



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе на тему:

Разработка приложения досок Scrum-команды

по дисциплине Базы данных

Студент гр. ИУК4-62Б _____ (подпись) (Карельский М.К.)
(Ф.И.О.)

Руководитель _____ (подпись) (Глебов С.А.)
(Ф.И.О.)

Оценка руководителя _____ баллов _____
30-50 (дата)

Оценка защиты _____ баллов _____
30-50 (дата)

Оценка работы _____ баллов _____
(оценка по пятибалльной шкале)

Комиссия: _____ (подпись) (Глебов С.А.)
(Ф.И.О.)

_____ (подпись) (Гришунов С.С.)
(Ф.И.О.)

_____ (подпись) (Гагарин Ю.Е.)
(Ф.И.О.)

Калуга, 2023

**Калужский филиал
федерального государственного бюджетного образовательного учреждения
высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой _____

_____ (_____)

«____» _____ 20__ г.

**ЗАДАНИЕ
на выполнение курсовой работы**

по дисциплине Базы данных

Студент Карельский М.К. ИУК4-62Б

(фамилия, инициалы, индекс группы)

Руководитель Глебов С.А.

(фамилия, инициалы)

График выполнения работы: 25% к 4 нед., 50% к 7 нед., 75% к 10 нед., 100% к 14 нед.

1. Тема курсовой работы

Разработка приложения досок Scrum-команды

2. Техническое задание

Разработать базу данных, ранние версии клиентской и серверной части приложения
досок Scrum-команды

3. Оформление курсовой работы

3.1. Расчетно-пояснительная записка на 36 листах формата А4.

3.2. Перечень графического материала КР (плакаты, схемы, чертежи и т.п.)

1. Структура БД

2. Демонстрационный чертеж

Дата выдачи задания «09» февраля 2023 г.

Руководитель курсовой работы _____

(подпись)

/

Глебов С.А

(Ф.И.О.)

Задание получил _____

(подпись)

/

Карельский М.К.

(Ф.И.О.)

/

«09» февраля 2023 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. МЕТОДЫ И ИНСТРУМЕНТЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ	5
1.1. Постановка задачи	5
1.2. Описание предметной области	5
1.3. Анализ аналогов	8
1.4. Перечень задач, подлежащих решению в процессе разработки	10
1.5. Обоснование выбора СУБД и средств разработки	10
2. ПРОЕКТИРОВАНИЕ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА..	12
2.1. Разработка базы данных	12
2.2. Описание таблиц базы данных	13
2.3. Описание API	15
3. ТЕСТИРОВАНИЕ И ИНТЕГРАЦИЯ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА	25
3.1. Пользовательское руководство для участника проекта	25
3.2. Пользовательское руководство для создателя проекта	29
ЗАКЛЮЧЕНИЕ	5
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	6

ВВЕДЕНИЕ

Актуальность темы курсовой работы обуславливается широким распространением и активным использованием большим количеством команд разработчиков методологии Scrum. В России данная методика доминирует среди других Agile-подходов.

Объектом работы является разработка desktop-приложений, **предмет** – приложения ОС Windows клиент-серверной архитектуры.

Цель работы – разработка базы данных досок команд разработчиков, работающих по методологии Scrum, а также создание прототипов клиентской и серверной части приложения Windows. В соответствии с целью определяются следующие задачи:

1. Выделить ключевые сущности предметной области,
2. Создать на их основе базу данных,
3. Связать базу данных с сервером, используя ORM,
4. Реализовать API на стороне сервера,
5. Создать приложение клиентской части.

1. МЕТОДЫ И ИНСТРУМЕНТЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ

1.1. Постановка задачи

Задача курсовой работы – разработать раннюю версию приложения досок Scrum-команды, состоящего из следующих частей:

- База данных,
- Сторона сервера,
- Сторона клиента.

1.2. Описание предметной области

Так как Scrum является одной из разновидностей Agile, сперва стоит рассмотреть, что представляет из себя гибкая методология разработки программных продуктов. Это обобщающий термин методик, нацеленных на сведение рабочего процесса к серии коротких циклов – итераций, – результатом которых является реализация очередных функциональных частей итогового продукта.

Основные идеи и принципы Agile описаны в соответствующем манифесте. В нем представлены следующие идеи:

1. Люди и взаимодействие важнее процессов и инструментов;
2. Работающий продукт важнее исчерпывающей документации;
3. Сотрудничество с заказчиком важнее согласования условий контракта;
4. Готовность к изменениям важнее следования первоначальному плану.

Также в документе выделяются следующие принципы:

1. Наивысшим приоритетом признается удовлетворение заказчика за счёт ранней и бесперебойной поставки ценного программного обеспечения;
2. Изменение требований приветствуется даже в конце разработки;

3. Частая поставка работающего программного обеспечения;
4. Общение представителей бизнеса с разработчиками должно быть ежедневным на протяжении всего проекта;
5. Проекты следует строить вокруг заинтересованных людей, которых следует обеспечить нужными условиями работы, поддержкой и доверием;
6. Самый эффективный метод обмена информацией в команде – личная встреча;
7. Работающее программное обеспечение – лучший измеритель прогресса;
8. Спонсоры, разработчики и пользователи должны иметь возможность поддерживать постоянный темп на неопределённый срок;
9. Постоянное внимание к техническому совершенству и хорошему проектированию увеличивают гибкость;
10. Простота, как искусство не делать лишней работы, очень важна;
11. Лучшие требования, архитектура и проектные решения получаются у самоорганизующихся команд;
12. Команда регулярно обдумывает способы повышения своей эффективности и соответственно корректирует рабочий процесс.

В свою очередь, Scrum реализует описанные идеи и принципы следующим образом. Выделяются такие роли, как:

- Product owner – менеджер, отвечающий за общее видение продукта и работающий с заказчиком,
- Scrum master – организатор работы, налаживающий ее в соответствии со Scrum и следящий за выполнением принципов данной методики,
- Команда разработчиков.

Все задачи проекта описываются в специальном документе – backlog'e. Сами задания имеют следующую иерархию:

- Эпики – обобщенные задачи, описывающие большую часть функционала программы,

- Стори – части эпиков, меньшие по размеру и более конкретные по своему содержанию. Могут отвечать на вопрос: «Я, будучи пользователем, хочу... (например, зарегистрироваться)»,
- Таски – части стори, непосредственно описывающие задачи, стоящие перед разработчиками.

Стори могут быть расположены на досках, отражающих, на какой стадии работы над ними они сейчас находятся:

- Ожидание начала выполнения;
- Выполняемые в данный момент;
- Готовые к проверке;
- Отклоненные после проверки, ожидающие исправлений;
- Выполненные.

В процессе работы над проектом периодически проводятся различные события. Самый частый из них – спринт. Это период времени, обычно кратный неделям, во время которого команда должна выполнить несколько стори, на выходе получая функциональный прирост разрабатываемого продукта. Невыполненные задачи переносятся в следующий спринт. Так как разработка должна быть гибкой, спринты обычно не длятся долго (1-4 недели).

В начале каждого спринта проводится его планирование. Эпик с самым большим приоритетом разбивается на стори, которые затем оцениваются специальными единицами – стори поинтами. Их количество зависит от сложности предстоящей работы. Для определения стоимости сперва команда определяется с эталонным стори – таким, с которым когда-то имел дело каждый. Относительно него в дальнейшем будут определяться цены остальных задач.

Происходит это может в виде своеобразной игры в покер. Во время перехода к рассмотрению стоимости очередной стори каждый разработчик в закрытую указывает на своей карте количество стори поинтов, которое он бы начислил данной задаче. Затем все одновременно показывают свои карты и коллективное значение принимается за итоговую цену. Если оценка сильно

отличается, то проводится опрос людей, записавших наибольший и наименьший баллы. Если после повторного голосования результаты все так же неточны, выбирается усредненная оценка.

Составленные на спринт стори распределяются между разработчиками и помещаются на начальную доску. После разбиения стори на задачи определяется, сколько примерно времени может занять их выполнение.

В конце спринта спринта также проводится несколько событий. На ревью клиенту демонстрируются итоги проделанной работы. Во время ретроспективы команда обсуждает, что было сделано хорошо и что – плохо, как можно улучшить свою работу. Также высчитывается количество полученных стори поинтов, чтобы при планировании нового спринта на данное число можно было ориентироваться.

Кроме того, каждый день обычно проводится daily Scrum. На нем разработчики обсуждают, что было сделано вчера и что планируется выполнить сегодня.

1.3. Анализ аналогов

Wrike

Wrike – это программа для управления проектами, подходящая для команд из пяти и более человек и включающая в себя инструменты и функции Scrum. Предоставляет списки задач, подзадачи, расписания, общие рабочие процессы, совместное использование файлов и совместную работу в реальном времени. Позволяет расставлять приоритеты заданий, планировать спринты с помощью диаграмм Ганта и досок Канбан и отслеживать успехи спринтов с помощью отчетов. Цена начинается от 9,80\$ за пользователя в месяц. Существует четыре разных ценовых уровня, включая бесплатную версию.

Jira

Jira – одно из самых популярных доступных программных средств agile Scrum. Предоставляет такие возможности, как пользовательские фильтры, настраиваемые Scrum доски и визуальные отчеты по проектам.

Однако, несмотря на то, что Jira довольно хорошо сбалансирована с точки зрения функциональных возможностей, новичкам в управлении Scrum проектами может показаться немного сложным ее использование.

Jira является частью экосистемы Atlassian. Стоимость Jira начинается от 7\$ в месяц для групп из более чем 10 пользователей. Также предлагается бесплатное лицензирование для команд меньших размеров.

Monday.com

Имеет интуитивно понятный и хорошо организованный интерфейс. Бесплатный план включает неограниченное количество досок и множество шаблонов. Также существуют приложения для iOS и Android. Предлагается 14-дневная пробная версия, начальная цена – 8\$ за пользователя в месяц.

ClickUp

Данная платформа позиционирует себя как универсальный продукт. Ее разработчики заявляют, что она объединяет в себе все функции и возможности, которые могут понадобиться команде, которая использует методологию Scrum.

Имеются такие возможности, как работа с основными задачами и документацией, мониторинг достижения поставленных задач, командный чат. Количество задач и участников в бесплатной версии ничем не ограничивается. Платные тарифы начинаются с 5\$ в месяц за каждого пользователя.

1.4. Перечень задач, подлежащих решению в процессе разработки

Разработка базы данных включает в себя реализацию таких сущностей, как:

- Разработчики,
- Проекты,
- Спринты,
- Задачи, сохраняющие иерархию, соответствующую предметной области.

С составленной базой данных должен работать сервер, использующий ORM. Связь между клиентской и серверной стороной реализуется посредством API.

Приложение должно иметь следующий функционал:

- Регистрация и авторизация пользователей,
- Личный кабинет, позволяющий изменить личные данные,
- Создание новых проектов, добавление в них других разработчиков,
- Составление backlog'а, хранящего внутри себя эпика, стори и задачи,
- Создание спринтов, включение в них составленных стори,
- Перемещение спринтов между досками.

1.5. Обоснование выбора СУБД и средств разработки

В качестве реляционной СУБД был выбран PostgreSQL, так как он доступный, имеет большое распространение и работает с Entity Framework Core.

Для реализации серверной части использовался фреймворк ASP.NET. Он достаточно популярен, стабильный и надежный. За язык программирования был взят широко известный и используемый C#. Средой разработки выступила Visual Studio 2022, ведь она имеет поддержку последней версии ASP.NET 6.0 и

технологии IntelliSense. Для тестирования реализованного API использовался Swagger.

Языком программирования для разработки клиента был выбран Python, так как он обладает огромной популярностью и сравнительно прост в работе. Основной библиотекой языка выступил PyQt5. Данный фреймворк позволяет создавать современный интерфейс, сравнительно легкий в освоении, имеет хорошую документацию и большое сообщество. В связке с данной библиотекой появилась возможность использовать QtDesigner – программу для удобного визуального составления окон продукта. Средой разработки был выбран Visual Studio Code, предоставляющий быструю и удобную работу с Python кодом.

2. ПРОЕКТИРОВАНИЕ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА

2.1. Разработка базы данных

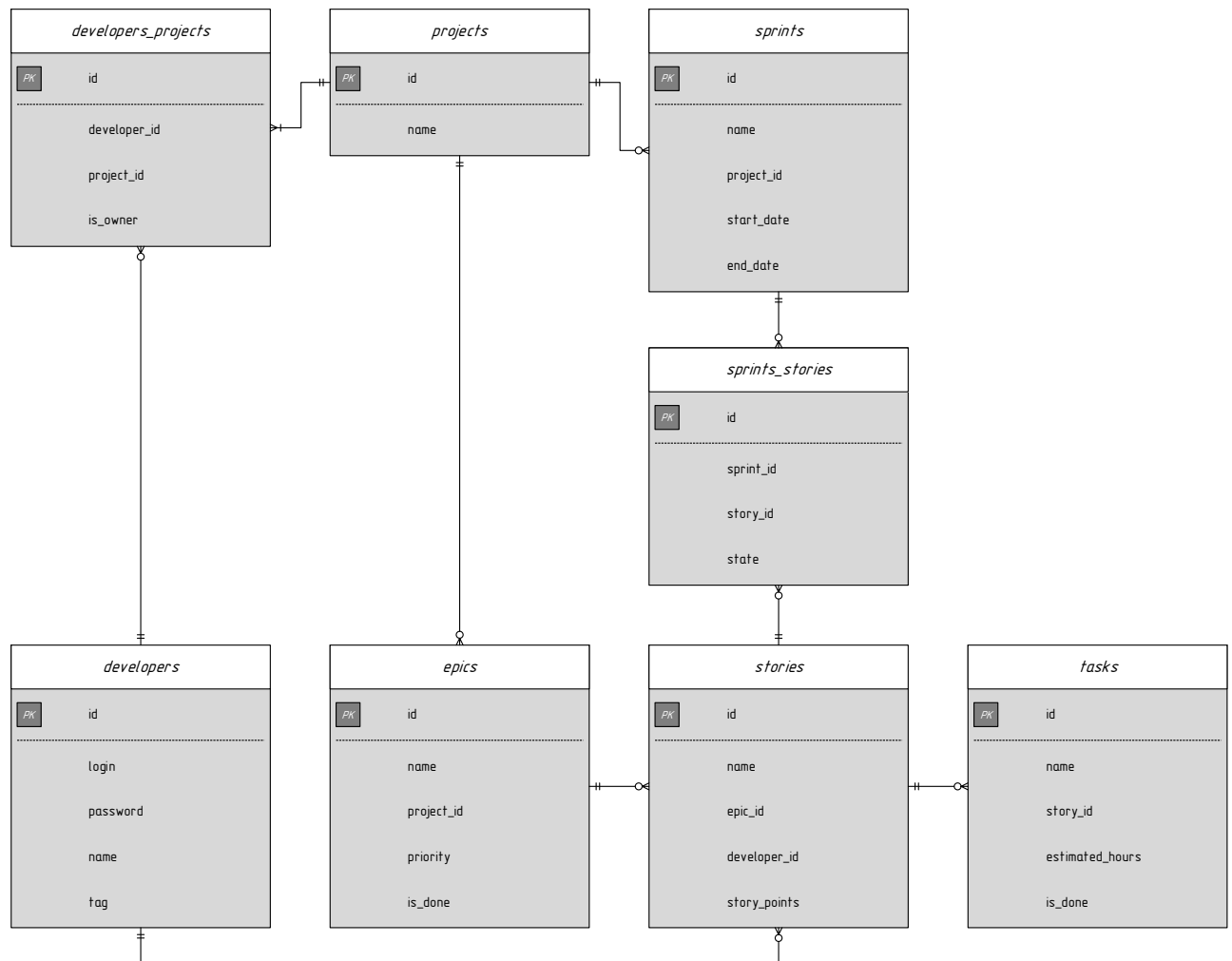


Рис. 1. База данных

База данных состоит из следующих таблиц (см. рис. 1):

- **developers** – хранит данные пользователей-разработчиков, используемые для авторизации и их поиска;
- **projects** – хранит данные проектов, создаваемых пользователями;
- **developers_projects** – таблица реализации связи «многое-ко-многим» между таблицами **developers** и **projects**;
- **epics** – таблица эпиков, записанных в backlog'е проекта;

- stories – таблица стори, перечисленных в созданных эпиках;
- tasks – таблица тасков, на которые разбиваются составленные стори;
- sprints – таблица спринтов, перечисленных в проекте;
- sprints_stories – таблица реализации связи «многое-ко-многим» между таблицами sprints и stories, позволяющая распределять стори по нужным спринтам.

В базе данных также было создано 2 типа данных:

- priority – предоставляет 5 уровней приоритета выполнения эпиков:
 - minor,
 - low,
 - medium,
 - high,
 - major.
- process_state – предоставляет 5 возможных состояний процесса выполнения стори, соответствующих доскам предметной области:
 - todo,
 - rejected,
 - in_progress,
 - ready_for_testing,
 - done.

2.2. Описание таблиц базы данных

Таблицы базы данных имеют следующие столбцы (см. табл. 1):

Таблица	Название	Тип	NN	U	PK	FK	Default
developers	id	integer	+	+	+		
	<i>Идентификатор</i>						
	login	text	+	+			
	<i>Логин, используемый пользователем при авторизации</i>						
	password	text	+				

	Пароль, используемый пользователем при авторизации						
	name	text	+				
	Имя разработчика, отображаемое другим пользователям						
	tag	text	+	+			
	Тег, по которому происходит поиск разработчика во время добавления его в участники проекта						
projects	id	integer	+	+	+		
	Идентификатор						
	name	text	+				new project
	Название проекта						
developers_projects	id	integer	+	+	+		
	Идентификатор						
	developer_id	integer	+			developers.id	
	Идентификатор участника проекта						
	project_id	integer	+			projects.id	
	Идентификатор проекта, в который был добавлен разработчик						
	is_owner	boolean	+				false
epics	Является ли участник владельцем (создателем) проекта						
	id	integer	+	+	+		
	Идентификатор						
	name	text	+				new epic
	Название эпика						
	project_id	integer	+			projects.id	
	Идентификатор проекта, backlog которого содержит данный эпик						
	priority	priority					
	Приоритет выполнения эпика						
	is_done	boolean	+				false
stories	Является ли эпик завершенным						
	id	integer	+	+	+		
	Идентификатор						
	name	text	+				new story
	Название стори						
	epic_id	integer	+			epics.id	
	Идентификатор эпика, в составе которого находится данная стори						
	developer_id	integer				developers.id	
	Идентификатор разработчика, назначенного на выполнение данной стори						
	story_points	integer					
tasks	Количество определенных командой стори поинтов						
	id	integer	+	+	+		
	Идентификатор						
	name	text	+				new task

	Название задачи						
	story_id	integer	+			stories.id	
	Идентификатор стори, в составе которой находится данный task						
	estimated_hours	integer					
	Примерное ожидаемое количество часов, нужное для решения данного taska						
	is_done	boolean	+				false
	Является ли task выполненным						
sprints	id	integer	+	+	+		
	Идентификатор						
	name	text	+				new sprint
	Название спринта						
	project_id	integer	+			projects.id	
	Идентификатор проекта, в рамках которого проводится данный спринт						
	start_date	date	+				
	Дата начала спринта						
	end_date	date	+				
	Дата завершения спринта						
sprints_stories	id	integer	+	+	+		
	Идентификатор						
	sprint_id	integer	+			sprints.id	
	Идентификатор спринта, во время которого будет решаться данная стори						
	story_id	integer	+			stories.id	
	Идентификатор стори, выполняемой в данном спринте						
	state	process_state	+				todo
	Текущее состояние выполнения стори (доска, на которой она находится)						

Табл. 1. Описание таблиц базы данных

2.3. Описание API

/Signup – регистрация пользователя. Через тело запроса получает параметры:

- name – имя пользователя,
- login – логин,
- password – пароль.

/Login – авторизация пользователя. Через тело запроса получает параметры:

- login – логин,
- password – пароль.

Возвращает поля:

- name – имя пользователя,
- tag – тег пользователя,
- token – JWT токен, используемый в дальнейшем для обращения к API, требующему авторизацию.

/EditDeveloperName – изменение имени пользователя. Через тело запроса получает параметры:

- name – новое имя пользователя.

/EditDeveloperPassword – изменение пароля пользователя. Через тело запроса получает параметры:

- oldPassword – нынешний пароль пользователя,
- newPassword – новый пароль пользователя.

/AddProjectDeveloper – добавление нового участника проекта. Через тело запроса получает параметры:

- tag – тег разработчика,
- projectId – идентификатор проекта.

/GetProjectDevelopers – получение списка участников проекта. Через строку запроса получает параметры:

- projectId – идентификатор проекта.

Возвращает поля:

- `isMember` – является ли пользователь, совершающий запрос, участником проекта,
- `developers` – список участников, содержит поля:
 - `id` – идентификатор участника,
 - `name` – имя участника,
 - `isOwner` – является ли участник создателем проекта.

`/RemoveProjectDeveloper` – исключение разработчика из участников проекта. Через тело запроса получает параметры:

- `projectId` – идентификатор проекта,
- `developerId` – идентификатор участника.

`/AddProject` – создание нового проекта. Через тело запроса получает параметры:

- `name` – название проекта.

Возвращает поля:

- `name` – название проекта.

`/GetProjects` – получение списка проектов, в которых разработчик является участником. Возвращает список, содержащий поля:

- `id` – идентификатор проекта,
- `name` – название проекта,
- `owner` – имя создателя проекта,
- `isOwner` – является ли пользователь, совершающий запрос, создателем проекта.

`/GetProject` – получение данных проекта. Через строку запроса получает параметры:

- `projectId` – идентификатор проекта.

Возвращает поля:

- projectName – название проекта.

/EditProjectName – изменение имени проекта. Через тело запроса получает параметры:

- projectId – идентификатор проекта,
- name – новое название проекта.

/RemoveProject – удаление проекта. Через тело запроса получает параметры:

- projectId – идентификатор проекта.

/AddEpic – добавление эпика в backlog проекта. Через тело запроса получает параметры:

- name – название эпика,
- projectId – идентификатор проекта
- priority – приоритет эпика.

/GetEpics – получение эпиков backlog'а проекта. Через строку запроса получает параметры:

- projectId – идентификатор проекта.

Возвращает список, содержащий поля:

- id – идентификатор эпика,
- name – название эпика,
- priority – приоритет эпика,
- isDone – является ли эпик завершенным.

/EditEpicName – изменение названия эпика. Через тело запроса получает параметры:

- epicId – идентификатор эпика,
- name – новое название эпика.

/EditEpicPriority – изменение приоритета эпика. Через тело запроса получает параметры:

- epicId – идентификатор эпика,
- priority – новый приоритет эпика.

/EditEpicIsDone – изменение статуса завершенности эпика. Через тело запроса получает параметры:

- epicId – идентификатор эпика,
- isDone – новый статус завершенности эпика.

/RemoveEpic – удаление эпика. Через тело запроса получает параметры:

- epicId – идентификатор эпика.

/AddStory – добавление в эпик новой стори. Через тело запроса получает параметры:

- name – название стори,
- epicId – идентификатор эпика,
- storyPoints – количество стори поинтов.

/GetStories – получение стори эпика. Через строку запроса получает параметры:

- epicId – идентификатор эпика.

Возвращает список, содержащий поля:

- id – идентификатор стори,
- name – название стори,

- developerId – идентификатор разработчика, назначенного на выполнение стори,
- storyPoints – количество стори поинтов.

/EditStoryName – изменение имени стори. Через тело запроса получает параметры:

- storyId – идентификатор стори,
- name – новое название стори.

/EditStoryPoints – изменение количества стори поинтов стори. Через тело запроса получает параметры:

- storyId – идентификатор стори,
- storyPoints – новое количество стори поинтов.

/EditStoryDeveloper – изменение назначенного на выполнение стори разработчика. Через тело запроса получает параметры:

- storyId – идентификатор стори,
- developerId – идентификатор разработчика.

/RemoveStory – удаление стори. Через тело запроса получает параметры:

- storyId – идентификатор стори.

/AddTask – добавление в стори нового таска. Через тело запроса получает параметры:

- name – название таска,
- storyId – идентификатор стори,
- estimatedHours – ожидаемое количество часов, необходимое на выполнение таска.

/GetTasks – получение задач стори. Через строку запроса получает параметры:

- storyId – идентификатор стори.

Возвращает список, содержащий поля:

- id – идентификатор задачи,
- name – название задачи,
- estimatedHours – ожидаемое количество часов на выполнение задачи,
- isDone – является ли задача выполненной.

/EditTaskName – изменение имени задачи. Через тело запроса получает параметры:

- taskId – идентификатор задачи,
- name – новое название задачи.

/EditTaskEstimatedHours – изменение ожидаемого количества часов на выполнение задачи. Через тело запроса получает параметры:

- taskId – идентификатор задачи,
- estimatedHours – новое ожидаемое количество часов на выполнение задачи.

/EditTaskIsDone – изменение статуса завершенности задачи. Через тело запроса получает параметры:

- taskId – идентификатор задачи,
- isDone – новый статус завершенности задачи.

/RemoveTask – удаление задачи. Через тело запроса получает параметры:

- taskId – идентификатор задачи.

/AddSprint – добавление в проект нового спринта. Через тело запроса получает параметры:

- name – название спринта,
- projectId – идентификатор проекта,
- startDate – дата начала спринта,
- endDate – дата окончания спринта.

/GetSprints – получение спринтов проекта. Через строку запроса получает параметры:

- projectId – идентификатор проекта.

Возвращает список, содержащий поля:

- id – идентификатор спринта,
- name – название спринта,
- startDate – дата начала спринта,
- endDate – дата окончания спринта.

/GetStoriesAndEpics – получение эпиков проекта вместе со стори для изменения содержания спринта. Через строку запроса получает параметры:

- projectId – идентификатор проекта
- sprintId – идентификатор спринта.

Возвращает список, содержащий поля:

- id – идентификатор эпика,
- name – название эпика,
- priority – приоритет эпика,
- isDone – является ли эпик завершенным
- stories – список стори эпика, содержащий поля:
 - id – идентификатор стори,
 - name – название стори,
 - storyPoints – количество стори поинтов,
 - isPicked – включена ли стори в спринт.

`/EditStoryIsPicked` – изменение статуса включенности в спринт стори.

Через тело запроса получает параметры:

- `isPicked` – включена ли стори в спринт,
- `sprintId` – идентификатор спринта,
- `storyId` – идентификатор стори.

`/EditSprintName` – изменение названия спринта. Через тело запроса получает параметры:

- `id` – идентификатор спринта,
- `name` – новое название спринта.

`/EditSprintDate` – изменение одной из дат спринта. Через тело запроса получает параметры:

- `id` – идентификатор спринта,
- `dare` – новая дата спринта,
- `isStart` – является ли указанная дата начальной.

`/RemoveSprint` – удаление спринта. Через тело запроса получает параметры:

- `sprintId` – идентификатор спринта.

`/GetSprintStories` – получение стори спринта. Через строку запроса получает параметры:

- `sprintId` – идентификатор спринта.

Возвращает список, содержащий поля:

- `id` – идентификатор стори,
- `name` – название стори,
- `state` – состояние выполнения стори (на какой доске находится),
- `epicName` – название эпика, в составе которого находится стори,

- storyPoints – количество стори поинтов,
- developerName – имя назначенного на выполнение стори разработчика,
- isAssigned – является ли пользователь, совершающий запрос, назначенным на выполнение стори.

/EditStoryState – изменение состояния стори (перемещение его на другую доску). Через тело запроса получает параметры:

- storyId – идентификатор стори,
- state – новое состояние (доска) стори.

/GetProjectChanges – получение числа изменений частей проекта. Через строку запроса получает параметры:

- projectId – идентификатор проекта.

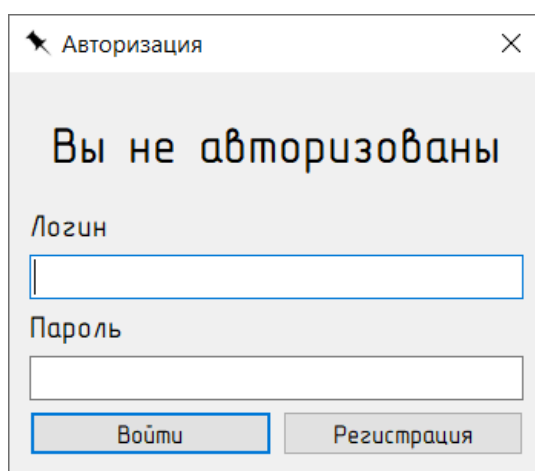
Возвращает словарь, содержащий поля:

- id – идентификатор проекта,
- name – количество изменений названия проекта,
- members – количество изменений списка участников проекта,
- sprints – количество изменений спринтов проекта,
- boards – количество изменений досок проекта,
- backlog – количество изменений backlog'а проекта.

3. ТЕСТИРОВАНИЕ И ИНТЕГРАЦИЯ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА

3.1. Пользовательское руководство для участника проекта

После запуска программы пользователю необходимо ввести логин и пароль, чтобы авторизоваться в системе (см. рис. 2).



Авторизация

Вы не авторизованы

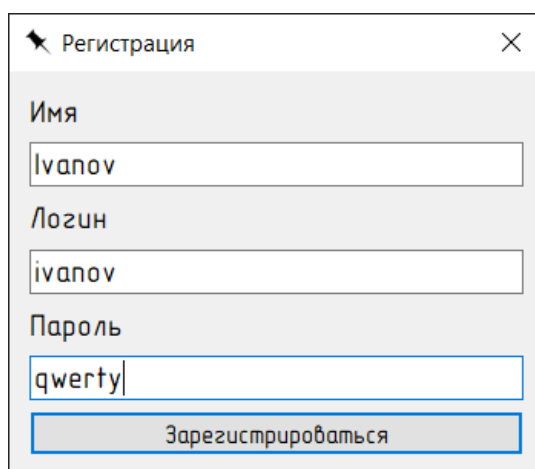
Логин

Пароль

Войти Регистрация

Рис. 2. Окно авторизации

Нажав на кнопку «Регистрация», пользователь перейдет в другое окно, где программа предложит ему ввести личные данные и создать новый аккаунт (см. рис. 3).



Регистрация

Имя

Ivanov

Логин

ivanov

Пароль

qwerty

Зарегистрироваться

Рис. 3. Окно регистрации

Пройдя авторизацию, пользователь попадает на главное окно (см. рис. 4). Здесь он может создать новый проект с помощью нажатия на кнопку «Новый проект». Ниже выводится список проектов, участником которых является пользователь. Если пользователь не является создателем проекта, он может перестать быть его участником с помощью нажатия на кнопку «Выйти», которая находится на строке проекта, из которого хочет выйти пользователь.

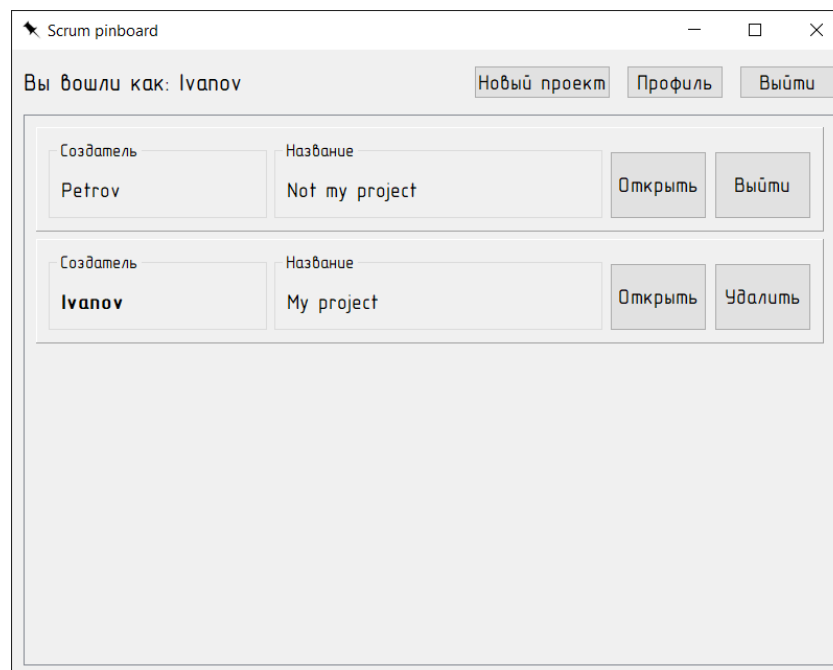


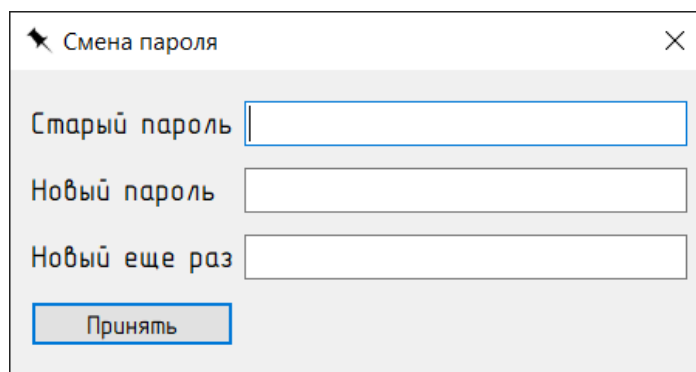
Рис. 4. Главное окно

Нажав кнопку «Выйти», находящуюся в правом верхнем углу, пользователь вернется обратно на окно авторизации. Кнопка «Профиль» открывает окно, содержащее личную информацию пользователя (см. рис. 5) и предоставляющее возможность изменить имя и пароль нажатием соответственно на кнопки «Изменить» и «Сменить пароль».



Рис. 5. Окно профиля пользователя

Для смены пароля необходимо указать прежний пароль и дважды новый (см. рис. 6):



Смена пароля

Старый пароль

Новый пароль

Новый еще раз

Принять

Рис. 6. Окно смены пароля

После нажатия на кнопку «Открыть», находящуюся на строке проекта, который пользователь хочет открыть, произойдет переход на окно выбранного проекта (см. рис. 7).



Проект

Backlog

Участники

Ivanov

Petrov

Not my project

Some sprint

Начало 01.05.2023 Конец 14.05.2023

To do Rejected In progress Ready for testing Done

Petrov's story

Ivanov's story

Another story

Рис. 7. Окно проекта со стороны добавленного участника

Сверху отображается название проекта. Слева приводится список участников, где жирным шрифтом отмечается создатель проекта. Под названием проекта находится кнопка выпадающего списка, отвечающая за выбор между спринтами проекта. Справа от этой кнопки выводятся даты начала и окончания выбранного спринта.

Снизу расположены 5 досок этапов выполнения стори. На них отображаются занесенные в выбранный спринт стори. Жирным шрифтом отмечаются те стори, назначенным на выполнение которых является смотрящий пользователь. Курсивом отмечены те стори, для которых еще не было выделено разработчика.

Если нажать на один из стори, появится окно, содержащее информацию о нем (см. рис. 8): название стори и эпика, в составе которого она находится, назначенный на выполнение разработчик, количество стори поинтов. Ниже находится кнопка выпадающего списка, отвечающая за расположение стори на одной из досок. В самом низу располагается список задач стори с указанием их названия, ожидаемого количества часов их выполнения и статуса завершенности.

The screenshot shows a window titled "Стори" (Story) with a close button. The main content area is titled "Petrov's story". It contains several input fields: "Эпик" (Epic) with the value "Some epic", "Разработчик" (Developer) with the value "Petrov", and "S. pts" (Story points) with the value "20". Below these is a dropdown menu labeled "На выполнение" (For execution) with a downward arrow. At the bottom, there is a table with three columns: "Task", "Часы" (Hours), and "Готово" (Ready). The first row of the table has the values "Some task", "3", and "Готово".

Task	Часы	Готово
Some task	3	Готово

Рис. 8. Окно стори

В окне проекта над списком участников находится кнопка «Backlog», после нажатия на которую открывается одноименное окно (см. рис. 9):

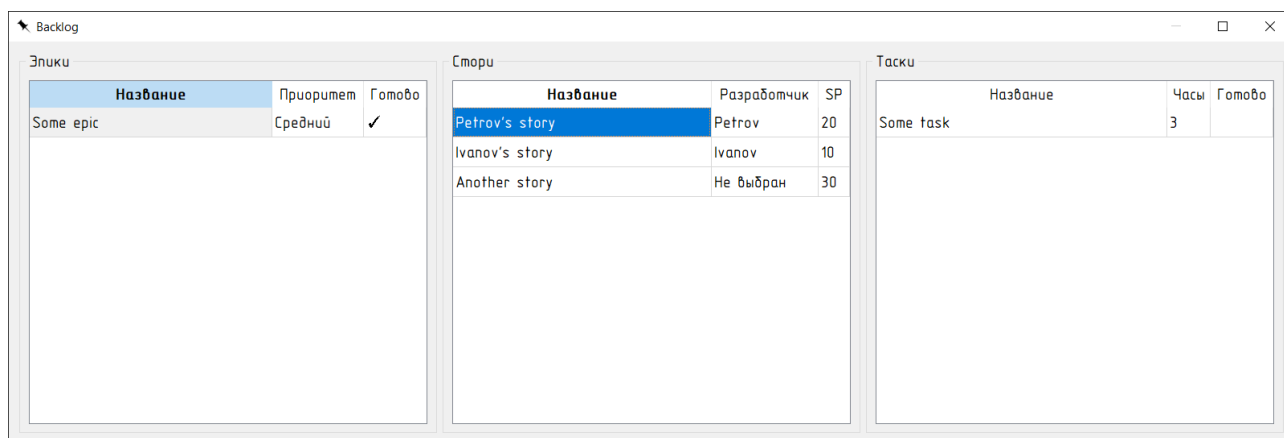


Рис. 9. Окно backlog'a со стороны добавленного участника

В данном окне присутствуют 3 области. В первой из них отображен список эпиков с указанием их названия, приоритета выполнения и статуса завершенности. При выборе одного из эпиков на второй области выводится список содержащихся внутри стори с указанием их названия, назначенного на их выполнение разработчика и количества стори поинтов. В свою очередь при выборе одного из стори в третьей области появляется список задач, из которых состоит данный стори. В нем отображаются названия задач, ожидаемое время их выполнения в часах и статус завершенности.

3.2. Пользовательское руководство для создателя проекта

Если текущий пользователь является создателем одного из проектов, то на главном окне кнопка «Выйти», находящаяся на строке проекта, будет заменена на кнопку «Удалить», нажатие на которую приводит к удалению соответствующего проекта.

Открыв проект, создателем которого является текущий пользователь, в окне проекта он сможет обнаружить ряд изменений (см. рис. 10).

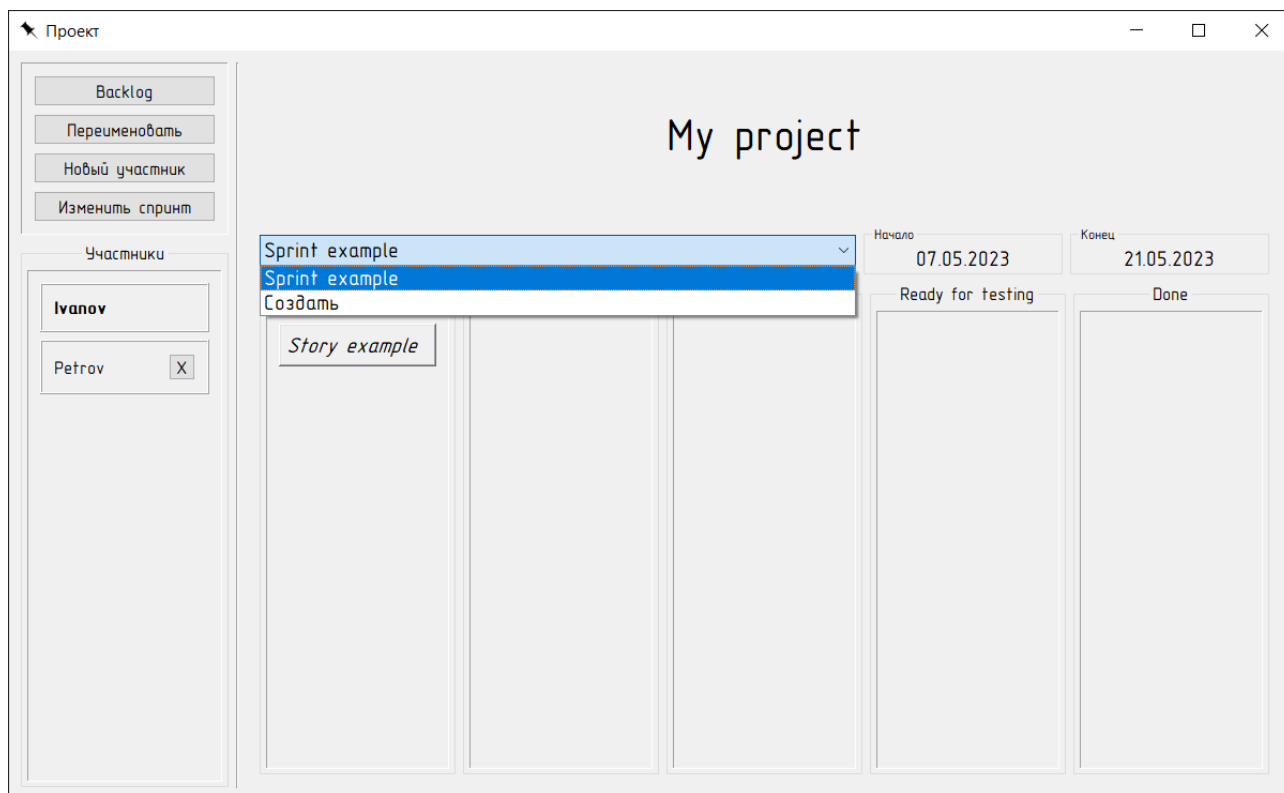


Рис. 10. Окно проекта со стороны создателя

В списке участников появятся кнопки, позволяющие отстранять их от участия в проекте.

В выпадающем списке выбора спринта появится новая опция создания, выбор которой приводит к появлению окна создания спринта (см. рис. 11). В данном окне необходимо указать название и даты начала и окончания спринта.

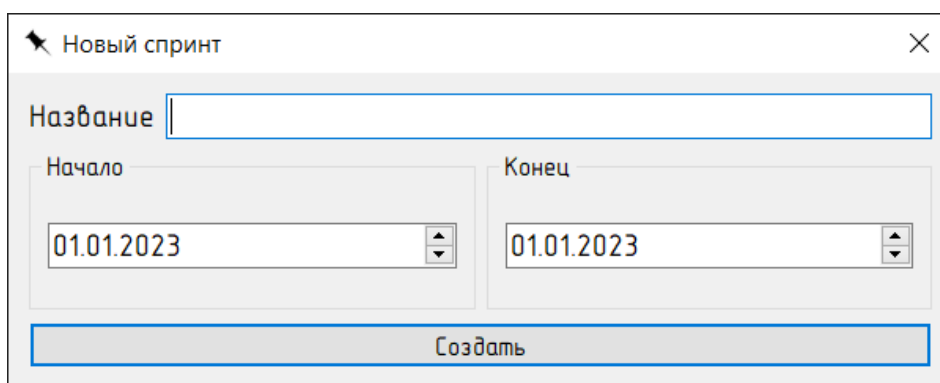


Рис. 11. Окно создания спринта

Над списком участников также появятся новые кнопки: «Переименовать» позволяет изменить название проекта, «Новый участник» предлагает

пользователю добавить участника через поиск по его тегу, «Изменить спринт» откроет соответствующее выбранному спринту окно (см. рис. 12).

Еpic example	Низкий	Выполняется
<input checked="" type="checkbox"/>	Story example	30 sp

Рис. 12. Окно изменения спринта

Здесь пользователь может переименовать спринт, изменить даты его начала и окончания, а также выбрать, какие стори стоит в него включить. Последнее реализуется с помощью области, находящейся снизу и содержащей список эпиков с отображением их названия, приоритета и статуса завершенности. Элементы списка также содержат внутри себя списки соответствующих стори. В них указываются названия стори, количества стори поинтов и кнопки, позволяющие включать стори в изменяемый спринт. В нижнем правом углу находится кнопка «Удалить», отвечающая за удаление спринта.

В окне backlog'а также появляются новые возможности (см. рис. 13).

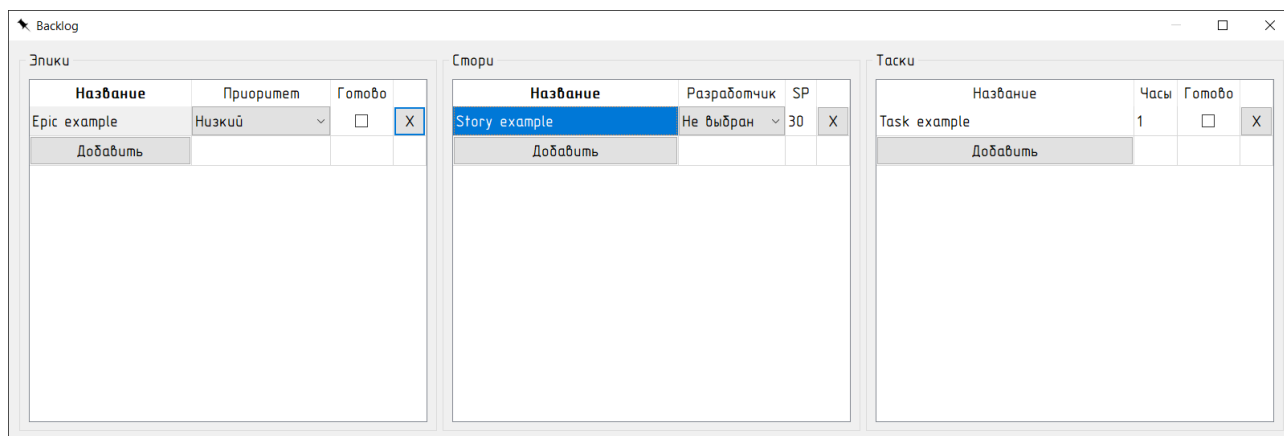


Рис. 13. Окно backlog'a со стороны создателя

Теперь пользователь может двойным нажатием изменять названия задач, количество стори поинтов и ожидаемых часов выполнения задач, с помощью соответствующих выпадающих списков указывать приоритеты задач и назначать на выполнение стори разработчиков, а также специальных кнопок отмечать статус завершенности эпиков и задач. Справа от каждой строки находятся кнопки, предназначенные для удаления задачи. В конце каждого списка находятся кнопки для добавления новых задач, нажатие на которые приводит к вызову соответствующих окон (см. рис. 14-16).

The screenshot shows a dialog box titled 'Новый эпик' (New Epic). It contains the following fields and buttons:

- Название** (Name): A text input field.
- Приоритет** (Priority): A dropdown menu with the selected value 'Незначительный' (Insignificant).
- Создать** (Create): A button to create the new epic.

Рис. 14. Окно создания эпика

Для создания эпика необходимо указать его название и приоритет.



Новый стори

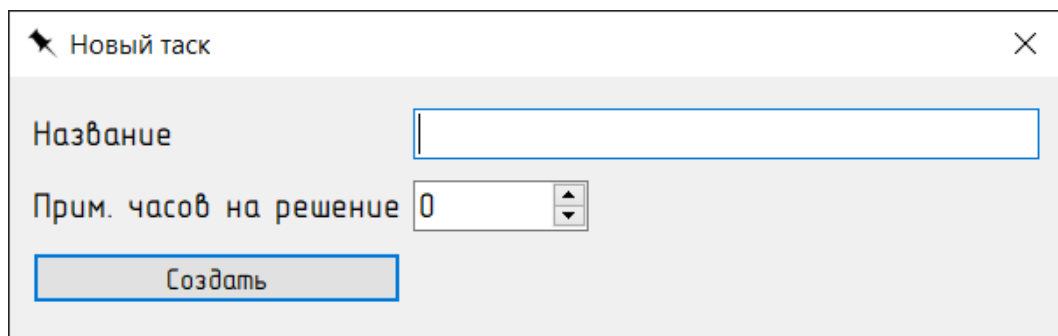
Название

Story points

Создать

Рис. 15. Окно создания стори

Для создания стори необходимо указать ее название и количество стори поинтов.



Новый таск

Название

Прим. часов на решение

Создать

Рис. 16. Окно создания таска

Для создания спринта необходимо указать его название и ожидаемое количество часов, необходимое на его выполнение.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы были получены обширные навыки в сфере работы с клиент-серверной архитектурой, а также базой данных PostgreSQL, библиотекой PyQt5 и фреймворком ASP.NET.

Результатом работы стали прототипы приложений клиентской и серверной сторон, дающие представление об основных функциональных возможностях данного программного продукта и, в целом, позволяющие вести работу в соответствии с методологией Scrum уже на данном этапе.

Продукт имеет множество сторон, доступных для дальнейшего развития и улучшения, например, интерфейс клиента, его производительность и синхронизация, валидация данных, безопасность и защищенность сервера и т.д.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Карпова, Т.С. Базы данных: модели, разработка, реализация: учебное пособие / Т.С. Карпова. - 2-е изд., исправ. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 241 с.: ил.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429003>
2. Давыдова, Е.М. Базы данных [Электронный ресурс]: учеб. пособие / Е.М. Давыдова, Н.А. Новгородова. — Электрон. дан. — Москва: ТУСУР, 2007. — 166 с. — Режим доступа: <https://e.lanbook.com/book/11636>. — Загл. с экрана.
3. Харрингтон, Д. Проектирование объектно ориентированных баз данных [Электронный ресурс] — Электрон. дан. — Москва: ДМК Пресс, 2007. — 272 с. — Режим доступа: <https://e.lanbook.com/book/1231>. — Загл. с экрана.
4. Голицина О.Л., Максимов Н.В., Попов И.И. Базы данных: учеб. пособие. — М.: Форум:Инфра-М, 2007.
5. Гагарин Ю.Е. Применение языка SQL в MS Access: учебнометодическое пособие. – М.: МГТУ им. Н.Э. Баумана, 2012.
6. Джуба, С. Изучаем PostgreSQL 10 / С. Джуба, А. Волков. — М.: ДМК Пресс, 2019.
7. Гутман, Г. Н. Объектно-реляционная СУБД PostgreSQL: учебное пособие / Г. Н. Гутман. — Самара: АСИ СамГТУ, 2016.
8. Разработка приложений на C# с использованием СУБД PostgreSQL: учебное пособие / И. А. Васюткина, Г. В. Трошина, М. И. Бычков, С. А. Менжулин. — Новосибирск: НГТУ, 2015.
9. Лок, Э. ASP.NET Core в действии: руководство / Э. Лок; перевод с английского Д. А. Беликова. — М.: ДМК Пресс, 2021.
10. Снетков, В. М. Программирование на ASP.NET: учебное пособие / В. М. Снетков. — 2-е изд. — М.: ИНТУИТ, 2016.

11. Конфигурирование и настройка Microsoft ASP.NET: учебное пособие. — 2-е изд. — М.: ИНТУИТ, 2016.
12. Тарланов, А. Т. Знакомство с библиотекой PyQT: учебно-методическое пособие / А. Т. Тарланов, Е. С. Карбова. — М.: РТУ МИРЭА, 2021.
13. Тарланов, А. Т. Базы данных и дополнительные компоненты библиотеки PyQT: учебно-методическое пособие / А. Т. Тарланов. — М.: РТУ МИРЭА, 2021.
14. Базы данных: учебно-методическое пособие / Г. И. Ревунков, Н. А. Ковалева, Е. Ю. Силантьева [и др.]. — М.: МГТУ им. Н.Э. Баумана, 2020.
15. Кузниченко, М. А. Основы баз данных: учебно-методическое пособие / М. А. Кузниченко. — 2-е изд., стер. — М.: ФЛИНТА, 2022.