**Федеральное государственное бюджетное образовательное**

**высшего образования**

**«Саратовский государственный технический университет**

**имени Гагарина Ю.А.»**

Институт прикладных информационных технологий и коммуникаций

Кафедра информационно-коммуникационных систем и программной инженерии

Направление 09.03.04 Программная инженерия

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ РОДИТЕЛЬСКОГО КОНТРОЛЯ «DIGITAL PARENT»**

Выполнил студент(ка)

Иванисенко Роман Максимович

курс 5

группа бПИНЖ41

Руководитель работы:

к.с.н., доцент кафедры ИКСП

Пчелинцева Елена Германовна

Допущен к защите

Протокол № \_\_\_\_\_\_\_\_\_\_ от \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2018 г.

Зав. кафедрой ИКСП д.т.н., профессор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А. Сытник

Саратов 2019 г.

«Саратовский государственный технический университет

имени Гагарина Ю.А.»

Институт прикладных информационных технологий и коммуникаций

Кафедра информационно-коммуникационных систем и программной инженерии

У Т В Е Р Ж Д А Ю

Директор

### « » 20 г.

#### З А Д А Н И Е

**на выпускную квалификационную работу магистра**

Студенту группы Иванисенко Роману Максимовичу

1. Тема проекта (работы) Система управления разработкой проектов

(утверждена распоряжением по ИКНТ от № )

1. Срок сдачи студентом оконченного проекта (работы)
2. Исходные данные к проекту (работе)

Техническое задание на разработку программного мо-

дуля

1. Содержание расчетно-пояснительной записки (перечень подлежащих разра- ботке вопросов)

|  |
| --- |
| а) Анализ предметной области; |
| б) Обзор существующих средств, для решения поставленной задачи; |
| в) Формулирование требований к системе; |
| г) Проектирование архитектуры системы; |
| д) Программная реализация; |
| е) Анализ результатов. |

1. Перечень графического материала (с точным указанием обязательных чертежей)

### \_

\_

\_

1. Консультанты по проекту (с указанием относящихся к ним разделов проекта, работы)
2. Дата выдачи задания Руководитель \_\_\_\_\_\_\_\_

Задание принял к исполнению

(дата)

### \_

(подпись студента)

Примечание: 1. Это задание прилагается к законченной ВКР и вместе с ВКР представляется в ГЭК.

2. Кроме задания, студент должен получить от руководителя календарный график работы над проектом (работой) на весь период проектирования (с указанием сро- ков исполнения и трудоемкости отдельных этапов)

#### РЕФЕРАТ

Данная работа содержит 35 страниц, 5 иллюстраций и 10 таблиц.

ВЕБ-ПРИЛОЖЕНИЕ, JAVA, SPRING BOOT, ИНТЕРНЕТ-СЕРВИС, УПРАВЛЕНИЕ ПРОЕКТАМИ, УПРАВЛЕНИЕ РАЗРАБОТКОЙ ПРОЕКТОВ

This task manager developed using Spring Boot, Spring Data, Hibernate, Bootstrap, Charts.js and JSP.

Работа описывает разработку веб-приложения Task Manager с трёхуровневой архитектурой. Серверная часть реализуется с помощью Spring Boot фреймворка, СУБД MySQL, клиентская часть реализуется на JSP с использованием фреймворка Charts.js для работы с графиками.

Главная цель данного интернет-сервиса – управление проектами и задачами в процессе разработки.

#### THE ABSTRACT

The work contains 35 pages, 5 illustrations and 10 tables.

WEB-APPLICATION, JAVASCRIPT, TYPESCRIPT, IT, THE INFORMATION PORTAL, MANAGEMENT OF THE PROJECTS, AR- CHITECTURE OF SYSTEM

The work is devoted to the development of a client-server applica- tion Task Manager. The server is implemented using the Spring framework, the client is developed on the JSP and Charts.js.

The main objective of this application is to manage projects and tasks of companies of various sizes and areas of activity.

# Содержание

[Список иллюстраций 7](#_bookmark0)

[Список таблиц 8](#_bookmark1)

[ВВЕДЕНИЕ 9](#_bookmark3)

[Актуальность работы 9](#_bookmark4)

[Цели и задачи 13](#_bookmark7)

[Краткое содержание работы 14](#_bookmark8)

[ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 15](#_bookmark9)

* 1. [Системы управления. Цели и задачи 15](#_bookmark10)
  2. [Сравнительный анализ 17](#_bookmark11)

[ГЛАВА 2. СБОР И АНАЛИЗ ТРЕБОВАНИЙ 21](#_bookmark14)

[2.1. Требования к системе. 21](#_bookmark15)

[ГЛАВА 3. АРХИТЕКТУРА И ОСОБЕННОСТИ](#_bookmark16) [РЕАЛИЗАЦИИ 22](#_bookmark16)

* 1. [База данных 23](#_bookmark18)
  2. [Сервер 24](#_bookmark19)
     1. [Сервис аутентификации и управления пользователями 24](#_bookmark20)
     2. [Сервис проектов 26](#_bookmark21)
  3. [Клиент 30](#_bookmark29)

[Заключение 31](#_bookmark33)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 32](#_bookmark34)

[Приложение 1 33](#_bookmark35)

# Список иллюстраций

[Рис. 1 Распространённость системы управления в России ... 11](#_bookmark5)

[Рис. 2 Распределение по отраслям 12](#_bookmark6)

[Рис. 3 Схема трехзвенной архитектуры «Сервер БД - Сервер](#_bookmark17) [приложений - Клиент» 22](#_bookmark17)

[Рис. 4 Модульная структура Spring Framework ... 36](#_bookmark31)

[Рис. 5 Свойства для MySQL и Hibernate ... 36](#_bookmark31)

# Список таблиц

[Таблица 1. Сравнение Task Manager и конкурентов, ч. 1 19](#_bookmark12)

[Таблица 2. Сравнение Task Manager и конкурентов, ч. 2 20](#_bookmark13)

[Таблица 3. Получение списка проектов. 27](#_bookmark22)

[Таблица 4. Получение списка спринтов проекта. 27](#_bookmark23)

[Таблица 5. Получение списка задач в спринте. 28](#_bookmark24)

[Таблица 6. Получение информации по задаче. 28](#_bookmark25)

[Таблица 7. Изменение информации задачи 29](#_bookmark26)

[Таблица 8. Создание нового проекта 30](#_bookmark27)

[Таблица 9. Создание нового спринта 30](#_bookmark28)

[Таблица 10. Создание новой задачи 31](#_bookmark30)

# ВВЕДЕНИЕ

# Актуальность работы

Современное управление проектом является одним из способов управления, который может иметь отношение к управлению любыми объектами вне зависимости от их характеристик. В настоящее время в России проектное управление имеет стремительное развитие из-за применения современных подходов в сфере управления проектами и внедрения современных автоматизированных систем управления проектами. Применение систем управления проектами повышает качество проектов, минимизирует расходы и сроки. Многие компании внедряют различные компоненты систем управления проектами, которые закладывают основу для дальнейшего развития компании.

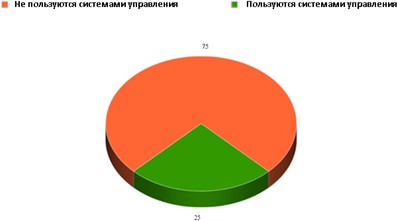
Для более глубокого понимания проблематики и выявления актуальности темы мною было осуществлено анкетирование людей, которые работают более года в организациях, использующие различные системы управления проектами. К таким организациям можно отнести IT компании, государственные и коммерческие учреждения, а также производственные фирмы и т.д.

В анкете предлагался следующий перечень вопросов:

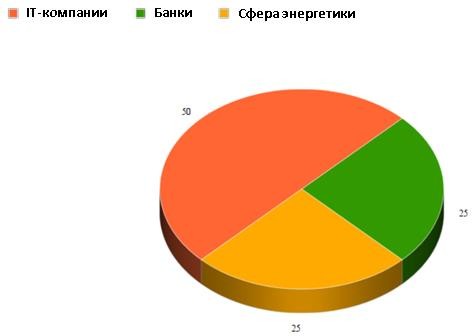
* + - Используете ли вы в процессе работы системы управления?
    - Имеются ли подобные системы на рабочих местах?
    - Насколько полезны подобные системы в работе? и другие.

Исходя из результатов анкетирования (рис. 1 и 2) было выявлено, что многие опрошенные работают в компаниях, где списки задач излагаются устно или в письменной форме. Но имеются компании, в которых уже большое количество времени внедрена система управления. Работники данных организация на вопрос “Насколько полезны подобные системы в работе ”, ответили, что: “Позволяет систематизировать рабочие задания, помогает оценить сроки выполнения, придаёт большую дисциплину в целях минимизации риска просрочки. Позволяет анализировать выполненные задачи, определяет скорость выполнения каждого работника”.

Более подробно хотелось бы рассмотреть работников IT компаний. В данных компаниях структурированность выполняемых задач и контроль выполнения является весьма актуальным. Эти компании применяют подобные системы довольно давно и очень активно. Среди прочих систем для управления проектами очень популярной является - Jira, она обладает гибкостью и в большинстве случаев функционал для бизнес-модели проектов данных организаций подходит для данной системы.



**Рис. 1 Использование систем управления проектами в России**



**Рис. 2 Разделение по отраслям**

На основе полученных данных, было выявлено, что система управления позволяет осуществлять рабочий процесс более быстро и качественно. Применение таких систем положительно влияет на управлении проектами и в некоторых случаях на управлении компанией. В настоящее время системы управления разработкой проектов являются весьма актуальными и полезными.

# Цели и задачи

Веб-приложение Task Manager представляет собой автоматизированное решение для управления разработкой проектов.

Данная система позволяет создавать задачи для проекта, редактировать, удалять, искать по ключевым словам, дате, отображать статистику выполнения задач в виде графиков, создавать проект и добавлять в него задачи, распределять их по пользователям.

Цель создания данного программного решения – упростить управление разработкой проектов.

# Краткое содержание работы

В главе 1 проводится анализ предметной области, описываются цели и задачи систем управления разработкой проектов, сравниваются разные системы.

В главе 2 описаны требования, предъявляемые к системам управления разработкой проектов.

В главе 3 рассматриваются архитектура и особенности системы управления разработкой проектов Task Manager.

В заключении производится обзор проделанной работы и под- водится итог.

# ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

# Системы управления. Цели и задачи.

Система управления - система для накопления знаний о проекте или некотором объекте, которая содержит в себе средства для воздействия. Целями внедрения систем управления проектами являются:

* + - Увеличение производительности работников организации в процессе работы над проектами;
    - Повышение качества управления проектами руководителями проектов;
    - Увеличение эффективности управления группой проектов компании – больше проектов в срок и в рамках бюджета с малыми расходами.

Чтобы обеспечить достижение поставленных целей, необходимо внедрять системы управления разработкой проектов, которые в свою очередь нужны для того, чтобы:

* + - Обеспечить руководителя проекта средствами планирования проекта и контроля реализации;
    - Предоставить участнику проекта средства для планирования задач проекта и доступа к нужной информации;
    - Руководителю отдела предоставить средства управления распределением работников по проектам, предоставить сведения для принятия решения о назначении работников на вновь созданные проекты;
* Руководителю компании дать возможность организовать мониторинг всех проектов компании.

Функциональные требования информационных систем управления разработкой проектов определяются из характеристик процессов управления проектами в определённой компании.

В зависимости от специфики имеются различные области их применения:

* + - * Системы управления строительными проектами. Ра- боту над проектом осуществляют разные специалисты и компании, которые должны совместно работать так, чтобы проект был выполнен в срок с высоким качеством и с запланированным бюджетом.
      * Системы управления инновационными проектами. В данных проектах необходимо организовать взаимодействие работников и мотивировать их в процессах создания новых идей.

Управление организационными проектами. «Организационные изменения успешны только тогда, когда эмоциональным и поведенческим факторам уделяется внимание не менее, чем производственным» - Джини Даниэль Дак [2]

Преимущества от системы управления проектами по отношению к бизнесу:

* + - Повышение эффективности использования ресурсов посредством улучшения распределения ресурсов между проектами, более тщательного управления загрузкой работников.
    - Уменьшение перерасходов бюджета. Организуется за счёт улучшения планирования и увеличения контроля за расходами.
    - Уменьшение процента не выполненных проектов. Осуществляться за счёт применения средств мониторинга проектов, внедрения методологии через информационную систему, проектный офис уменьшает процент проектов, которые не могут быть выполнены.

# Сравнительный анализ

Системы управления разработкой проектов бывают следующих видов:

* + Локальные/приложения с оконным интерфейсом (например, Microsoft Project);
  + Клиент-серверные, (например, Microsoft Project Server);
  + Веб-базированные – для использования данных приложений необходим только интернет-браузер.

Современные информационные системы управления проектами в большинстве своём являются веб-базированными интернет-приложениями. По-моему мнению, наиболее оптимальное решение – остановиться на выборе онлайн системы управления проектами с возможностью хранения всех данных на серверном оборудовании.

На сегодняшний день наиболее популярными Веб-базированными системами являются: Jira, Taiga, Trello, Redmine.

Jira – является коммерческой системой. Основная цель приложения - это организация взаимодействия с пользователями и контроль за появлением ошибок. Но в большинстве случаев, она часто применяется для управления проектами.

Taiga - это платформа управления проектами для разработки с применением гибкой методологии разработки Agile. Целевой аудиторией Taiga являются разработчики программного обеспечения, дизайнеры и менеджеры проектов.

Trello - это система управления проектами, которая применяется для малых команд. Она позволяет использовать методологию управления проектами, именуемую как Kanban.

Redmine – является открытым серверным веб-приложением для управления проектами и задачами.

Сравним рассмотренные системы управления проектами с приложением Task Manager:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Task Manager | Jira | Taiga |
| Свободное распространение | + | - | + |
| Наличие ServiceDesk | - | + | - |
| Простота | + | - | + |
| Анализ производительности | + | - | + |
| Удобство интерфейса | + | - | + |
| Наличие поиска по задачам | + | + | - |

**Таблица 1. Сравнение Task Manager и программ-аналогов, ч. 1**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Task Manager | Trello | Redmine |
| Свободное распространение | + | + | +/- |
| Наличие ServiceDesk | - | - | + |
| Простота | + | + | - |
| Анализ производительности | + | - | + |
| Удобство интерфейса | + | + | - |
| Наличие поиска по задачам | + | + | - |

**Таблица 2. Сравнение Task Manager и программ-аналогов, ч. 2**

# ГЛАВА 2. СБОР И АНАЛИЗ ТРЕБОВАНИЙ

В данной главе рассматриваются требования для системы управления разработкой проектов Task Manager.

# Требования к системе.

* + - Система должна быть реализована в виде веб-приложения, имеющего трехзвенную архитектуру (клиент - сервер – сервер базы данных).
    - Система должна быть многопользовательской.
    - Система должна иметь простой и удобный интерфейс
    - В системе должны учитываться права доступа пользователей.
    - Данные пользователей должны шифроваться с использованием Base64 и секретного ключа.
    - Система должна уметь:
      * Создавать новые проекты;
      * Создавать задачи в данных проектах;
      * Добавлять в проекты спринты;
      * Снабжать спринты задачами.
    - Система должна уметь показывать график выполнения задач.
    - Система должна показывать задачи в спринте в виде доски со стикерами.

# ГЛАВА 3. АРХИТЕКТУРА И ОСОБЕННОСТИ РЕАЛИЗАЦИИ

Веб-приложение Task Manager построено с использованием трехзвенной архитектуры «Сервер БД - Сервер приложений - Клиент» (архитектура представлена на рис. 3).



**Рис. 3 Представление трехзвенной архитектуры «Сервер БД - Сервер приложений - Клиент»**

### 22

**Клиент** – основной функцией является взаимодействие с пользователем. Клиент это, то, что пользователь наблюдает на странице в браузере, то есть является веб-клиентом. Графический интерфейс отображается в браузере. Именно посредством браузера пользователь взаимодействует с веб-приложением. Клиентская часть формирует запросы к серверу и получает ответы от него.

**Сервер** отвечает за организацию логики веб-приложения. Логика веб-приложения распределена между сервером и клиентом, обмен информацией происходит по сети.

**База данных** отвечает за хранение данных и их выдачу по запросу. Серверная часть веб-приложения обращается к базе данных с помощью запросов и получает необходимые данные.

Рассмотрим каждое звено архитектуры более подробно.

# База данных

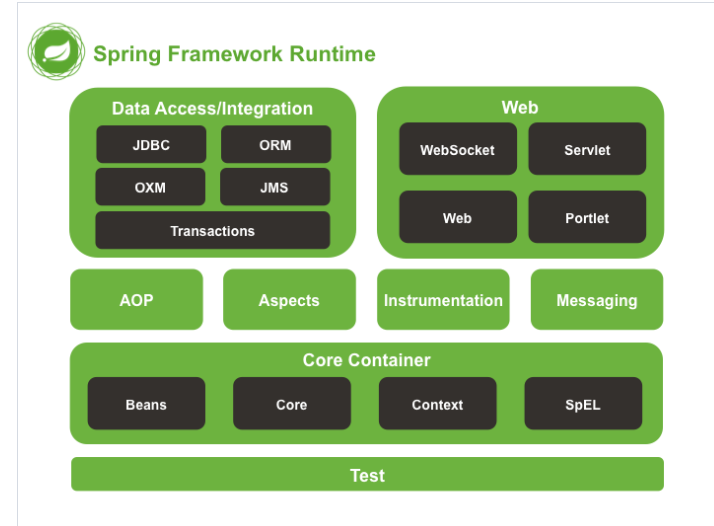
В качестве СУБД было принято решение использовать СУБД MySQL. Данная база данных является реляционной. Она была выбрана в первую очередь, из-за того, что в ней реализована многопоточность, что позволяет обеспечивать высокую скорость работы. Также немаловажным является то, что она гарантирует безопасность и основной сферой применения рассматриваемой СУБД является – веб. Поддерживаются сложные запросы и аналитика, а также легко использовать данные из разных приложений.

# Сервер

Для серверной части был выбран язык программирования Java, так как он является кроссплатформенным, поддерживает парадигму ООП, используется большинством компаний для разработки веб-проектов и имеет соответствующие для этого инструменты. Одним из основных преимуществ языка Java является возможность переноса программ из одной системы в другую. Поскольку программы на Java не зависят от платформы как на уровне исходного кода, так и на двоичном уровне, их можно запускать в различных системах, что особенно важно для программ, предназначенных для World Wide Web.

Большое количество возможностей Java, простота применения, независимость от платформы и встроенные функции защиты делают данный язык программирования одним из лучших для создания приложений для Internet. Для создания веб-приложения наилчшим решением было выбрать фреймворк для постороения веб-проектов Spring, так как он Один из самых популярных фреймворков для создания веб-приложений на Java. **Фреймворк** — это что-то похожее на библиотеку, но есть один момент. Используя библиотеку, происходит создание объектов классов, которые в ней есть, вызывая нужные методы, и таким образом получается нужный результат. То есть, тут более императивный подход: здесь более точно необходимо указывать в своей программе в какой конкретный момент надо создать какой объект, в какой момент вызвать конкретный метод, и т.д. С фреймворками дела обстоят слегка иначе. Необходимо найти определенные классы, настроить там часть логики, а создает объекты классов и вызывает методы за вас уже сам фреймворк. Чаще всего, ваши классы имплементируют какие-то интерфейсы из фреймворка или наследуют какие-то классы из него, таким образом получая часть уже написанной за вас функциональности. Но не обязательно именно так. В спринге например стараются по максимуму отойти от такой жесткой связности (когда классы напрямую зависят от каких-то классов/интерфейсов из этого фреймворка), и используют для этой цели аннотации. С**принг** — это набор классов и интерфейсов, которые уже написаны заранее разработчиками.

Но спринг — это не один конкретный фреймворк. Это общее названия для целого ряда небольших фреймворков, каждый из которых выполняет какую-то свою работу.



**Рис. 4 Модульная структура Spring Framework**

Как видно, у спринга модульная структура. Это позволяет подключать только те модули, которые необходимы для приложения и не подключать те, которые заведомо не будут использоваться. Именно этот подход и помог спрингу обойти своего конкурента в то время (EJB) и захватить лидерство. Потому что приложения, использующие ***EJB***  имели очень много зависимостей за собой, да и вообще получались медленные и неповоротливые. На изображении видно, что спринг фреймворк состоит из нескольких модулей***:***

* data access;
* web;
* core;
* и других.

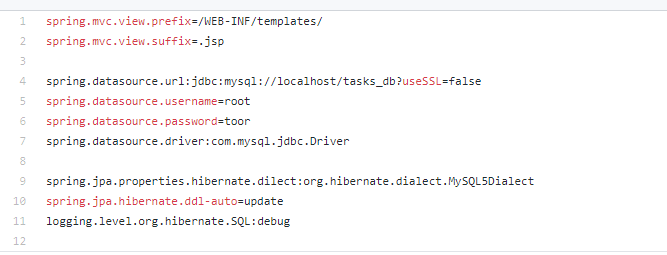
Модуль *data access* содержит в себе средства для работы с данными (в основном, с базами данных), *web* — для работы в сети. Кроме того, есть еще так-называемая целая спринг-инфраструктура: множество других проектов, которые не входят в сам фреймворк официально, но при этом интегрируются в проект на спринге (например, тот же *spring security* для работы с авторизацией пользователей на сайте). Существует **три основных способа конфигурации приложения** (то-есть, указания спрингу какие именно объекты нам нужны для работы):

* при помощи xml файлов/конфигов;
* при помощи java-конфигов;
* автоматическая конфигурация.

**Разработчики спринга выстраивают их в таком порядке приоритетности:**

* наиболее приоритетный способ — это автоматическая конфигурация;
* если при помощи автоматической конфигурации нет возможности правильно настроить все возможные бины — использовать джава-конфигурацию (создание объектов используя джава код);
* ну и самый низкоприоритетный способ — используя xml конфиги.

Кроме того, спринг позволяет комбинировать эти способы. Например, все то, что может быть настроено автоматически — пусть спринг сделает сам, там где надо указать какие-то особые параметры — сделать при помощи джава-конфигов, и кроме того, можно подключить конфиги в xml формате. В общем, достаточно гибко это все можно сделать. Но все же, если все можно сделать при помощи автоматической настройки — необходимо её использовать. Я буду рассматривать только автоматическую настройку и джава-конфиги; xml конфиги. Для настройки работы с MySQL и Hibernate необходимо создать файл с настройками xml ( на рис. )



**Рис. 5 Файл для настроек MySQL и Hibernate**

# Клиент

Клиент предоставляет интерфейс для отображения и взаимодействия с данными, предоставляемыми сервисами. Реализован с помощью JSP и фреймворка Charts.js. JSP является технологией, которая даёт возможность веб-разработчикам создавать содержимое, которое имеет как статические, так и динамические компоненты. Страница JSP содержит текст двух типов: статические исходные данные, которые могут быть оформлены в одном из текстовых форматов [HTML](https://ru.wikipedia.org/wiki/HTML), [SVG](https://ru.wikipedia.org/wiki/SVG), [WML](https://ru.wikipedia.org/wiki/WML), или [XML](https://ru.wikipedia.org/wiki/XML), и JSP- элементы, которые конструируют динамическое содержимое. Кроме этого могут использоваться библиотеки JSP-тегов, а также [EL](https://ru.wikipedia.org/wiki/Expression_Language) ([Expression Language](https://ru.wikipedia.org/wiki/Expression_Language" \o "Expression Language)), для внедрения Java-кода в статичное содержимое JSP-страниц.

Код JSP-страницы транслируется в Java-код сервлета с помощью компилятора JSP-страниц [Jasper](https://ru.wikipedia.org/wiki/Apache_Tomcat#Jasper), и затем компилируется в [байт-код](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D1%82-%D0%BA%D0%BE%D0%B4) виртуальной машины [java](https://ru.wikipedia.org/wiki/Java) ([JVM](https://ru.wikipedia.org/wiki/Java_Virtual_Machine)). [Контейнеры сервлетов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%82%D0%B5%D0%B9%D0%BD%D0%B5%D1%80_%D1%81%D0%B5%D1%80%D0%B2%D0%BB%D0%B5%D1%82%D0%BE%D0%B2), способные исполнять JSP-страницы, написаны на платформонезависимом языке Java. JSP-страницы загружаются на сервере и управляются из структуры специального Java server packet, который называется [Java EE](https://ru.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition)Web Application. Обычно страницы упакованы в файловые архивы [.war](https://ru.wikipedia.org/wiki/WAR_(%D1%82%D0%B8%D0%BF_%D1%84%D0%B0%D0%B9%D0%BB%D0%B0)) и [.ear](https://ru.wikipedia.org/wiki/EAR).

JSP является платформонезависимой, переносимой и легко расширяемой технологией для разработки веб-приложений. Charts.js – JavaScript библиотека для визуализации данных.

Скриншоты приложения можно посмотреть в приложении 1.

# Заключение

В ходе проделанной работы были достигнуты следующие ре- зультаты:

* Разработано веб-приложения для управления разработкой про- ектов, включающее в себя:
  + Серверная часть на Java с использованием Spring Framework
  + База данных MySQL
  + Клиентская часть на JSP и Charts.js
* Получен практический опыт создания с нуля frontend части веб- приложения на JSP и Charts.js
* Получен практический опыт создания с нуля backend части веб- приложения на Spring Framework
* Получены знания в программировании на языке Java.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Чурилов А. А. Понятие, роль и актуальность проектного управ- ления в России // Молодой ученый. — 2013. — №3. — С. 299-

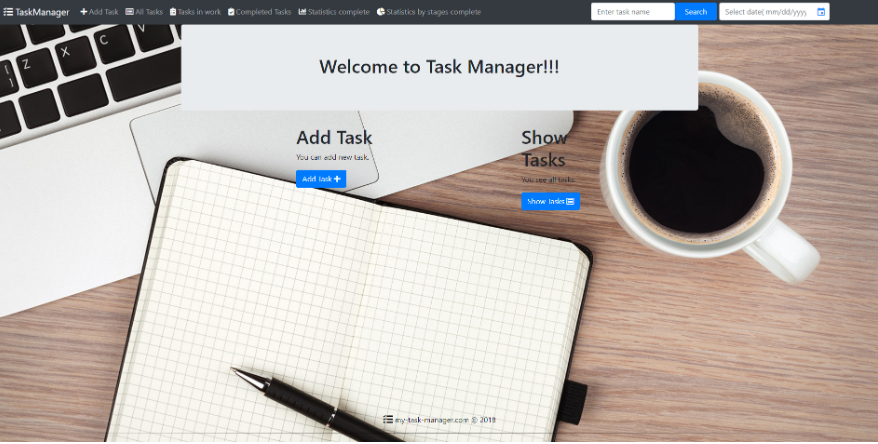
301. — URL https://moluch.ru/archive/50/6423/ (дата обращения: 08.05.2018).

1. Дак Дж. Д. Монстр перемен. Причины успеха и провала орга- низационных преобразований. – М.: Альпина Бизнес Букс, 2007. – 320 с.
2. Документация Material // URL: [https://v5.material.angu- lar.io/components/dialog/overview](https://v5.material.angular.io/components/dialog/overview) (Дата обращения 01.04.2018)
3. Херман Дэвид Сила JavaScript. 68 способов эффективного ис- пользования JS; Питер – М., 2015. 952 с.
4. Творчество. Свобода. Жизнь. – 2018 // URL: https://[www.adme.ru/zhizn-nauka/](http://www.adme.ru/zhizn-nauka/)

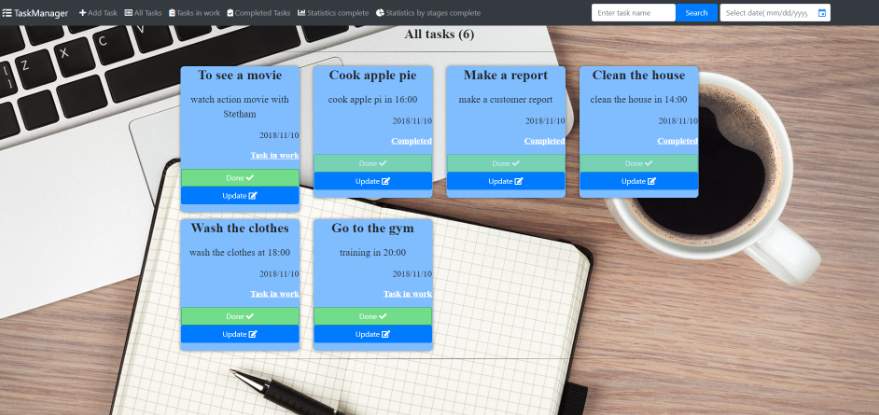
# Приложение 1

### Скриншоты Task Manager

### Главная страница



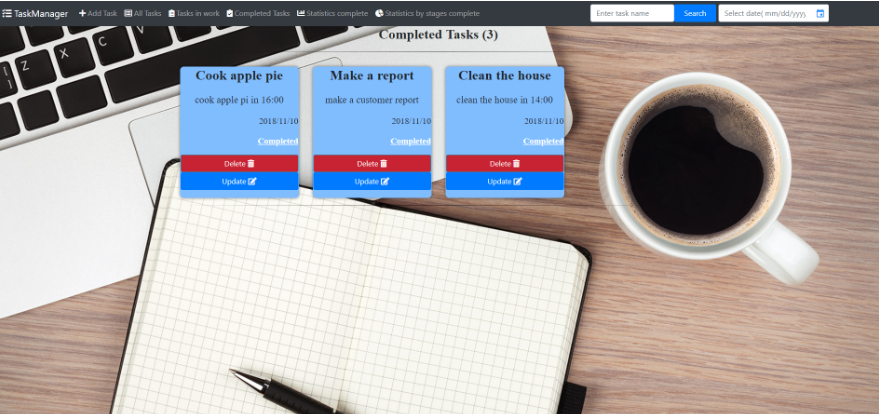
Страница со всеми задачами



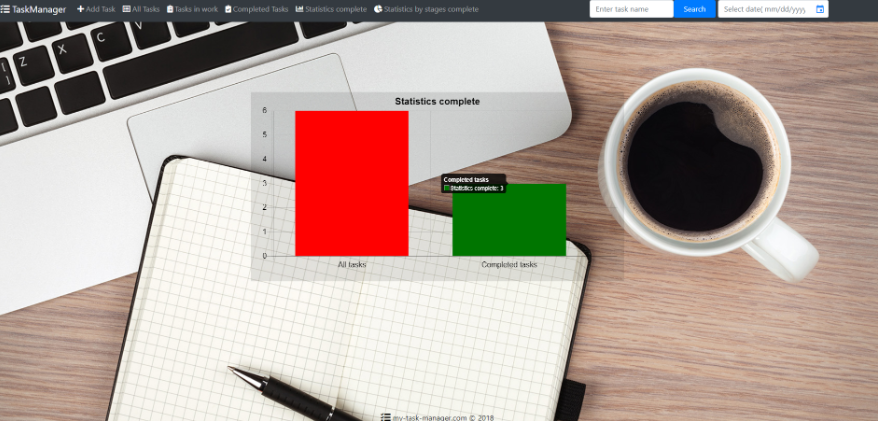
Страница с задачами, которые находятся “В работе”



Страница с выполненными задачами



Страница с отображением отношения всех задач к выполненным



Статистика выполнения задач

