

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет «СТАНКИН»**

**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

|  |  |
| --- | --- |
| **Институт**  **информационных**  **систем и технологий** | **Кафедра**  **информационных систем** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Махмудов Бобурбек Нодирбекович** | | | |
| Тема: «**Исследование методов повышения производительности систем веб-шаблонов и разработка системы шаблонизаци на их основе**» | | | |
| **Выпускная квалификационная работа на присвоение квалификации «бакалавр» по направлению 09.03.02 «Информационные системы и технологии»** | | |
|  | Регистрационный № \_\_\_\_\_\_\_ | |
| **Заведующий кафедрой**  **д.т.н., проф.** |  | **Позднеев Б.М.** |
|  | подпись |  |
| **Руководитель**  **к.т.н., доцент** |  | **Бумарин Д.П.** |
|  | подпись |  |
| **Студент** |  | **Махмудов Б.Н.** |
|  | подпись |  |
|  |  |

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc31124573)

[ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc31124574)

[1.1. Общий принцип работы систем веб-шаблонов 4](#_Toc31124575)

[1.2. Обзор существующих решений 4](#_Toc31124576)

[1.3. Основные проблемы производительности системы веб-шаблонов 4](#_Toc31124577)

[ГЛАВА 2. ИССЛЕДОВАНИЕ ВОЗМОЖНЫХ СПОСОБОВ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМ ВЕБ ШАБЛОНОВ 5](#_Toc31124578)

[2.1. Решение проблемы использования интерпретаторов 5](#_Toc31124579)

[2.2. Решение проблемы простоя системных ресурсов 5](#_Toc31124580)

[2.3. RESTful сервис, выполняющий роль системы веб шаблонов 5](#_Toc31124581)

[ГЛАВА 3. ПРОЕКТИРОВАНИЕ СЕРВИСА ПО ПРЕДОСТАВЛЕНИЮ УСЛУГ СИСТЕМЫ ВЕБ-ШАБЛОНОВ 6](#_Toc31124582)

[3.1. Моделирование работы сервиса 6](#_Toc31124583)

[3.2. Выбор инструментов 6](#_Toc31124584)

[3.3. Разработка сервиса 6](#_Toc31124585)

[3.4. Результат разработки, оценка производительности 6](#_Toc31124586)

[3.5. Потенциал к масштабированию 6](#_Toc31124587)

[ЗАКЛЮЧЕНИЕ 7](#_Toc31124588)

[СПИСОК ЛИТЕРАТУРЫ 8](#_Toc31124589)

[ПРИЛОЖЕНИЕ А. 9](#_Toc31124590)

[ПРИЛОЖЕНИЕ Б. 10](#_Toc31124591)

ВВЕДЕНИЕ

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## Термины и определения

Система – комбинация взаимодействующих элементов, организованных для достижения одной или нескольких поставленных целей [1].

Веб-шаблон – HTML-код с готовым дизайном и версткой, а также с дополнительной разметкой на языке шаблонизации, который используется для построения веб-документа.

Шаблонизатор – программное обеспечение, позволяющее генерировать конечные веб-документы с использованием веб-шаблонов и с учётом контекста (данных).

Система веб-шаблонов (СВШ) – система, состоящая из трёх элементов:

1. веб-шаблонов;
2. источника данных (JSON, XML, база данных);
3. шаблонизатора.

Контекст шаблонизации – структура данных, содержащая переменные окружения и методы, которые могут быть использованы шаблонизатором в процессе построения веб-документа.

Абстрактное синтаксическое дерево (АСД) – конечное помеченное ориентированное дерево, внутренние вершины которого поставлены в соответствие (помечены) с операторами языка программирования, а листья — с операндами [2].

## Общий принцип работы систем веб-шаблонов

Все существующие системы веб-шаблонов функционируют похожим образом. Для работы системы необходимо выполнение следующих условий:

* наличие шаблонизатора – основной компонент системы, который генерирует конечный документ;
* наличием веб-шаблонов, размеченных языком шаблонизации, синтаксис которого поддерживается используемым шаблонизатором;
* наличие источника данных, при этом структура данных должна соответствовать правилам формирования контекста, который может быть обработан шаблонизатором.

Процесс шаблонизации происходит последовательно в 4 этапа:

1. Шаблон загружается в оперативную память. Источником шаблона может быть поток байт, файл, запись в базе данных и т.п.
2. Специальная подпрограмма, предназначенная для лексического анализа, именуемая «лексер», проводит анализ шаблона, и разбивает его на лексические единицы – лексемы, которые также называют токенами. Результатом этого процесса является поток токенов, которые значительно проще обрабатывать.
3. Другая подпрограмма, целью которой является синтаксический разбор потока токенов, именуемая «парсер», производит преобразование потока токенов в особую древовидную структуру данных, известной как абстрактное синтаксическое дерево.
4. На основе абстрактного синтаксического дерева и переданного контекста шаблонизации, шаблонизатор производит построение конечного веб-документа.

Далее можно ознакомится с функциональной моделью вышеописанного процесса в нотации IDEF0. На Рис. 1.2.1 представлена контекстная диаграмма процесса шаблонизации, а на Рис. 1.2.2 приведена декомпозиция верхнего уровня данного процесса.



Рис. 1.2.1 Контекстная диаграмма процесса шаблонизации



Рис. 1.2.2 Декомпозиция верхнего уровня процесса шаблонизации

## Обзор существующих решений

На сегодняшний день существует большое множество систем веб-шаблонов, как проприетарных, так и с открытым исходным кодом.

Системы веб шаблонов можно выделить в две основные категории, в зависимости от способа их размещения относительно конечного пользователя:

* server-side – работающие на стороне сервера;
* client-side – работающие в составе клиентского приложения.

Первые системы веб шаблонов работали на стороне сервера, и представляли из себя препроцессоры текста. Распространённой практикой было использование технологии CGI (Common Gateway Interface), которая подразумевает вызов веб-сервером внешнего программного кода, как правило скрипта, задачей которого была генерация веб-документа с использование данных, полученных в запросе от клиентского приложения. Со временем появились полноценные веб-фреймворки, которые имеют в своём составе систему веб-шаблонов в качестве отдельного программного модуля. Далее приведен перечень, наиболее популярных систем веб-шаблонов, работающих на стороне сервера.

Blade – входит в состав веб-фреймворка Laravel. Написан на языке PHP. Также как и сам фреймворк, частью которого является, Blade обладает свободной лицензией и открытым исходным кодом.

Django – по сути сам является веб-фреймворком, написанном на языке Python, обладает широкими возможностями шаблонизации. На основе синтаксиса языка шаблонизации, используемого в Django было создано большое количество других СВШ. Свободная лицензия.

Mustache – это скорее спецификация по созданию систем веб-шаблонов, нежели отдельно взятый программный продукт, но существует не мало имплементаций данной спецификации на разных языках программирования. Синтаксис, определенный в спецификации, является одним из самых распространённых и широко-используемых. Открытая спецификация [3].

Twig – система веб-шаблонов, синтаксис которой основан на Django. Написан на языке PHP, наиболее часто используется при проектировании веб-приложений по паттерну MVC (Model View Controller). Свободная лицензия.

Lasso – является сервером приложений, имеющий в своем составе одноимённую подсистему веб-шаблонов. Является примером проприетарной СВШ. По возможностям значительно уступает свободным аналогам.

Также существуют системы веб-шаблонов, которые могут исполнятся прямо в клиентском приложении, как правило это веб-браузер.

## Основные причины низкой производительности шаблонизаторов

# ИССЛЕДОВАНИЕ ВОЗМОЖНЫХ СПОСОБОВ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМ ВЕБ ШАБЛОНОВ

## Устранение причины низкой производительности, связанной с использованием интерпретаторов

## Решение проблемы простоя системных ресурсов

## RESTful сервис, выполняющий роль системы веб шаблонов

# ПРОЕКТИРОВАНИЕ СЕРВИСА ПО ПРЕДОСТАВЛЕНИЮ УСЛУГ СИСТЕМЫ ВЕБ-ШАБЛОНОВ

## Моделирование работы сервиса

## Выбор инструментов

## Разработка сервиса

## Результат разработки, оценка производительности

## Потенциал к масштабированию

ЗАКЛЮЧЕНИЕ

СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ Р ИСО/МЭК 15288-2005. Информационная технология. Системная инженерия. Процессы жизненного цикла систем.
2. Михаил Вюрш. Улучшение распознавания изменений в исходных кодах с помощью абстрактных синтаксических деревьев: дипломная работа [Текст]. Университет Цюриха 2006. – 64 с.
3. Chris Wanstrath. Mustache – Logic-less templates [Электронный ресурс]: 2009. Режим доступа: <https://mustache.github.io/mustache.5.html>, свободный. (Дата обращения: 07.03.2020).

ПРИЛОЖЕНИЕ А.

**ДОКУМЕНТАЦИЯ API**

ПРИЛОЖЕНИЕ Б.

**СИНТАКСИС ДЛЯ НАПИСАНИЯ ШАБЛОНОВ**