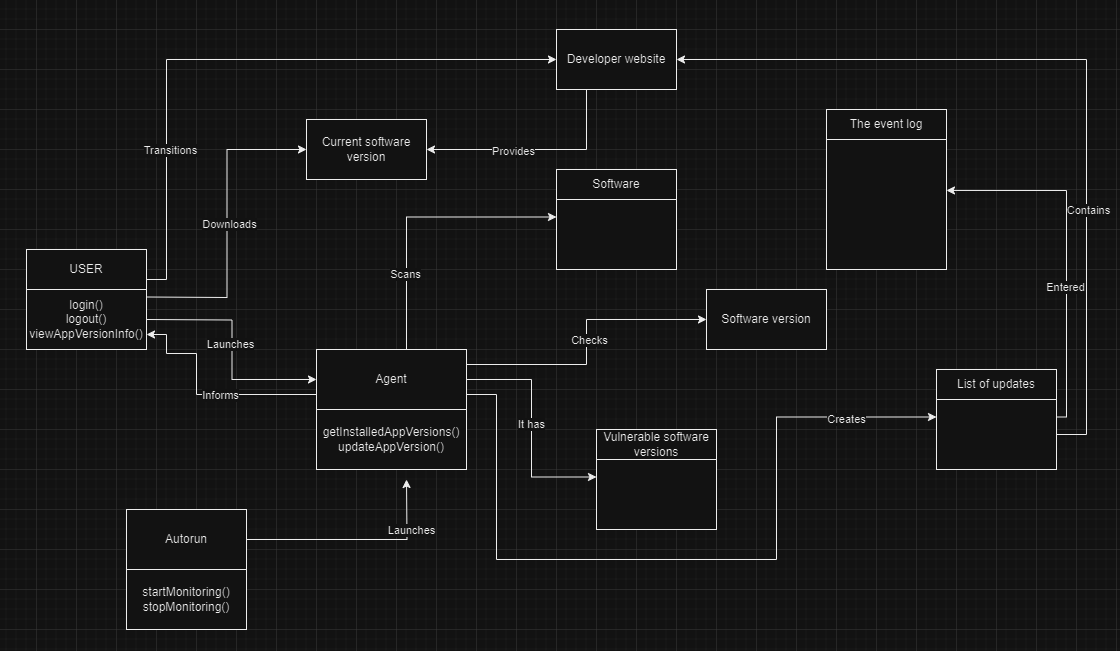


Рисунок 4. Концептуальная модель

# Диаграмма классов

  
Рисунок 5. Диаграмма классов

# Прототип UI

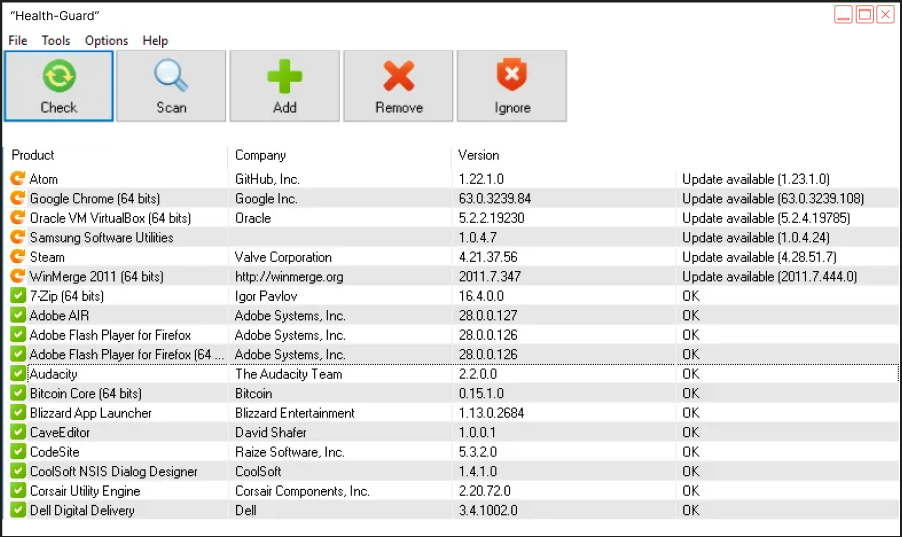


Рисунок 6. Прототип UI