

Сервис мониторинга состояния серверов

Студентка ИУ7-756 Оберган Татьяна

Научный руководитель: Ю.В. Строганов

Цель работы

Цель данной работы - создание программы, выполняющей мониторинг серверов.

Задачи:

- формализация цели;
- анализ аналогов;
- анализ и выбор способов достижения поставленной цели;
- разработка программного обеспечения, которое позволит решить задачу мониторинга серверов.

Аналоги

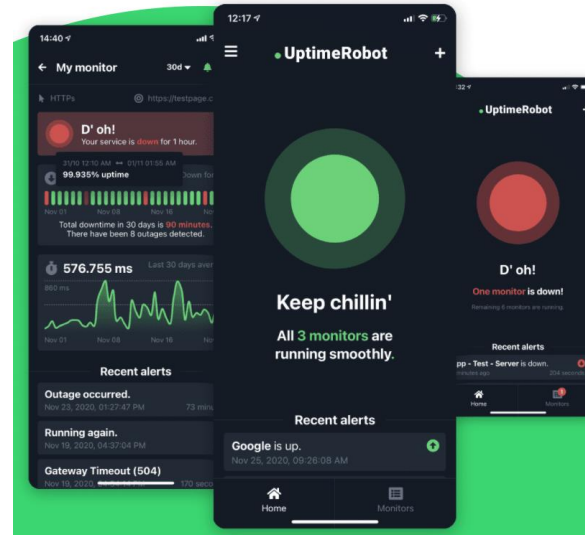
Uptime Robot

Достоинства:

- ✓ Оповещения
- ✓ Логирование
- ✓ Мобильное приложение

Недостатки:

- ❖ Платная подписка
- ❖ Не предоставляет информацию о серверных ресурсах и процент выполнения задачи



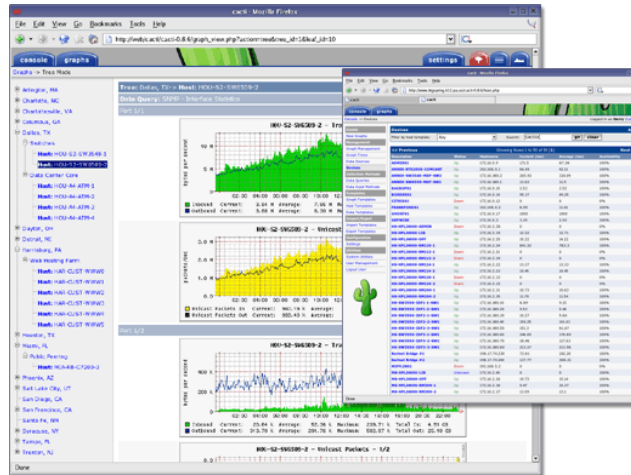
Cacti

Достоинства:

- ✓ Открытый исходный код
- ✓ Поддержка циклический баз данных
- ✓ Статистика по серверным ресурсам

Недостатки:

- ❖ Высокий порог вхождения
- ❖ Требуется MySQL, Apache



Виды систем мониторинга

1. системы, которые устанавливаются непосредственно на сервер;

- ✓ Простота администрирования и настройки
- ❖ Влияет на производительность сервера

2. клиент-серверные системы мониторинга.

- ✓ Надёжный мониторинг
- ✓ Сохранность ресурсов сервера
- ❖ Сложность администрирования и настройки

Передача данных

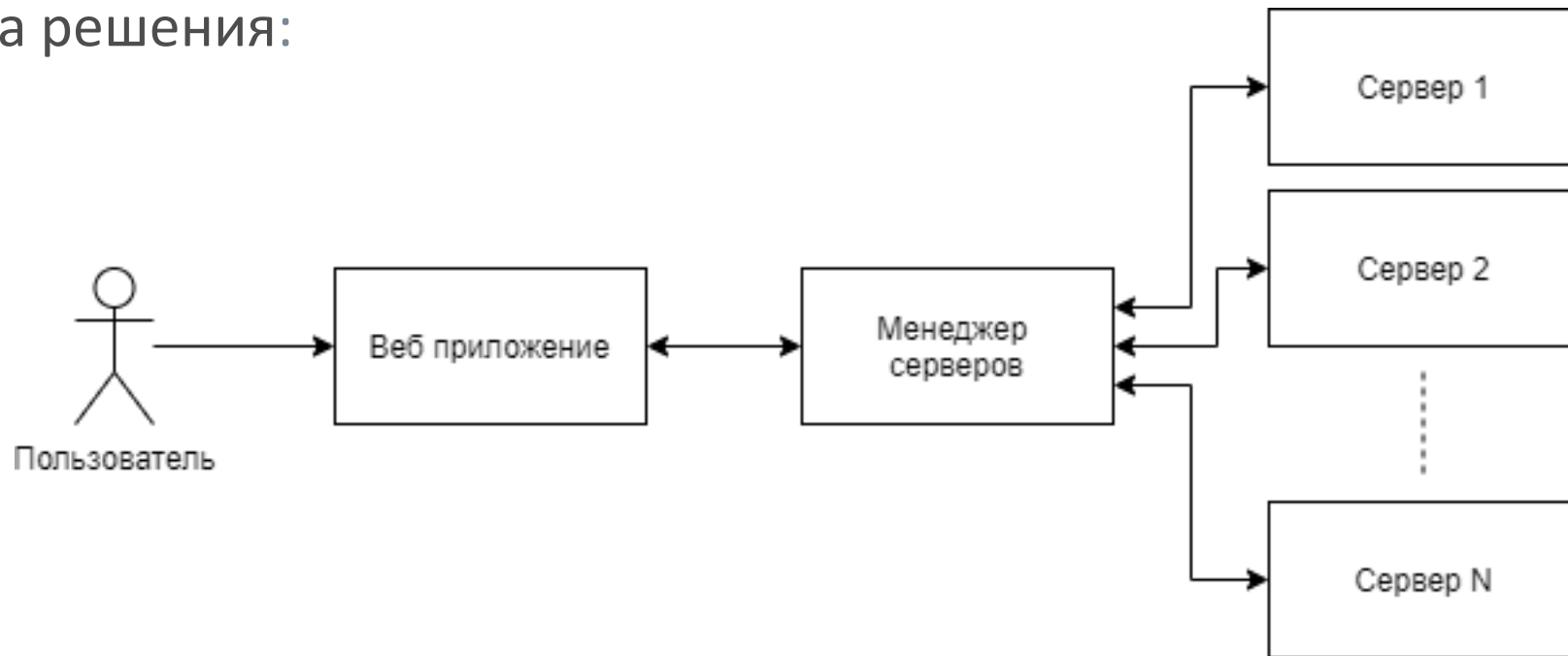
- HTTP – протокол прикладного уровня передачи произвольных данных
 - Клиенты – отправляют запросы
 - Серверы – обрабатывают запросы
 - Прокси – посредники для выполнения транспортных служб
- SNMP – протокол для управления устройствами в IP-сетях на основе архитектур TCP/UDP. Используется в системах сетевого управления для контроля подключённых к сети устройств. (маршрутизаторы, коммутаторы, серверы, и т.д.)

Требования к программе

- опрос всех серверов для обновления их статуса в системе;
- добавление нового сервера в пул серверов;
- удаление сервера из пула серверов;
- отправление запроса выбранному серверу;
- отправление запроса на автоматически выбранный сервер;
- мониторинг состояния запроса.

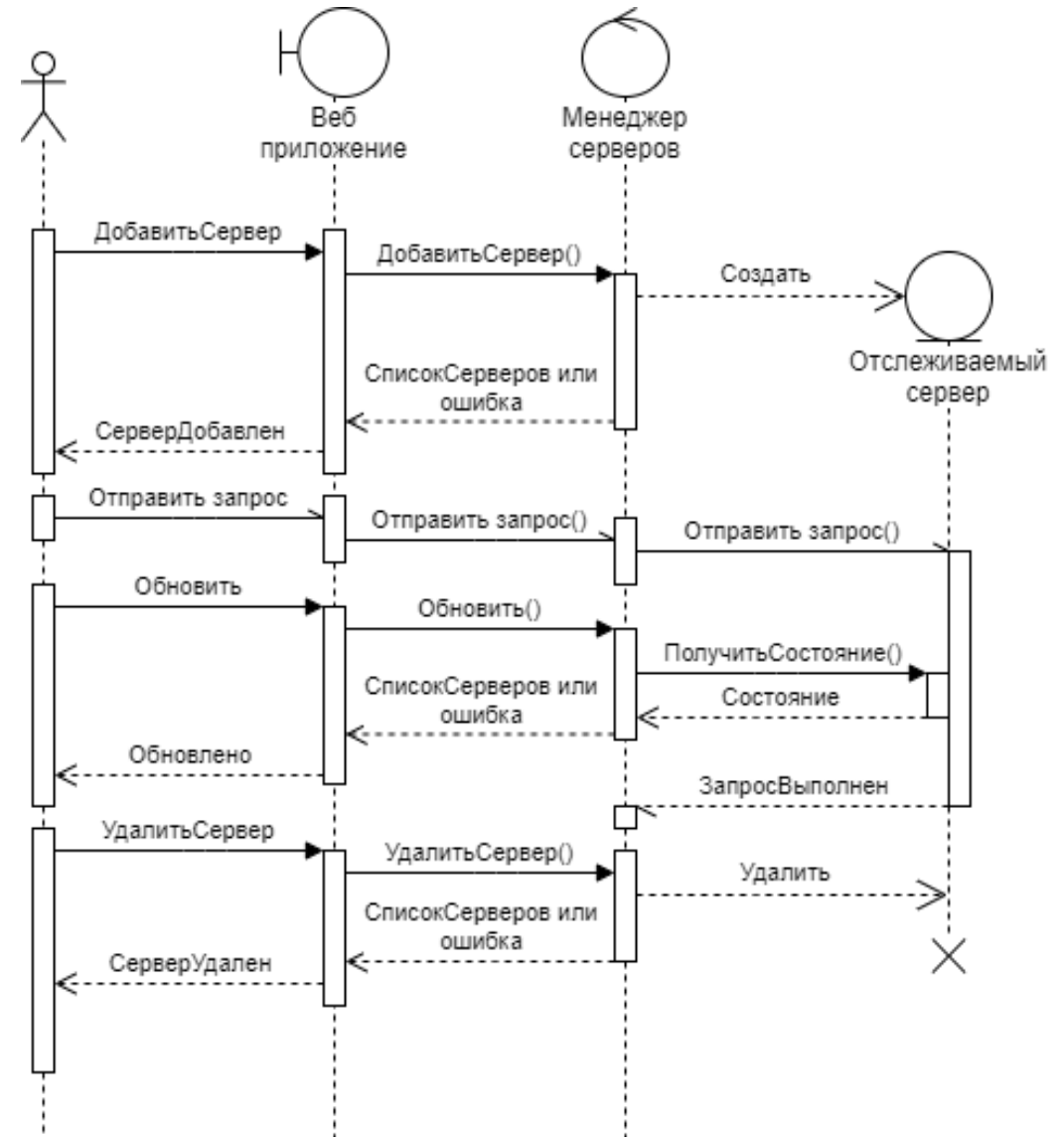
Взаимодействие компонентов

Структура решения:



Взаимодействие КОМПОНЕНТОВ

На рисунке справа
представлена диаграмма
последовательностей



Use-case диаграмма



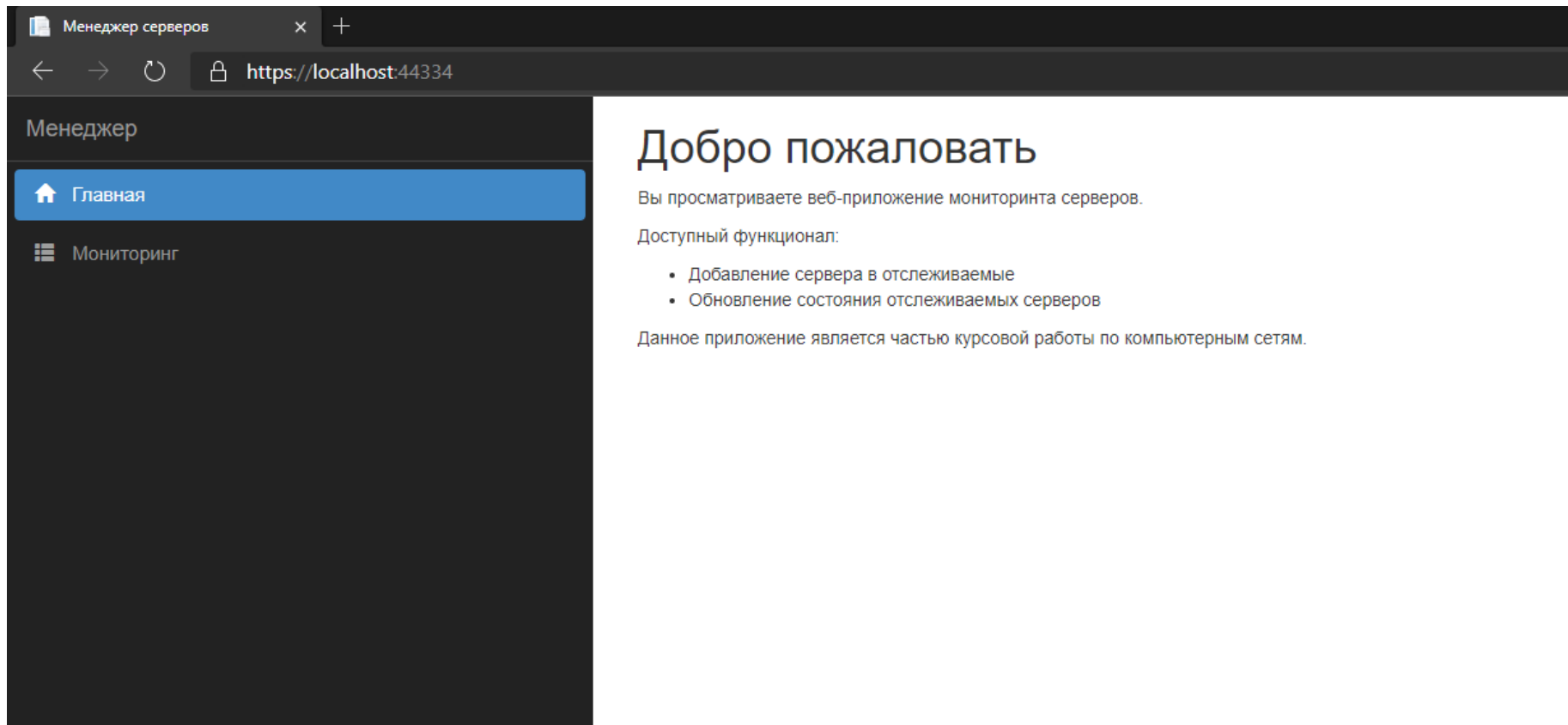
Технологический стек

Веб приложение: Angular

Менеджер серверов: C#

Отслеживаемый сервер: Swi-Prolog

Интерфейс. Главная страница



Интерфейс. Сервер занят

Мониторинг серверов

Данная страница демонстрирует возможности менеджера серверов.

<input type="text" value="http://localhost:1234/"/>	<input type="button" value="Добавить"/>	<input type="button" value="Удалить"/>	<input type="button" value="Обновить состояние серверов"/>	<input type="text"/>	<input type="button" value="Отправить запрос"/>		
Название	Адрес	Состояние	Прогресс	Номер текущей задачи	Общее число задач	Запрос	Ответ
swipl	http://localhost:1234/	Busy	10	95887	1000000	fact/1000000	

Интерфейс. Запрос выполнен

Мониторинг серверов

Данная страница демонстрирует возможности менеджера серверов.

Название	Адрес	Состояние	Прогресс	Номер текущей задачи	Общее число задач	Запрос	Ответ
swipl	http://localhost:1234/	Free	100	5	5	fact/5	{"answer":120}
swipl	http://localhost:1235/	Free	0	0	0		

Заключение

Во время выполнения курсового проекта были достигнуты поставленные цель и задачи:

- формализована цель;
- проанализированы аналоги;
- выбран способ достижения поставленной цели;
- разработана программа, которая позволяет решить задачу мониторинга серверов.

Спасибо за внимание

Москва, 2020