МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

Факультет прикладной математики, информатики и механики

Кафедра программного обеспечения   
и администрирования информационных систем

**Разработка web-приложения «kanban-доска»**

Курсовая работа

Направление 02.03.03 – Математическое обеспечение   
и администрирование информационных систем  
Профиль Информационные системы и базы данных

|  |  |  |
| --- | --- | --- |
| Зав. кафедрой\_\_\_\_\_\_\_\_\_\_\_\_\_ | д. ф.-м. н., проф. | М.А. Артёмов |
| Обучающийся\_\_\_\_\_\_\_\_\_\_\_\_ | 2 к. 9 гр. | Ж.С. Данилова |
| Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_ | преп. | И.И. Каширская |

Воронеж 2020

Аннотация

В ходе данной работы было спроектировано web-приложение “Kanban-доски”. Приложение было создано с целью упрощения процесса разработки различных проектов. У разработчиков есть возможность совместного редактирования Kanban-доски, на которой визуализирован процесс выполнения задач.

Содержание

[Содержание 3](#_Toc42692892)

[Введение 4](#_Toc42692893)

[1. Постановка задачи 5](#_Toc42692894)

[2. Анализ задачи 6](#_Toc42692895)

[2.1. Анализ существующих решений 6](#_Toc42692896)

[2.2 Функциональность приложения 7](#_Toc42692897)

[3. Средства реализации 8](#_Toc42692898)

[4. Требования к аппаратному и программному обеспечению 9](#_Toc42692899)

[5. Интерфейс пользователя 10](#_Toc42692900)

[6. Реализация 11](#_Toc42692901)

[7. Тестирование 14](#_Toc42692902)

[Заключение 15](#_Toc42692903)

[Список литературы 16](#_Toc42692904)

[Приложение 1. Листинг 17](#_Toc42692905)

Введение

Существует множество различных методологий разработки программного обеспечения. Одной из самых популярных является agile-методология, основанная на ценностях Манифеста гибкой разработки программного обеспечения.

Один из популярных подходов к реализации agile-разработки ПО – Kanban. Он предполагает обсуждение производительности в режиме реального времени и полную прозрачность рабочих процессов.

Разработка любого IT-продукта состоит из большого количества подзадач. Без систематизации и контроля процесса их выполнения невозможно создать качественный продукт.

Одной из актуальных задач в наше время является создание приложений, позволяющих контролировать необходимые для выполнения задачи. Это необходимо как одному определенному человеку для «бытовых вопросов», так и большим компаниям, занимающихся разработкой. Структурированный список задач помогает руководителям отслеживать производительность сотрудников, скорость выполнения тех или иных задач, составлять отчеты. Они также легко могут назначить сотрудника на задачу или наоборот снять с нее. Разработчикам проще уложиться в сроки, так как не нужно помнить о каждой задаче и времени, отведенном на ее выполнение.

В данной работе рассматривается реализация сформулированной выше задачи.

Когда необходимо оптимизировать перенасыщенные рабочие процессы, обычных to-do листов может быть недостаточно. В этом случае стоит поискать волшебный функционал, который будет отслеживать все задачи, над которыми работает ваша команда, задачи, которые только планируются, а также показывать полную картину по уже выполненным.  
  
Kanban-доска — отличное решение. Этот инструмент для совместной работы над проектами широко используется в разработке ПО, маркетинге, строительстве, логистике и в любых решениях, где присутствует постоянный поток задач. Kanban-подход помогает командам визуализировать рабочие процессы, грамотно анализировать их и повышать эффективность управления задачами.   
Kanban — это метод управления проектами, направленный на минимизацию многозадачности, повышение эффективности производства и оптимизацию скорости и качества работ. Метод широко применяется в крупных командах, стартапах и для индивидуальных целей.

Kanban – японский термин, который начали использовать применительно к производству в 60-х годах 20-го века в компании Toyota. В основу данного принципа положен конвейерный метод производства, а также различные скорости выполнения отдельных технологических операций на производстве.

Важнейшее преимущество Kanban — наглядность.

Еще один прием — разбивать доску на технологические этапы. Разработка программы часто проходит через несколько стадий:

* постановка задачи (разработка технического задания),
* создание кода,
* тестирование,
* сборка рабочей программы,
* передача пользователю.

Создайте на Kanban-доске колонки, соответствующие каждому этапу, и будет легче определить, как движутся задачи в рабочих группах, насколько специалисты загружены и не «провисает» ли работа над проектом.

1. Постановка задачи

Необходимо разработать и реализовать web-приложение для контроля процесса выполнения задач.

В приложении должны быть реализованы:

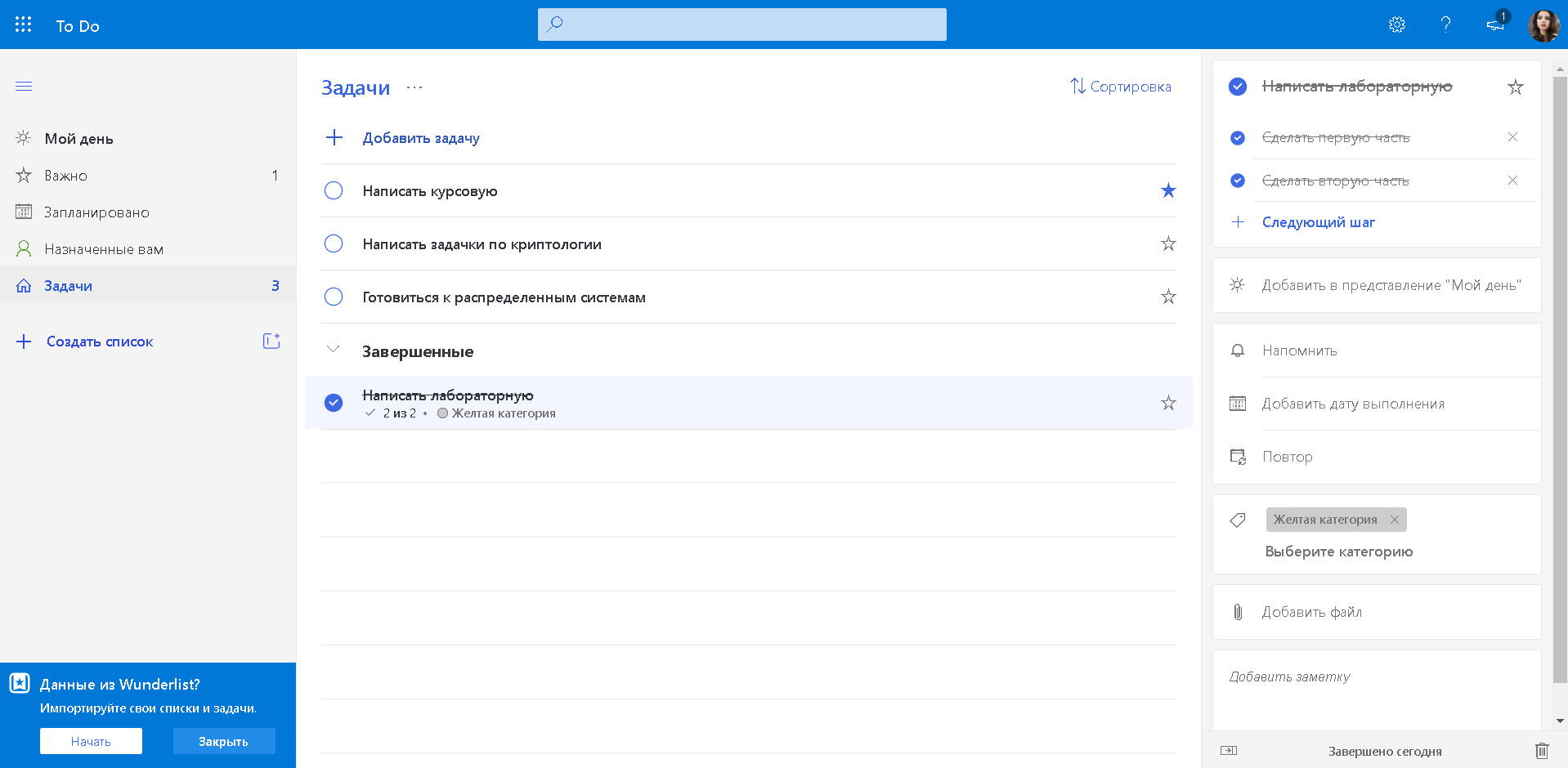
* удобная регистрация разработчиков;
* возможность добавления задач на доску;
* разные доступные действия в зависимости от роли пользователя;
* возможность изменения и удаления задач;
* возможность обновлять статус готовности задачи ответственными за выполнение задачи сотрудниками;
* удобный интерфейс.

2. Анализ задачи

2.1. Анализ существующих решений

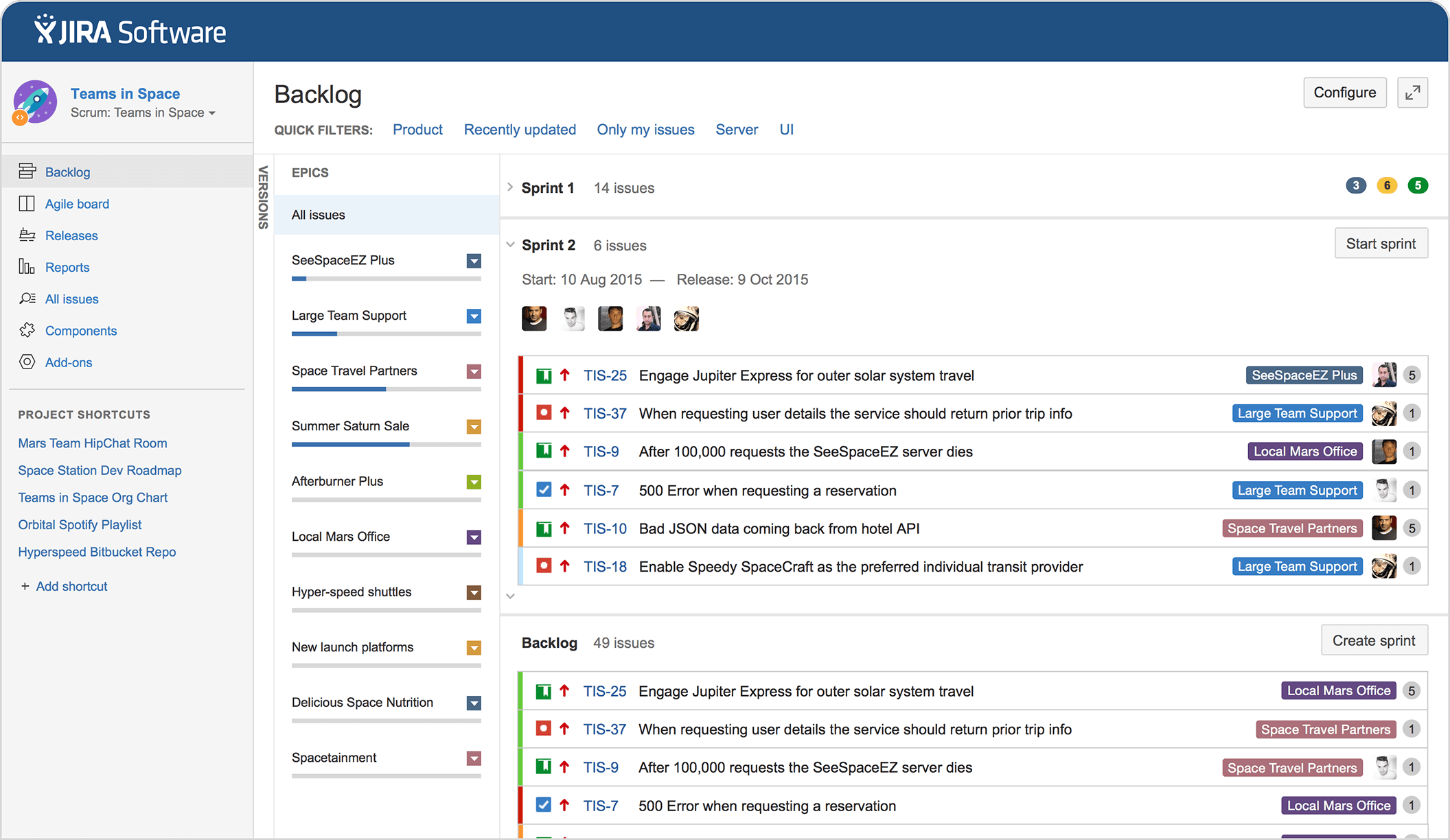
Одним из решения проблемы систематизации задач являются множество приложений типа To Do List (список дел), однако в основном большинство приложений создано для конкретного пользователя и его личных задач, без возможности совместного доступа для команды. Кроме того, в большинстве приложений невозможно указать тип и приоритет задачи, а также обновлять статус выполнения задачи (например, «задача открыта», «задача в процессе реализации» и т.д.).

Например, приложение Microsoft To Do (рис. 2.1) дает возможность добавлять задачи, помечать выборочно как важные, выбирать цвет категории и отмечать дату окончания.

  
рис. 2.1 Web-приложение Microsoft To Do

Однако там нет возможности назначения сотрудников на задачи и нет возможности выбора типа задачи.

Также существуют специальные приложения, созданные для учета задач IT-проектов. Например, Jira – сервис, который позволяет командам-разработчикам планировать проекты, назначать исполнителей задач, выставлять приоритеты и дедлайны (рис. 2.2).

  
рис. 2.2 Приложение Jira

Кроме того, там большой выбор настроек фильтрации и есть возможность составлять детальные отчеты.

Из минусов можно выделить:

* ограниченный функционал бесплатной версии;
* не везде интуитивно понятный интерфейс;
* нет возможности назначения нескольких исполнителей для 1 задачи;
* применение определенных настроек для фильтров, которые будут полезными для команды, может заблокировать уже используемые настройки.

ДОБАВИТЬ БОЛЬШЕ КАРТИНОК К МИНУСАМ

2.2 Функциональность приложения

Разрабатываемое приложение должно предоставлять конечному пользователю удобный интерфейс с возможностью:

3. Средства реализации

Сервер node.js

БД Oracle

Фронт React+Redux

4. Требования к аппаратному и программному обеспечению

Найти в инете для node react оракла и тд

5. Интерфейс пользователя

Регистрация

Вход

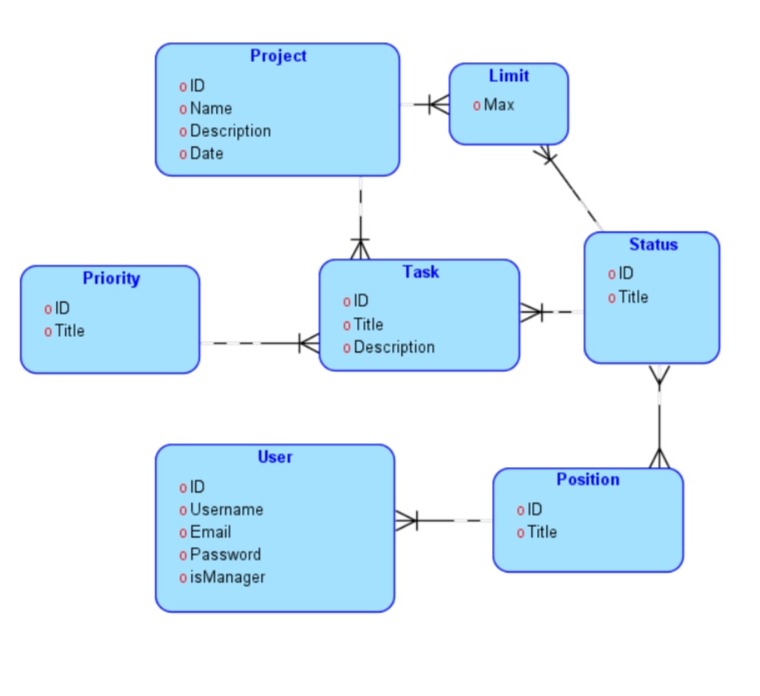
Профиль

Список проектов

Список задач по проекту

6. Реализация

Логическая модель базы данных представлена на рис. 6.1.



1. Проект

Содержит основную информацию о проекте: название, дата начала создания проекта и ID ответственного сотрудника.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Название атрибута** | **Домен** | **NOT NULL** | **PK** | **FK** | **UF** |
| ID\_Проект | Number | + | + | - | + |
| Название | String | + | - | - | + |
| Описание | String | + | - | - | - |
| Дата начала | Date | + | - | - | - |

2. Пользователь

Содержит информацию о сотруднике.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Название атрибута** | **Домен** | **NOT NULL** | **PK** | **FK** | **UF** |
| ID\_Сотрудник | Number | + | + | - | + |
| Имя | String | + | - | - | - |
| Фамилия | String | + | - | - | - |
| Отчество | String | - | - | - | - |
| Дата рождения | Date | + | - | - | - |

3. Информация о должности

Содержит информацию о сотрудниках и должностях.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Название атрибута** | **Домен** | **NOT NULL** | **PK** | **FK** | **UF** |
| ID | Number | + | + | - | + |
| Дата вступления | Date | + | - | - | - |
| Дата окончания | Date | + | - | - | - |
| ID\_Вакансия | Number | - | - | - | - |
| ID\_Сотрудник | Number | + | - | - | - |

Доделать описание модели

Физическая модель данных, описывающая структуру базы данных, представлена на рис. 6.2.

7. Тестирование

Заключение

В ходе данной работы было спроектировано web-приложение для учета задач IT-проектов.

Список литературы

Приложение 1. Листинг