

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра теории функции и стохастического анализа

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ УЧЕТА ЗАДАЧ
ЛАБОРАТОРИИ

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 248 группы
направления 09.04.03 Прикладная информатика

механико-математического факультета
Беяева Александра Ильича

Научный руководитель
к. ф.-м. н. доцент

Р. Н. Фадеев

Заведующий кафедрой
д. ф.-м. н., доцент

С. П. Сидоров

Саратов 2020

Введение. У человечества за всю историю накопился внушительный список успешно реализованных сложных проектов. От строительства Пирамид в Гизе до отправки человека на Луну, самые смелые человеческие начинания требовали слаженной работы тысяч людей. А это подразумевает сложную систему управления проектами.

Говоря простыми словами, управление проектами – это управление и организация всего, что нужно для достижения цели – вовремя и в рамках бюджета. Будь то разработка нового программного обеспечения, проведение маркетинговой компании или высадка человека на Марс – проектное управление позволяет добиться успеха.

Управление проектом строится на совокупности методов и систем поддержки, управления, разработки, обслуживания и т.п. Одна из фундаментальных систем для управления проектом, является система учета задач или Task Manager.

Целями данной практики является ознакомление с методами управления проектами, готовыми решениями, а так же применение полученных знаний на практике, в виде реализации собственного программного обеспечения.

- Изучение моделей системы учета и готовых решений.
- Ознакомление с особенностями работы языка программирования Java.
- Средствами языка Java реализовать программное обеспечение учета задач лаборатории.

Работа прошла апробацию на различных конференциях, в частности, на ежегодной студенческой конференции "Актуальные проблемы математики и механики которую проводил механико-математический факультет СГУ в апреле 2019 года, в секции "Анализ данных в VIII Международной молодежной научно-практической конференции «Математическое и компьютерное моделирование в экономике, страховании и управлении рисками», ноябрь 2019 года.

Системы управления задачами. Руководство задачами автоматизируется через программные системы, или по-английски «Task Management Tool». Они предназначены для автоматизации процессов, сложных для расчета “руками” или процессов постоянных, часто повторяемых. Такие системы упрощают делегирование задач, систематизируют их, автоматически созда-

ют отчетность за определенный период времени с установленной частотой (раз в неделю, месяц, квартал).

В основе управления проектами лежит планирование – краткосрочное или на более длительный период. В бизнес-процессах планирование основывается на определённых методиках, в зависимости от приоритета задач и сроков их выполнения.

Все проекты разные. Не существует идеальной системы управления проектами, подходящей для каждого из видов проектов. Также не существует системы, которая бы подходила каждому руководителю и была удобна для всех членов команды. Однако за время существования проектного управления было создано немало эффективных подходов, методик и стандартов, которые можно взять на вооружение. Разработанные подходы сильно отличаются друг от друга. Они различаются по областям применения, детализированности, самодостаточности и формализации.

Суть процесса контроля над проектами и задачами сводится к надобности:

- Планировать задачи и проекты на длительное время.
- Делегировать поставленные задачи.
- Следить за процессом исполнения.
- Контролировать продолжительность исполнения задач.
- Обозначать четкие временные рамки.
- Анализировать продуктивность сотрудников.

Рассмотри некоторые из уже готовых решений.

Jira – это коммерческая система отслеживания ошибок, предназначена для организации взаимодействия с пользователями, хотя в некоторых случаях используется и для управления проектами. Мощный онлайн сервис, позволяющий командам-разработчикам планировать проекты, назначать исполнителей задач, планировать спринты и собирать задачи в бэклог, выставлять приоритеты и дедлайны. Система позволяет работать с несколькими проектами. Для каждого из проектов создаёт и ведёт схемы безопасности и схемы оповещения.

Jira имеет большое количество возможностей конфигурации: для каждого приложения может быть определён отдельный тип задачи с собственным workflow, набором статусов, одним или несколькими видами представления.

Кроме того, с помощью так называемых «схем» можно определить для каждого индивидуального Jira-проекта собственные права доступа, поведение и видимость полей и многое другое.

К плюсам данного решения относятся:

- Drag and Drop для перетаскивания задач в бэклоге, в спринте.
- Большое количество настроек фильтрации.
- Визуализация — позволяет быстро ориентироваться и делать изменения в спринте.
- Возможность выбора причин для закрытия задач.
- Детальные отчеты.
- Простота в управлении проектами.
- Простой и удобный трэкер времени.

Redmine - это открытое серверное веб-приложение для управления проектами и задачами (в том числе для отслеживания ошибок). Redmine написан на Ruby и представляет собой приложение на основе широко известного веб-фреймворка Ruby on Rails. Распространяется согласно GNU General Public License.

Некоторые возможности:

- Гибкая система доступа, основанная на ролях.
- Система отслеживания ошибок и создания записей об ошибках на основе полученных писем.
- Ведение новостей проекта, документов и управление файлами, а также создание форумов и вики-страниц проектов.
- Оповещение об изменениях с помощью RSS-потоков и электронной почты
- Настраиваемые произвольные поля для инцидентов, временных затрат, проектов и пользователей.
- Лёгкая интеграция с системами управления версиями (SVN, CVS, Git, Mercurial, Bazaar и Darcs).
- Поддержка СУБД MySQL, PostgreSQL, SQLite, Oracle.

Redbooth - это Web-сервис, доступный также на iOS и Android, позволяющий отслеживать ошибки, планировать проекты и активности, управлять ресурсами и ставить задачи участникам проекта, устанавливая дедлайны и

вехи проекта, следить за расходами проекта. Также сервис предлагает делиться файлами и синхронизацию с Google Drive, Gmail и Outlook.

Плюсы за:

- Простой и интуитивный интерфейс.
- Возможности чата.
- Интеграция с другими сервисами.
- Тэги для задач.
- Возможность создания отчетов.

Реализация системы учета задач лаборатории. Java — строго типизированный, объектно-ориентированный язык программирования, разрабатываемый компанией Sun Microsystems с 1991 года и официально выпущенный 23 мая 1995 года. Язык разрабатывался для бытовой электроники, но впоследствии стали использоваться для написания апплетов, приложений и серверного программного обеспечения.

Программы на Java транслируются в байт-код, выполняемый на виртуальной машине Java (Java Virtual Machine) - программе, обрабатывающей байт-код и передающей инструкции оборудованию, как интерпретатор, но с тем отличием, что байт-код обрабатывается значительно быстрее, нежели текст. Также выполнение программ из байт-кода является полностью независимым от операционной системы и оборудования.

Java как правило, более высокого уровня, чем аналогичные языки (такие как C++), что означает, что языку Java не хватает функций, таких как аппаратные или для работы с конкретными типами данных или, например, низкоуровневыми указателями на произвольные ячейки памяти. Хотя этими функциями часто злоупотребляют или используются программистами неправильно, они также являются мощными инструментами.

Плюсами данного языка являются:

- Объектно-ориентированный. В Java все является объектом;
- Платформонезависимый. В отличие от многих других языков, включая C и C++, Java, когда был создан, не компилировался на платформе конкретной машины, а в независимом от платформы байт-коде. Этот байт код распространяется через интернет и интерпретируется в Java Virtual Machine (JVM), на которой он в настоящее время работает;

- Высокая безопасность. Поскольку все приложения выполняются и, соответственно, контролируются Java-машиной, то при любой попытке несанкционированного доступа или соединения программа просто прерывает свою работу;
- Автоматическое управление памятью. Программы очистки памяти ищут объекты которые больше не используются и удаляют их. Это влияет на работу процессора, однако умная оптимизация и настройка позволяют снизить это влияние;
- Стабильность. Уже много лет развитию Java способствуют сообщество, поддержка Oracle и изобилие приложений и языков на JVM. Кроме того, постоянно выпускаются новые версии Java с новыми интересными функциями;
- Богатая библиотека. Богатая библиотека сводит тривиальную часть работы программиста к поиску готового решения. Что освобождает программиста от рутины и предоставляет ему возможность более серьезно вложиться в творческий аспект, сэкономить время и силы;
- Многопоточность. Можно писать программы, которые могут выполнять множество задач одновременно. Введение в язык Java этой конструктивной особенности позволяет разработчикам создавать отлаженные интерактивные приложения;
- Гибкость. Java доказала, что C — процедурный, управляемый вручную и зависящий от платформы код — это не предел совершенства. Благодаря Java, все больше людей начали применять объектно-ориентированное программирование, которое сейчас используется повсеместно.
- Апплеты. Еще до появления JavaScript, в Java добавили апплеты — небольшие веб-программы, которые предоставляют интерактивные элементы для визуализации и обучения. Они не используются ни для чего, кроме простой анимации, однако апплеты привлекли внимание многих программистов и подтолкнули их к разработке HTML5, Flash и JavaScript.

Описание реализации программы.

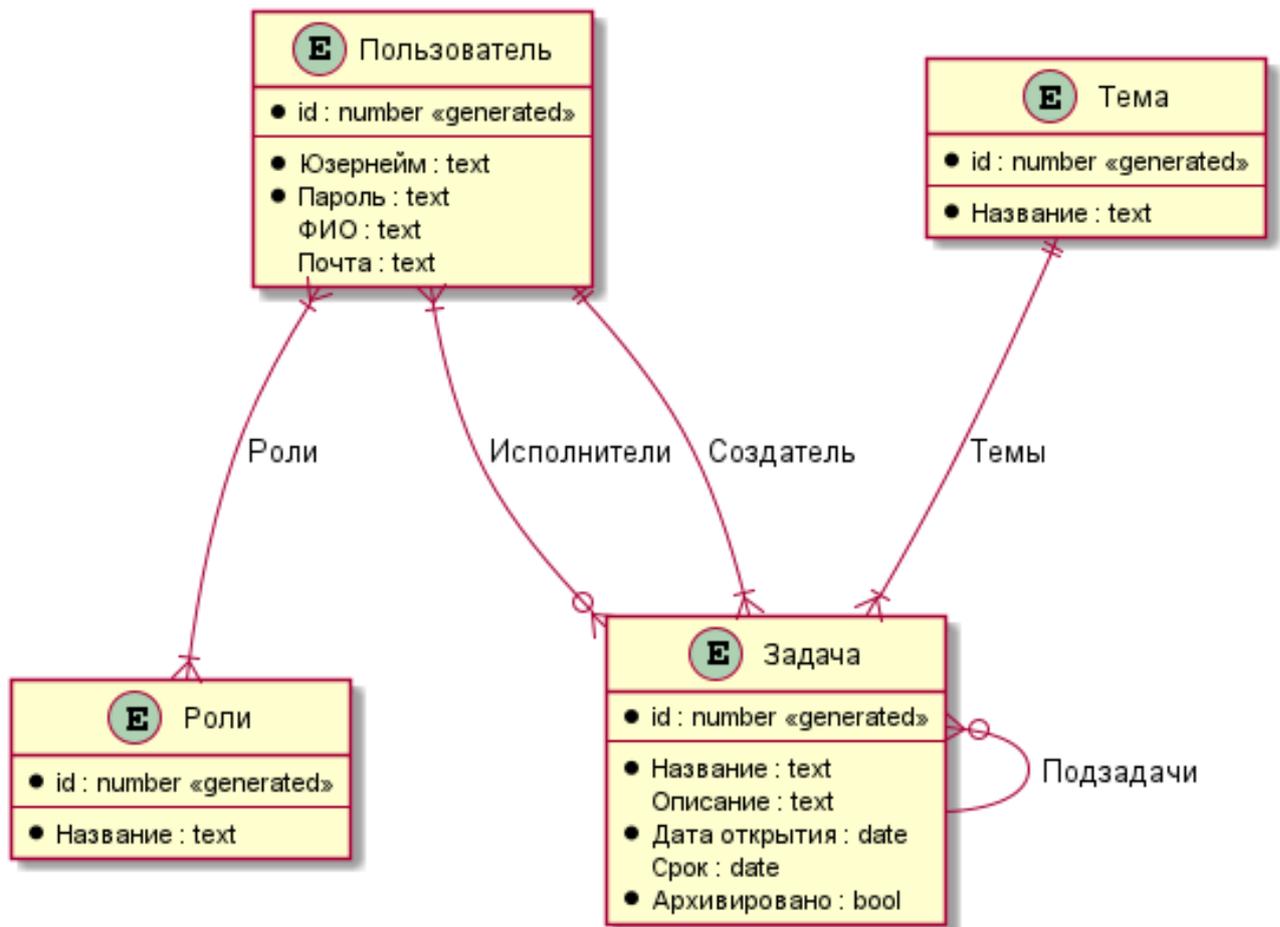


Рисунок 1 — Схема структуры базы данных

На рисунке 1 изображена структура базы данных. Пользователи и роли связаны отношением many-to-many (у пользователя может быть много ролей и наоборот), задачи и темы one-to-many (у одной темы может быть много задач). Пользователи и задачи имеют две связи:

- 1 Для исполнителей many-to-many (у задачи может быть много исполнителей, и у пользователя может быть много задач для выполнения),
- 2 Для создателя one-to-many (у задачи может быть один создатель).

Также есть внутренняя связь таблицы задач для подзадач.

Скриншоты

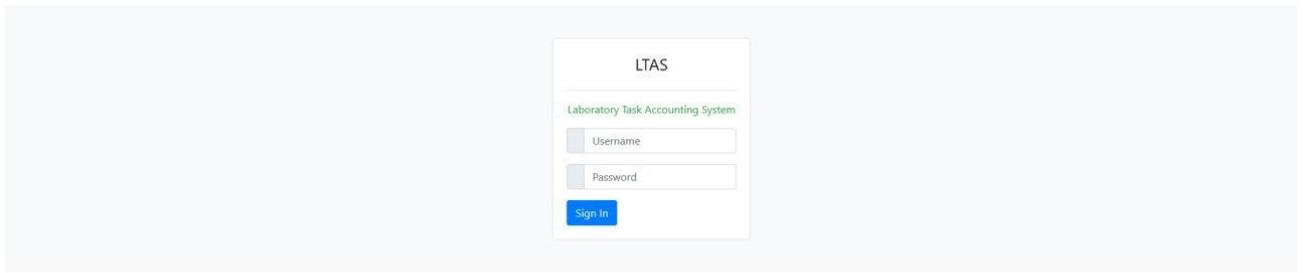


Рисунок 2 — Страница входа



Рисунок 3 — Таблица задач

LTAS [Задачи](#) [Пользователи](#) [Темы](#) [Роли](#) [Выход](#)

Новая задача

Название

0/255

Тема

Описание

0/255

Исполнители

Дата начала Срок выполнения

Если не установлена, будет использована сегодня

[Добавить](#)

Рисунок 4 — Форма добавления задачи

LTAS [Задачи](#) [Пользователи](#) [Темы](#) [Роли](#) [Выход](#)

id	Юзернейм	ФИО	Почта	Роли
1	admin	No fullname	No email	ADMIN

[Previous](#) [1](#) [Next](#)

Рисунок 5 — Таблица пользователей

LTAS Задачи Пользователи Темы Роли Поиск Выход

Новый пользователь

Имя пользователя

Email

Пароль Подтверждение пароля

ФИО

Добавить

Рисунок 6 — Форма добавления пользователя

LTAS Задачи Пользователи Темы Роли Поиск Выход

id	1: Название темы	И
1	Без темы	

Previous 1 Next

Рисунок 7 — Таблица тем для задач

id	1: Название темы
1	Без темы

Рисунок 8 — Таблица пользовательских ролей

На рисунке 2 изображена страница входа в систему, данные пользователя хранятся в базе данных, их пароли хранятся в хешированном виде.

На рисунках 3, 5, 7, 8 изображены таблицы с выводом задач, пользователей, тем и ролей соответственно. У таблицы задач есть кнопка Print, выводящая их на печать в виде списка. Данные поля можно сортировать по любому из представленных столбцов, как например название задачи, ее ID в базе данных, или времени начала или конца. Так же реализован функционал поиска по таблицы в которой находится пользователь в данный момент.

На рисунках 4, 6 показаны формы добавления новых задач и пользователей. Форма добавления исполнителей, даты начала и срока завершения, выбор темы из созданных ранее, описание задачи и название самой задачи. На форме добавления пользователя реализованы возможность задать полного ФИО, почты, имени пользователя, а так же задание пароля. Данная форма выступает в качестве закрытой регистрации внутри приложения.

Для лучшей масштабируемости была использована одна из сильных сторон механизма шаблонов Apache FreeMarker под названием макросы. Например, все страницы, включая страницу входа, используют макрос page, в котором подключены библиотеки, необходимые для их корректной работы. Страницы самой программы используют макрос base, в котором определена навигация.

гационная панель. Для элементов данной панели используется макрос `navto`, за счет которого сильно уменьшается количество шаблонной `html`-разметки. В зависимости от типа страницы (вывести все элементы или добавить элемент) за счет макросов `all` и `add`, определенных в файле `items`, генерируется таблица или форма, а также добавляются библиотеки, необходимые для их корректной работы.

Заключение. В настоящее время следует отметить непрерывно растущую потребность в системе управления задачами. Данные ПО позволяют оптимизировать и значительно сократить временные и материальные расходы для управления проектами. На данный момент доступно много приложений для управления вашими задачами, найти можно совершенно любой софт, подходящий под нужды вашей, конкретной компании. Проблема в выборе.

В данной работе был проведен обзор методов и систем управления проектами. Рассмотрены популярные продукты реализующие тот или иной функционал для ведения системы учета задач

В ходе данной исследовательской практике, было проведено:

- Ознакомление с готовыми решениями системы учета задач.
- Проведена разработка программного обеспечения для системы учета задач лаборатории средствами языка программирования Java.

Проведя сравнение уже готовых решений, были найдены наиболее популярные ПО, такие как Jira, RedBooth и Redmine. Но данные решения излишне нагружены в плане функциональности и являются не достаточно подходящими под все платформы.

В итоге, можно сказать, что задача управления актуальна и по сей день. Существует уже множество готовых продуктов предоставляющих тот или иной функционал. Найти подходящий продукт, соответствующий абсолютно всем требованиям становится крайне сложно. По этому при высокой востребованности данной системы и простоте в нынешнее время написания собственного программного обеспечения имеет смысл его реализации под свои нужды, исключая излишний функционал предоставляемый уже готовыми продуктами.