

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

Факультет Компьютерных наук

Кафедра технологий обработки и защиты информации

Курсовой проект

Сайт для учёта рабочего времени «ITime»

09.03.02 Информационные системы и технологии

Обучающийся _____ *Смирнов А.К., 3 курс, д/о*

Обучающийся _____ *Железной А.С., 3 курс, д/о*

Обучающийся _____ *Шаталов А.А., 3 курс, д/о*

Руководитель _____ старший преподаватель *Тарасов В.С.*

Воронеж 2021

Содержание

1	Введение	4
2	Постановка задачи	5
3	Анализ предметной области	6
3.1	Глоссарий	6
3.2	Анализ существующих решений	7
3.3	Анализ задачи	9
3.3.1.	Варианты использования приложения.....	9
3.3.2.	Взаимодействие компонентов системы	10
3.3.3.	Варианты состояния системы	11
3.3.4.	Варианты действия в системе	12
3.3.5.	Развёртывание приложения.....	14
3.3.6.	IDEF0	15
4	Реализация	16
4.1	Сущности.....	16
4.2	Сценарии воронок конверсии.....	20
4.3	Графический интерфейс	20
4.3.1.	Страница авторизации	20
4.3.2.	Страница профиля	20
4.3.2.1.	Страница изменения профиля.....	21
4.3.3.	Страница проектов	22
4.3.3.1.	Страница описания проекта	22
4.3.3.2.	Страница заданий проекта	23
4.3.3.2.1.	Страница описания задания проекта.....	24
4.3.3.2.2.	Страница уведомлений задания проекта	24
4.3.3.2.3.	Страница разработчиков над заданием проекта	25
4.3.3.3.	Страница разработчиков проекта	26
4.3.3.4.	Страница добавления проекта	26
4.3.4.	Страница разработчиков.....	27
4.3.5.	Страница уведомления.....	27
4.3.6.	Страница управления временем	28
4.3.7.	Страница достижений	29

5	Тестирование	30
5.1	Дымовое тестирование.....	30
5.2	UI тесты	31
5.3	Юзабилити тестирование.....	34
6	Заключение	36

1 Введение

В современном мире есть множество компаний, предприятий, которым надо следить за работой сотрудников. Это можно делать разными способами, но в 21 веке удобнее и практичнее применять современные технологии. В настоящее время существуют различные приложения по учету времени работы сотрудников, но у каждого из таких приложений различные функциональные возможности, разный интерфейс.

Наше web-приложение должно облегчить и упростить введение учета рабочего времени сотрудников. web-приложение должно предоставлять основную необходимую функциональность:

- Учёт времени и контроль сотрудников
- Мониторинг времени работы сотрудников над заданиями проекта
- Администрирования работы сотрудников

Данный курсовой проект посвящен разработке именно такого, простого в использовании, но в то же время выполняющего самые необходимые функции, web-приложения, способного прозрачно показывать затраченное время на работу сотрудником.

2 Постановка задачи

Цель курсовой работы: реализовать web-приложение, которое отвечает следующим требованиям:

- стабильная работа на современных веб-браузерах,
- сдержанный внешний вид, выдержанный в едином стиле,
- отсутствие необходимости в справке для возможности

осуществления основных задач:

- 1) авторизация,
- 2) работа с календарем,
- 3) учет времени,
- 4) администрирование проектов и задач,
- 5) редактирование информации в личном кабинете.

Для достижения данной цели были выделены следующие задачи:

- 1) разработка Front-end части сайта,
- 2) разработка Back-end части сайта,
- 3) создание связи между Front-end и Back-end частями сайта,
- 4) разработка базы данных.

3 Анализ предметной области

3.1 Глоссарий

- **front-end** - клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса.
- **back-end** - программно-аппаратная часть сервиса.
- **REST API** — это стиль архитектуры программного обеспечения для построения распределенных масштабируемых веб-сервисов.
- **Микросервисная архитектура** - вариант сервис-ориентированной архитектуры программного обеспечения, направленный на взаимодействие насколько это возможно небольших, слабо связанных и легко изменяемых модулей — *микросервисов* (структурная единица микросерверной архитектуры).
- **GitHub** - крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.
- **Веб-сервис, интернет-сервис, система, веб-приложение, проект** - идентифицируемая уникальным веб-адресом (URL-адресом) программная система со стандартизированными интерфейсами, а также HTML-документ сайта, отображаемый браузером пользователя.
- **Header** - визуальный элемент, расположенный в верхней части страницы.
- **Footer** - визуальный элемент, расположенный в нижней части страницы.
- **Администратор** - человек, имеющий доступ к расширенному функционалу веб-сервиса.
- **Личный кабинет** — это раздел сервиса, в котором Пользователь может получить доступ к персональным данным
- **Контент** - наполнение сайта.
- **Разработчик** — человек или организация, задействованная в разработке ПО.

- **Task Leader** – руководитель задачи
- **Project Leader** – руководитель проекта
- **Проект** — это направленная деятельность, нацеленная на получение определенных результатов.
- **Задание** — это декомпозиция проекта на определённые этапы.
- **Сотрудник** – лицо, обладающее не только определенными правами, но и кругом возложенных на него должностных обязанностей.

3.2 Анализ существующих решений

1. Time Doctor - учёт и контроль рабочего времени команд дистанционных сотрудников. Программа Time Doctor содержит в себе множество функциональных особенностей – от онлайн-мониторинга сотрудников до учета рабочего времени. Интерфейс выглядит несколько устаревшим, но это надежная платформа, которая определенно заслуживает внимания.

Time Doctor рассчитан на широкую пользовательскую аудиторию – будь то удаленные сотрудники (даже один фрилансер) или инженеры, разрабатывающие ПО в крупных компаниях, таких как Apple и Verizon. Дополнительные функции: управление проектами, отслеживание сотрудников.

Достоинства:

- Хорошие функциональные возможности для контроля персонала
- Есть опция для выписки инвойсов в мобильном приложении
- Интеграция со сторонними сервисами, такими как Asana.

Недостатки:

- Устаревший интерфейс. Для функции таймера нужно использовать десктопную версию или расширение в Chrome.
- Из-за неудобного интерфейса и некоторых других недостатков по части юзабилити Time Doctor заметно уступает конкурентам.

2. Kickidler - система по контролю и учету рабочего времени для работников организаций. Сервис акцентирует внимание на наблюдении за

работниками в текущий момент, а также видеофиксации всех нарушений. Использование системы гарантирует, что специалисты будут тратить свое рабочее время на выполнение трудовых обязанностей, а не личных нужд.

Достоинства:

- Онлайн-мониторинг ПК: с его помощью можно увидеть, что делает сотрудник, какие утилиты использует
- Учет рабочего времени: тайм-трекер - контроль рабочих часов, выявление переработок, опозданий, ранних уходов, любых отвлечений персонала от рабочих моментов. Вся статистика отображается в виде диаграмм;
- Анализ эффективности сотрудников: система показывает, какую долю времени специалист посвящает рабочим моментам, а какую - соцсетям, мессенджерам, играм и т.д.;
- Запись видео с монитора: видеофиксация всех действий в рабочее время - настройка сжатия и частоты кадров экономят место на серверах;
- Удаленный доступ: дистанционное управление компьютером пользователя;
- Кейлоггер: запоминание нажатий клавиш сотрудником - сопоставление набора и действий на экране монитора.

Недостатки:

- Возможные проблемы с установкой, но они решаются с помощью эффективной тех поддержкой

3.3 Анализ задачи

3.3.1. Варианты использования приложения

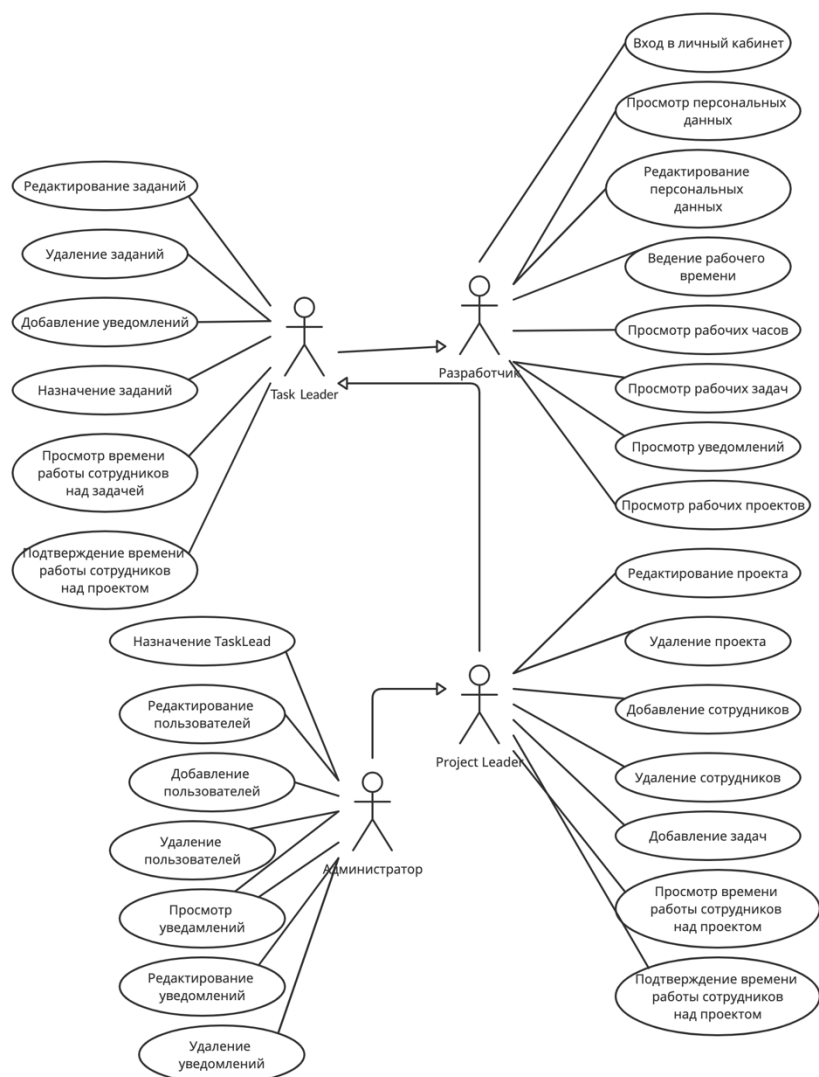


Рисунок 1 - Диаграмма прецедентов

При взаимодействии с web-приложением у пользователя есть определенный список возможностей, зависящий от уровня доступа, который изображен на рисунке 1.

3.3.2. Взаимодействие компонентов системы



Рисунок 2 - Диаграмма последовательностей

На рисунке 2 показана диаграмма последовательности, на которой изображено упорядоченное во времени взаимодействие объектов при авторизации пользователя.

Для авторизации пользователь обращается к форме авторизации, которая передаёт введённые данные на проверку в модуль проверки пользователя. Тот в свою очередь проверяет существование данного пользователя в базе данных и совпадение введённого пароля с паролем, хранящимся в базе данных. Модуль проверки пользователя посылает ответ проверки на форму авторизации, которая выводит пользователю результат авторизации.

3.3.3. Варианты состояния системы

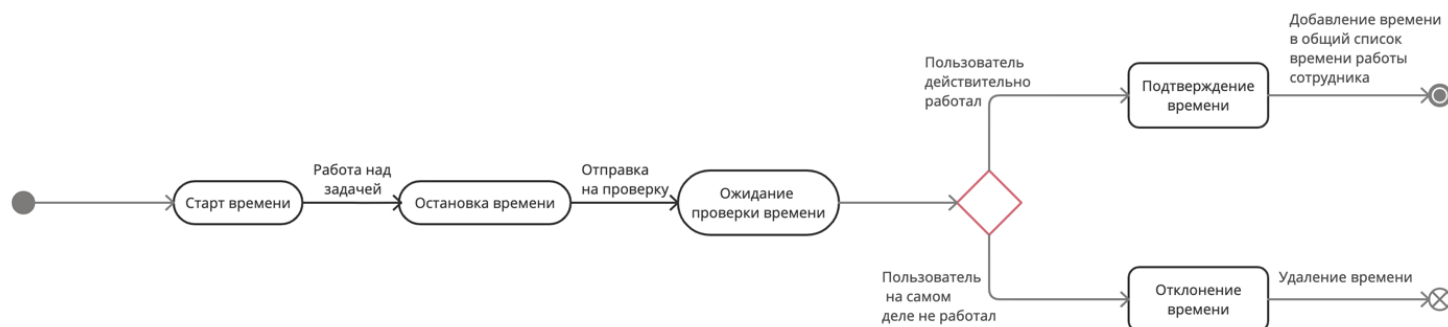


Рисунок 3 - Диаграмма состояний

Изображенная на Рисунке 3, диаграмма состояний, отражает возможные состояния отработанного времени. После старта времени, таймер запускается, когда время останавливается, оно отправляется на проверку и переходит в состояние ожидания проверки. Если время прошло проверку, то оно переходит в состояние подтверждения и добавляется в общий список времени работы сотрудника. Если время не прошло проверку, то оно переходит в состоянии отклонения и удаляется.

3.3.4. Варианты действия в системе

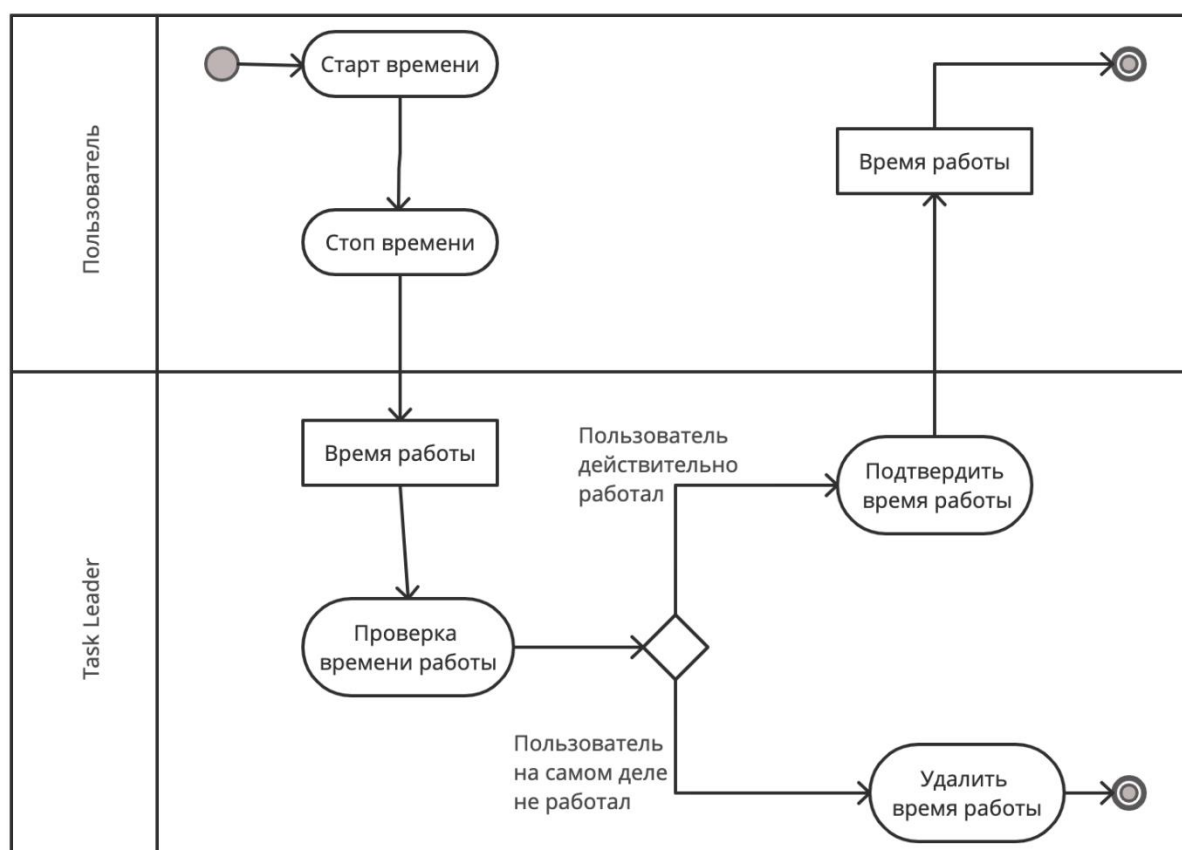


Рисунок 4 - Диаграмма активностей

Диаграмма активности, изображенная на Рисунке 5, отражает действия пользователя и Task Leader. После того, как пользователь заканчивает работу над проектом, время переходит к администратору. При получении времени Task Leader должен ее проверить. Если время не соответствует требованиям, то администратор удаляет ее. Если время соответствует требованиям, то Task Leader подтверждает ее, и время переходит пользователю. Пользователь получает собственное время, и время в свою очередь закрепляется за ним.

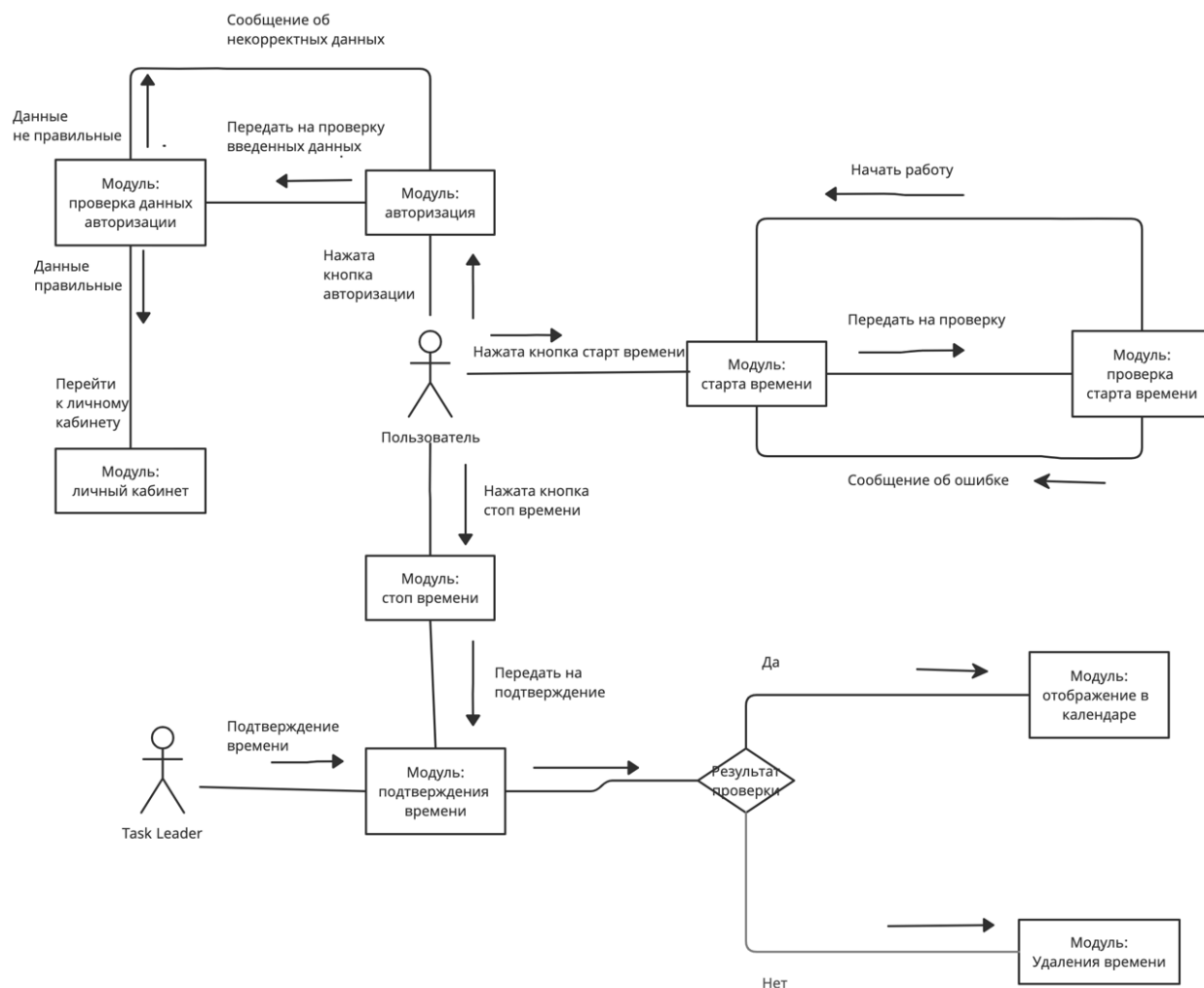


Диаграмма взаимодействий, изображенная на рисунке 5, отражает возможные действия пользователя и системы.

Если пользователь захочет начать работу над проектом, ему надо запустить таймер нажав кнопку старта времени, система проверит возможность старта времени, если это невозможно, то система предупредит его об этом. Если старт времени возможен, то отчет запущен, после нажатия кнопки стоп времени, время попадает на модерацию, где Task Leader одобряет

время. Если время одобрено, то оно заносится в календарь, а если не одобрено удаляется.

3.3.5. Развёртывание приложения

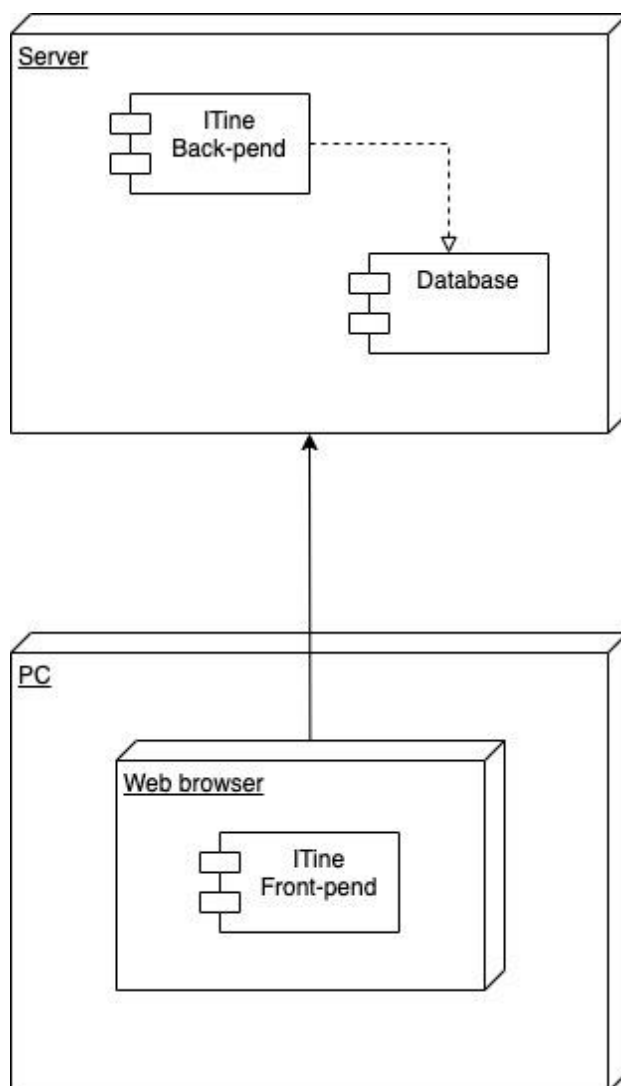


Рисунок 6 - Диаграмма развертывания

На Рисунке 6 представлена диаграмма развертывания, чтобы определить какие аппаратные компоненты («узлы») существуют, какие программные компоненты работают на каждом узле и как различные части этого комплекса соединяются друг с другом. Для разрабатываемого web-приложения узлом устройства является персональный компьютер и сервер, а в качестве узла среды выполнения выступает web-браузер. В браузере развернут front-end приложения, а на серверной части back-end и база данных.

3.3.6. IDEF0



Рисунок 7 – IDEF0

Рассмотрим основной бизнес – процесс на примере контекстной диаграммы, представленной на Рисунок 7 - Контекстная диаграмма. Данная диаграмма представляет собой общее видение процесса работы Сервиса. Работу сервиса регулируют Правила Компании и Пользовательское соглашение, которые определяются совместно с Законами РФ, Уровни доступа к Системе, которые определяются Администратором. Обеспечивают работу Системы Администратор и Исходная БД. На вход в Систему поступает Сотрудник компании, которому надо учитывать рабочее время. На выходе Система выдает Подтвержденное/Неподтвержденное время.

4 Реализация

4.1 Сущности

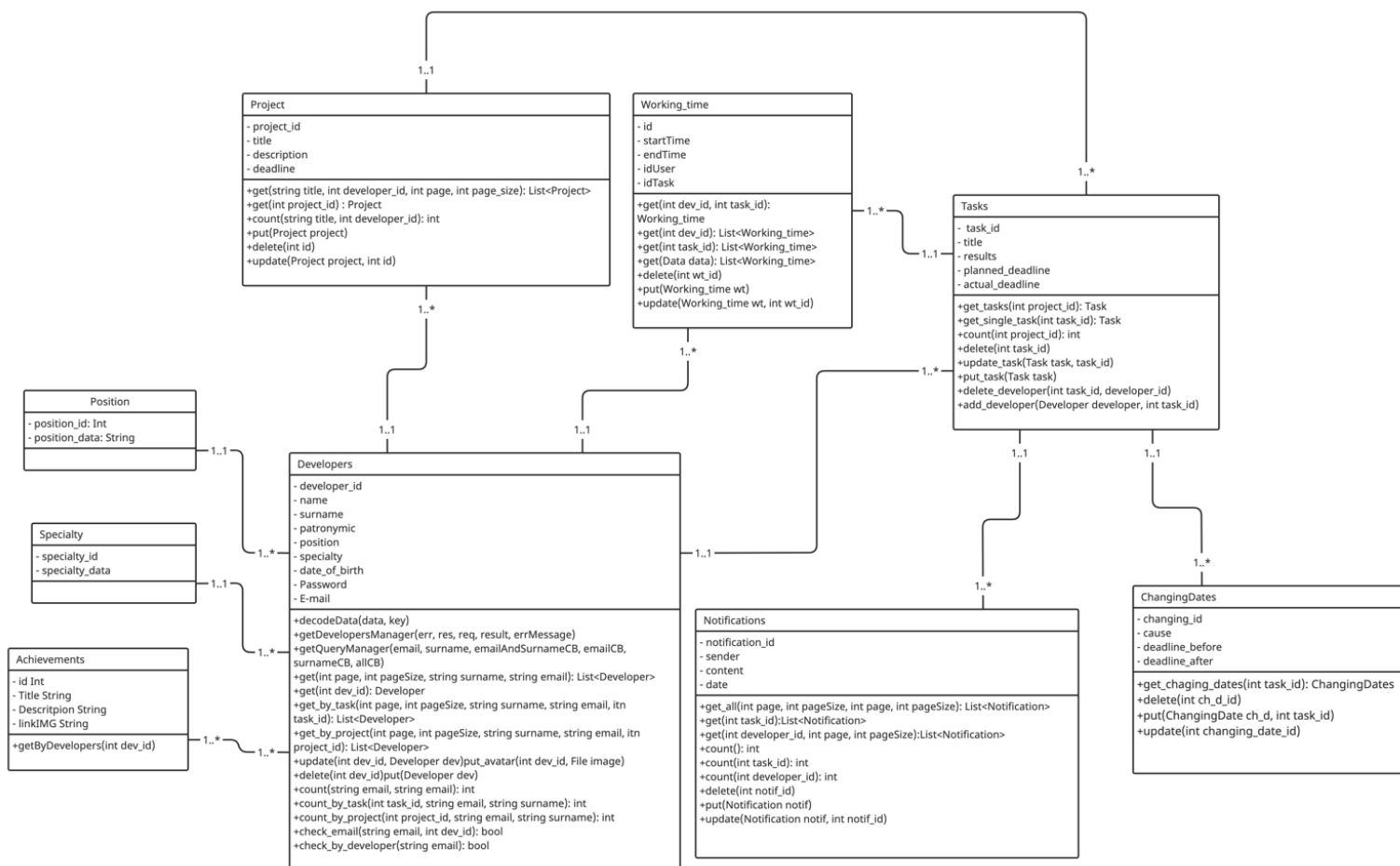


Рисунок 8 - Диаграмма классов

На Рисунке 8 изображена диаграмма классов. Она отображает их отношения.

1. Класс «Speciality» – представляет собой отражение сущности специальности. Класс имеет следующие свойства:

- «specialty_id» – уникальный идентификатор,
- «specialty_data» – наименование специальности.

2. Класс «Position» – отражает информацию о позиции для какого-либо разработчика (junior, middle, senior и т.д.). Класс имеет следующие свойства:

- «position_id» – уникальный идентификатор,

— «position_data» – название/описание позиции.

3. Класс «Working_time» – представляет собой отражение сущности.

Класс имеет следующие свойства:

— «id» – уникальный идентификатор,

— «start time» – начало работы над проектом,

— «end time» – конец работы над проектом,

— «idUser» – уникальный идентификатор developer, который работал

это время,

— «idTask» – уникальный идентификатор task над каким работали это

время.

4. Класс «Tasks» – класс задачи в рамках проекта:

— «task_id» - уникальный идентификатор задачи,

— «title» - название задачи,

— «results» - текстовое описание результатов задачи,

— «planned_deadline» - планируемое время окончания задачи,

— «actual_deadline» - актуальное время окончания задачи,

— «project_id» - уникальный идентификатор проекта, в рамках которого

выполняется данная задача,

— «task_lead_id» - уникальный идентификатор developer, являющегося

ответственным за эту задачу.

5. Класс «Project» – отражает информации о проекте. Класс имеет следующие свойства:

— «project_id» – уникальный идентификатор Project

— «title» – название проекта

— «description» – описание проекта

— «deadline» – дата, к которой должен быть выполнен проект

6. Класс «Developers» – класс разработчика с полями:

— «developer_id» - уникальный идентификатор разработчика,

— «name» - имя разработчика,

- «surname» - фамилия разработчика,
- «patronymic» - отчество / middle name разработчика,
- «position» - уникальный идентификатор, по которому получается информация о позиции разработчика,
- «specialty» - уникальный идентификатор, по которому получается информация о специальности разработчика,
- «date_of_birth» - дата рождения разработчика,
- «password» - хешированное значение пароля для разработчика,
- «email» - почта разработчика, логин по совместительству.

7. Класс «Notifications» – содержит информацию о уведомлениях.

Класс имеет следующие свойства:

- «notification_id» – уникальный идентификатор уведомления,
- «sender» - уникальный идентификатор отправителя уведомления,
- «content» – текстовая информация уведомления,
- «date» – дата отправки уведомления.

8. Класс «Changing_dates» – класс отражает историю изменения актуального срока выполнения задачи. Класс имеет следующие свойства:

- «changing_id» – уникальный идентификатор изменения дедлайна,
- «cause» – причина изменения актуального срока,
- «deadline_before» – дата актуального срока выполнения задачи до изменения,
- «deadline_after» – новая дата дедлайна,
- «task_id» – уникальный идентификатор задачи, у которой изменили актуальный срок.

9. Класс «Achievements» – класс достижений, которые может получать разработчик в течение своей работы с приложением. Класс имеет следующие свойства:

- «achievements_id» - уникальный идентификатор достижения,
- «title» - название достижения,

- «description» - описание достижения. За что было получено, информация о достижении и т.д.,
- «linkImg» - ссылка на изображение достижения.

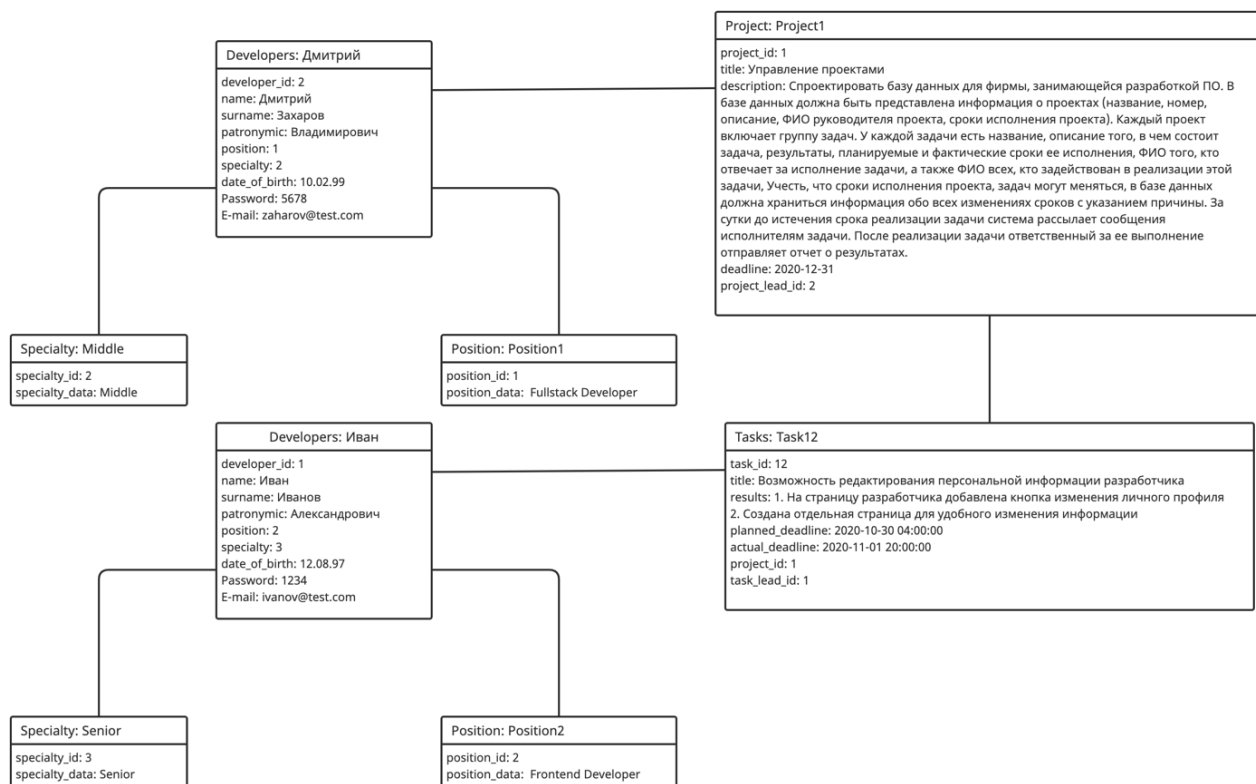


Рисунок 9 - Диаграмма объектов

На Рисунке 9 изображена диаграмма объектов, которая отражает множество экземпляров классов и отношений между ними в некоторый момент времени. На ней изображены два экземпляра класса “Developers”, экземпляр класса “Task”, которая хранит в себе ссылку на “Developers: Иван” и экземпляр класса “Project”, которая хранит в себе ссылку на “Developers: Дмитрий”. В свою очередь экземпляры класса “Developers” хранит в себе ссылку на экземпляр класса “Specialty” и “Position”.

4.2 Сценарии воронок конверсии

1) Посетил главную страницу - Авторизовался - Перешел на страницу управления времени - Нажал кнопку "Добавить".

2) Посетил главную страницу - Авторизовался - Перешел на страницу добавления проекта - Нажал кнопку "Добавить".

3) Посетил главную страницу - Авторизовался – Перешел на страницу разработчиков проекта - Выбрал нужные фильтры - Нажал кнопку "Найти".

4.3 Графический интерфейс

4.3.1. Страница авторизации

Страница авторизации предоставляет пользователю возможность авторизации. На рисунке 10 представлен вид данной страницы:

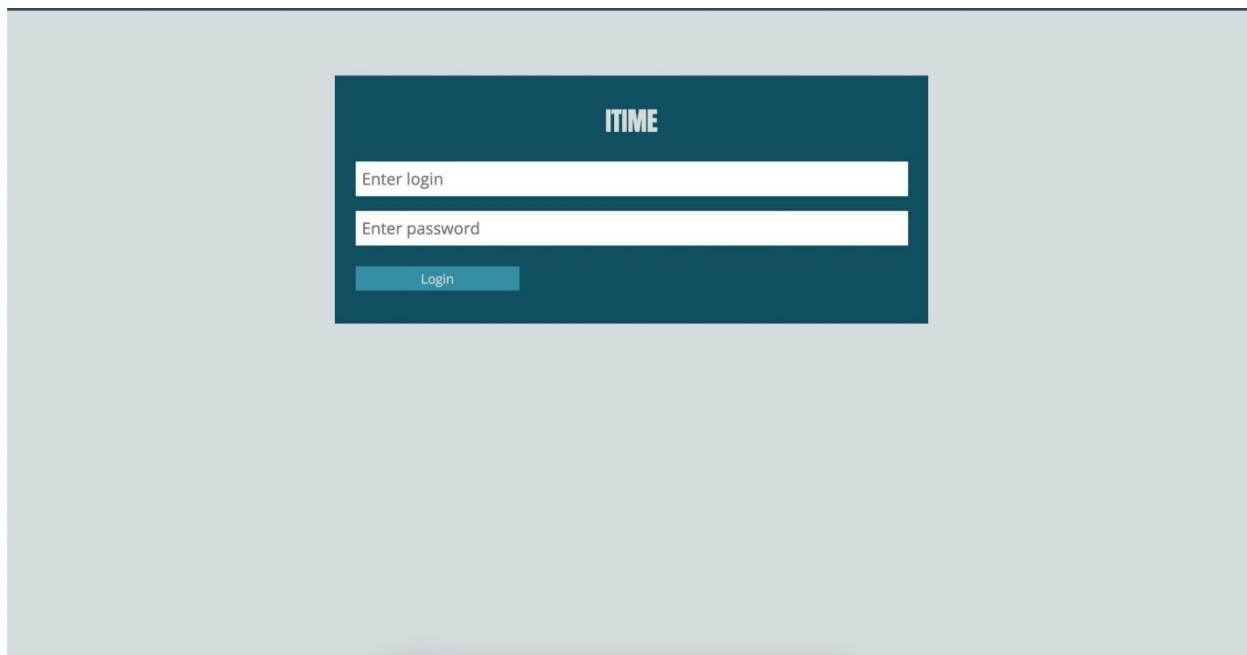


Рисунок 10 – Страница авторизации

4.3.2. Страница профиля

На странице профиля указана информация о сотруднике и его достижениях, а также есть возможность изменить и удалить профиль.

Страница пользователя изображена на рисунке 11:

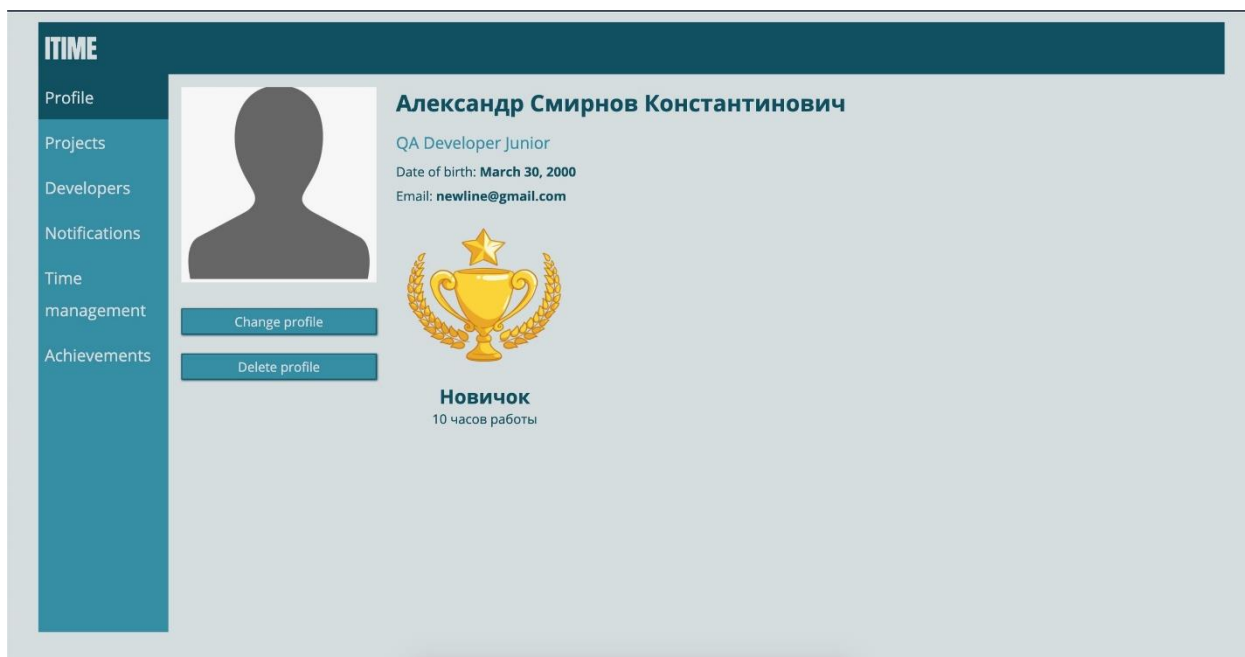


Рисунок 11 – Страница пользователя

4.3.2.1. Страница изменения профиля

На странице изменения профиля пользователю доступна возможность изменять личную информацию.

Страница изменения профиля изображена на рисунке 12:

The screenshot shows the profile editing page. The sidebar is identical to the previous page, with 'Profile' highlighted. The main content area contains a series of input fields for editing the user's information: 'Александр', 'Смирнов', 'Константинович', '30.03.2000' (with a calendar icon), 'newline@gmail.com', 'QA Developer' (dropdown menu), and 'Junior' (dropdown menu). Below these fields is a checkbox labeled 'Is admin:' which is checked. At the bottom are two buttons: 'Save' and 'Back'.

Рисунок 12 – Страница изменения профиля

4.3.3. Страница проектов

На странице проектов отображён список проектов с возможностью поиска, а также кнопкой для добавления проекта, которая доступна только для администратора.

Страница проектов изображена на рисунке 13:

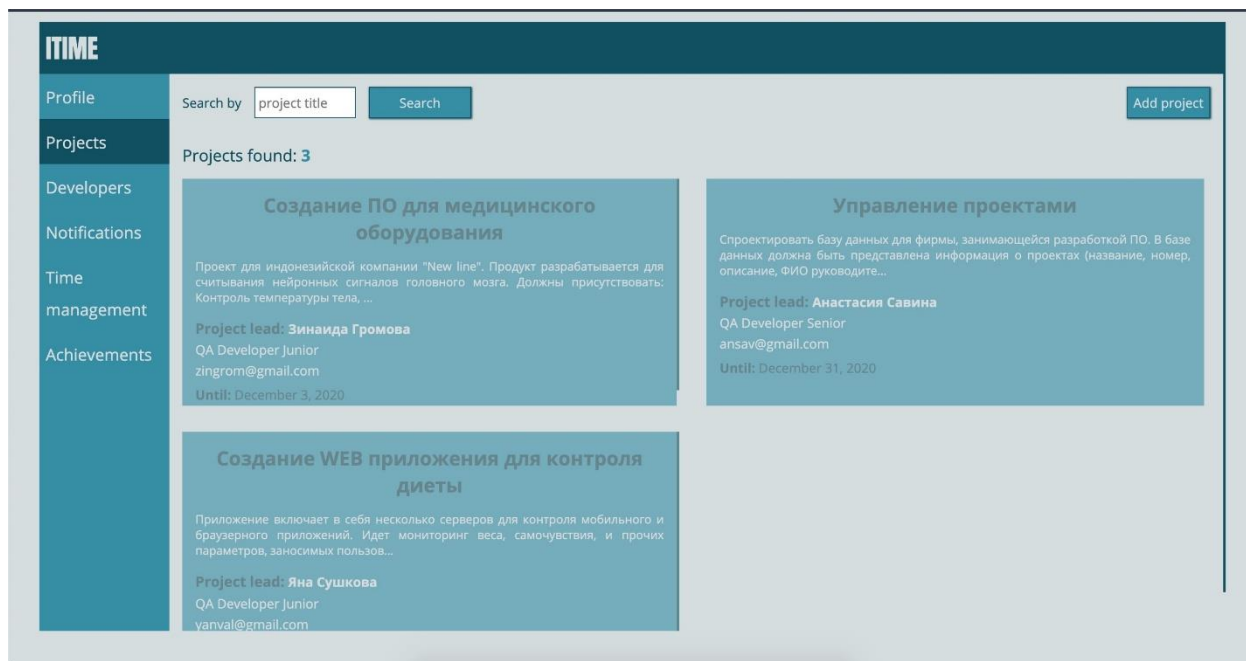


Рисунок 13 – Страница проектов

4.3.3.1. Страница описания проекта

На странице проекта представлена информация о проекте, а также для администратора доступна возможность изменять проект. Есть возможность посмотреть задания по проекту и участвующих в нём разработчиков.

Страница описания проекта изображена на рисунке 14:

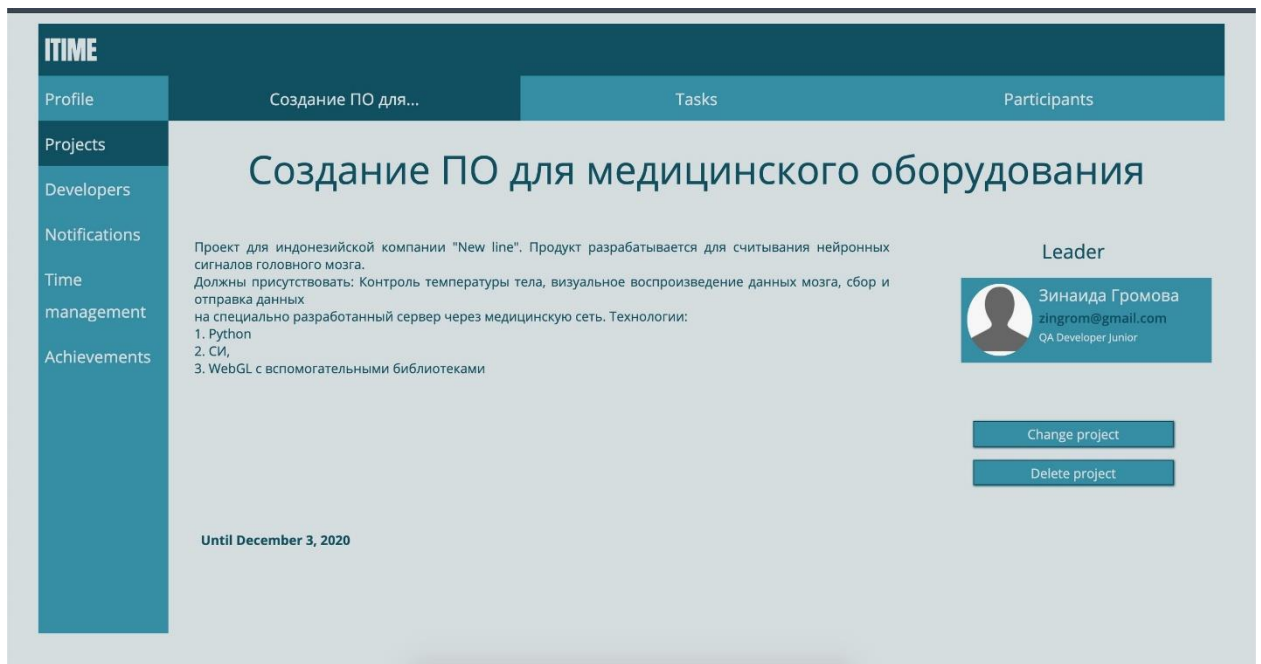


Рисунок 14 – Страница описания проекта

4.3.3.2. Страница заданий проекта

На странице заданий проекта представлен список проектов с возможностью поиска и добавления нового задания.

Страница заданий проекта изображена на рисунке 15:

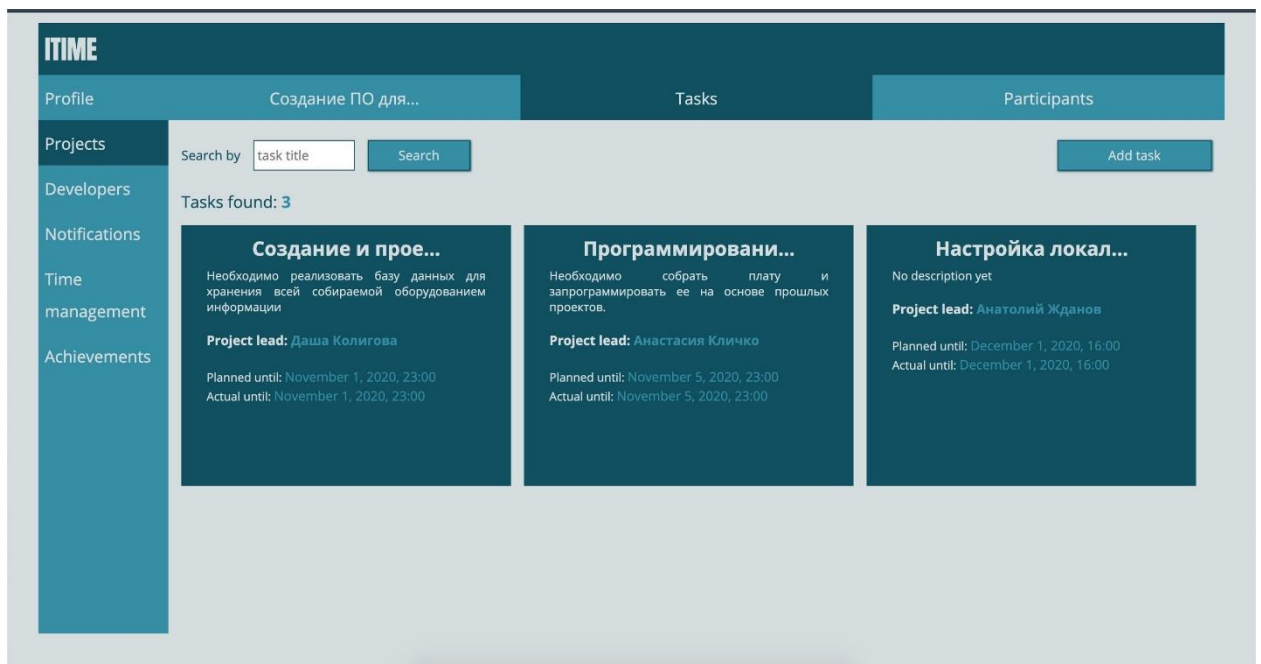


Рисунок 15 – Страница заданий проекта

4.3.3.2.1. Страница описания задания проекта

На странице описания задания проекта представлена информация о задании, с возможностью редактирования и удаления задания.

Страница описания задания проекта изображена на рисунке 16:

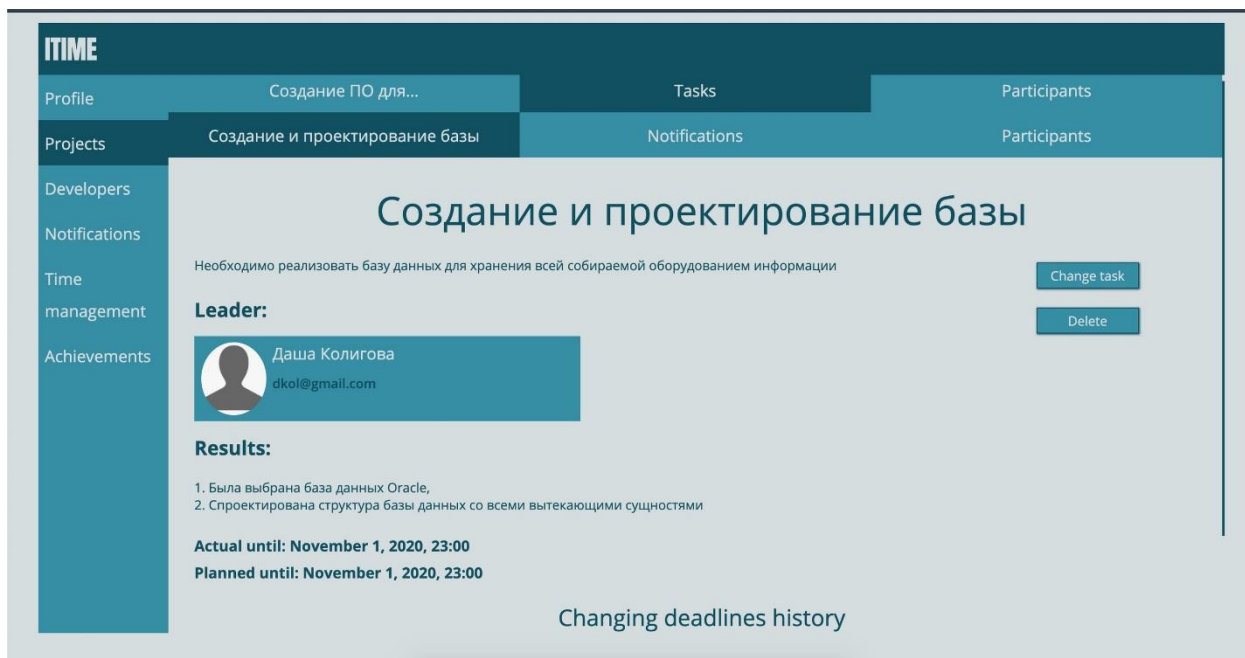


Рисунок 16 – Страница описания задания проекта

4.3.3.2.2. Страница уведомлений задания проекта

На странице уведомлений задания проекта представлен список уведомлений (рисунок 17):

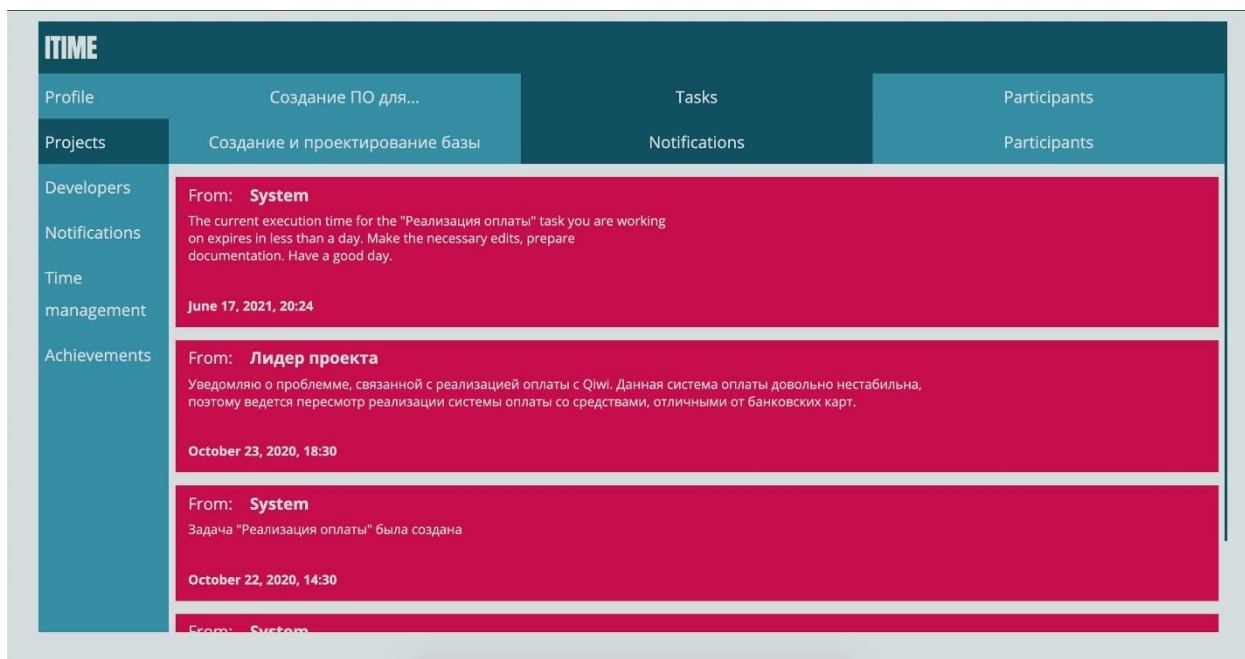


Рисунок 17 – Страница уведомлений задания проекта

4.3.3.2.3. Страница разработчиков над заданием проекта

На странице (рисунок 18) представлены разработчики, работающие над задачей. Есть возможность поиска по email и фамилии.

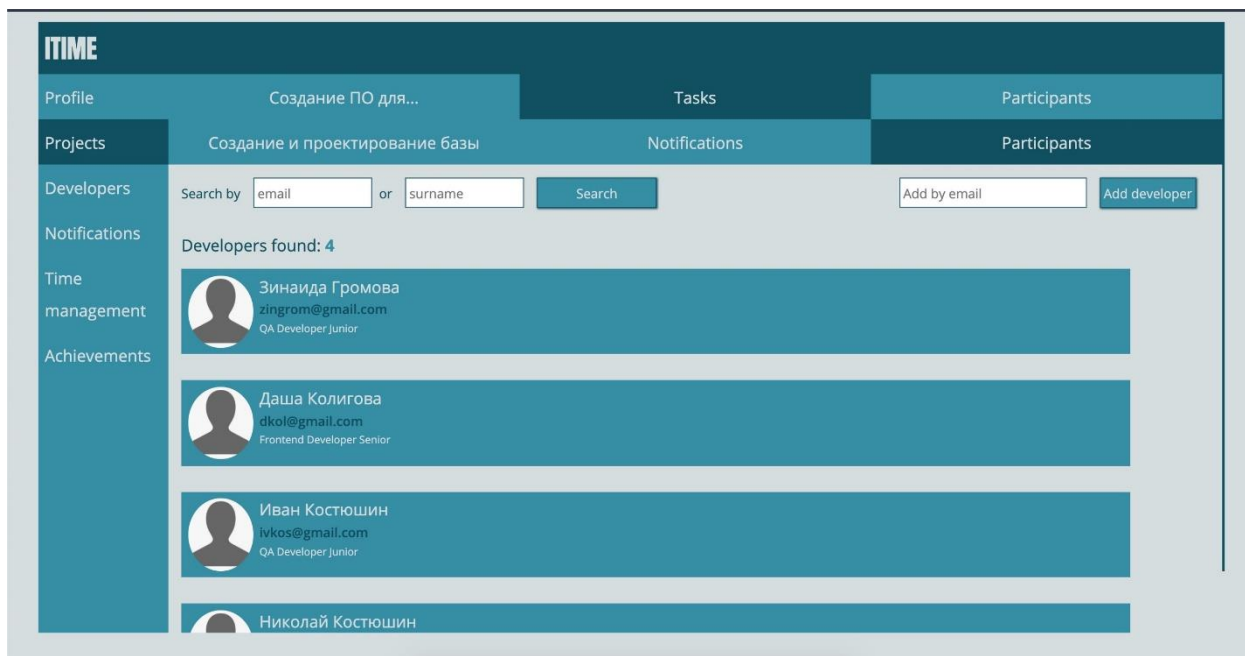


Рисунок 18 – Страница разработчиков над заданием проекта

4.3.3.3. Страница разработчиков проекта

На странице (рисунок 19) представлен список разработчиков, работающих над проектами с возможностью поиска по email и фамилии.

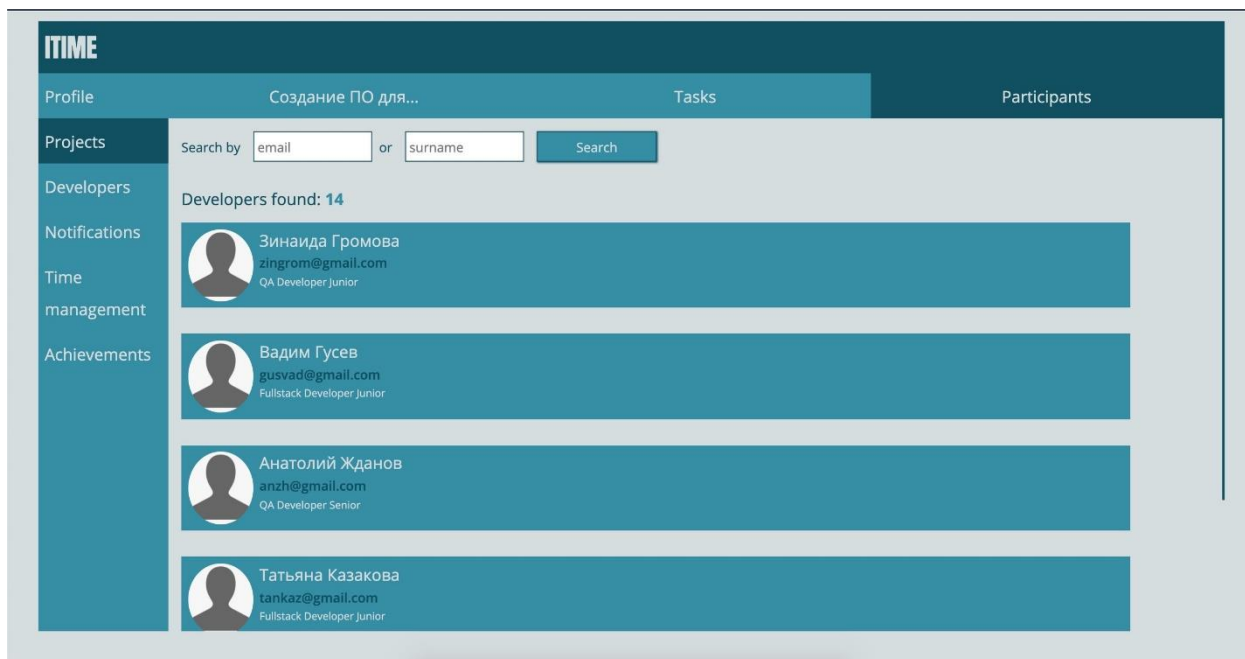


Рисунок 19 – Страница разработчиков проекта

4.3.3.4. Страница добавления проекта

На странице (рисунок 20) представлены поля ввода информации для добавления проекта:

The screenshot shows the 'Add Project' page in the ITIME system. The top navigation bar has 'Profile', 'Создание ПО для...', 'Tasks', and 'Participants'. The left sidebar has 'Profile', 'Projects', 'Developers', 'Notifications', 'Time management', and 'Achievements'. The main content area has a form with the following fields: 'Title' (text input), 'Description of project' (text area), 'Date' (calendar icon, placeholder 'ДД.ММ.ГГГГ'), and 'Project lead's email' (text input). There is a 'Put project' button and a 'Back' link. The URL at the bottom is '35.234.116.28:3000/profile/'.

Рисунок 20 – Страница добавления проекта

4.3.4. Страница разработчиков

На странице (рисунок 21) представлен список разработчиков с возможностью поиска по email и фамилии, а также добавлении нового разработчика.

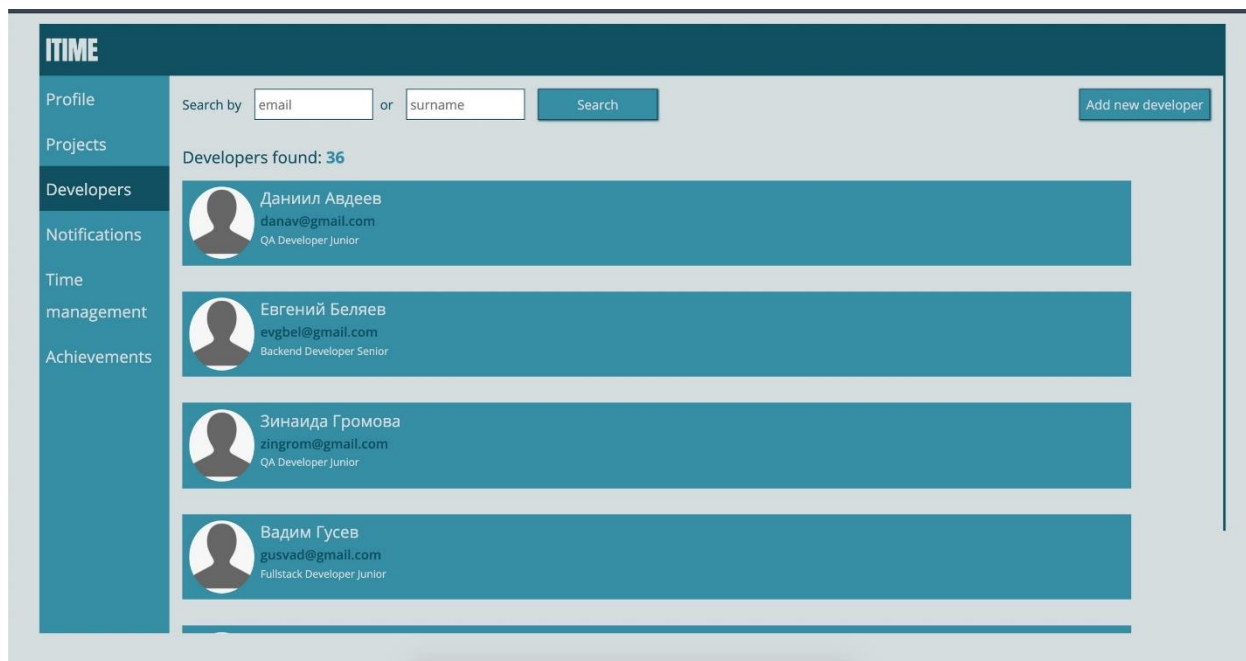


Рисунок 21 – Страница разработчиков

4.3.5. Страница уведомления

На странице (рисунок 22) представлен список уведомлений.

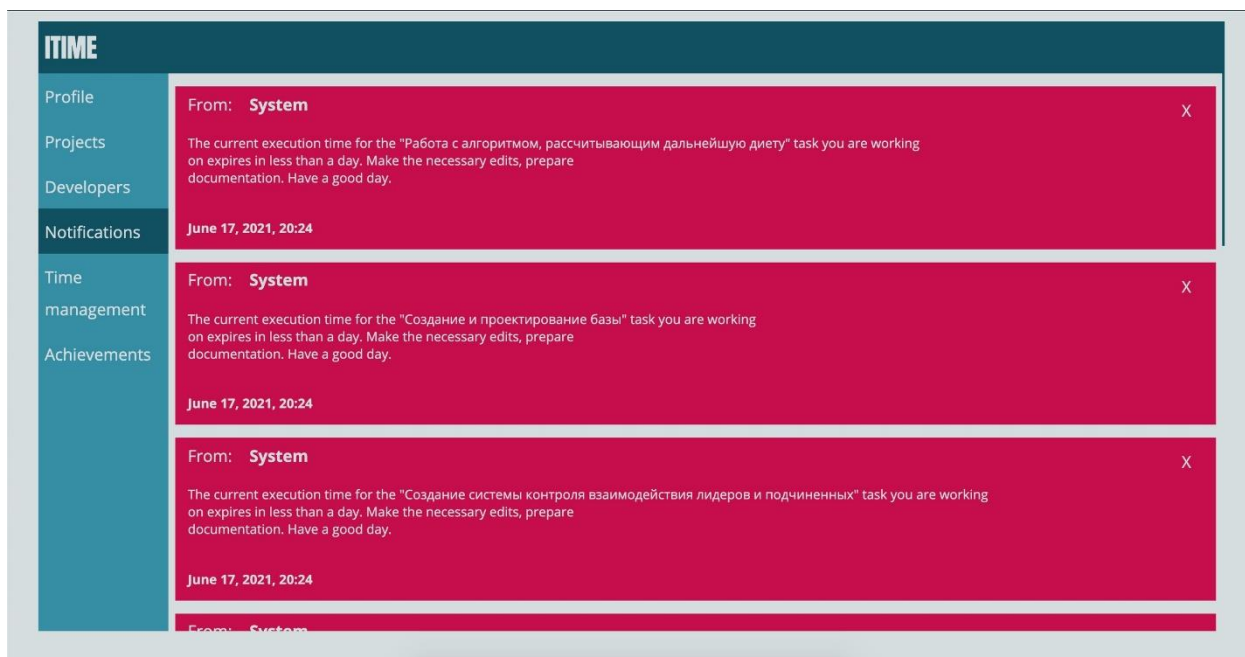


Рисунок 22 – Страница уведомлений

4.3.6. Страница управления временем

На странице (рисунок 23) представлен календарь с возможностью просмотра и добавления отработанного времени.

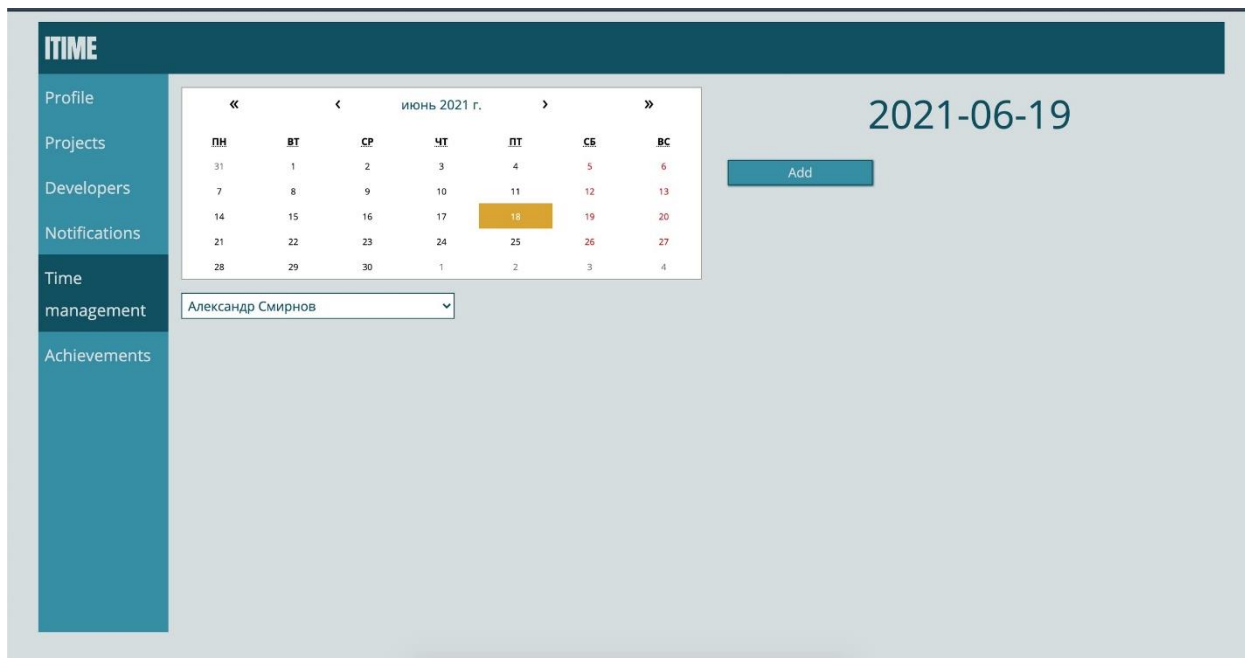


Рисунок 23 – Страница управления временем

4.3.7. Страница достижений

На странице (рисунок 24) представлен список достижений с возможностью редактирования, удаления и добавление.

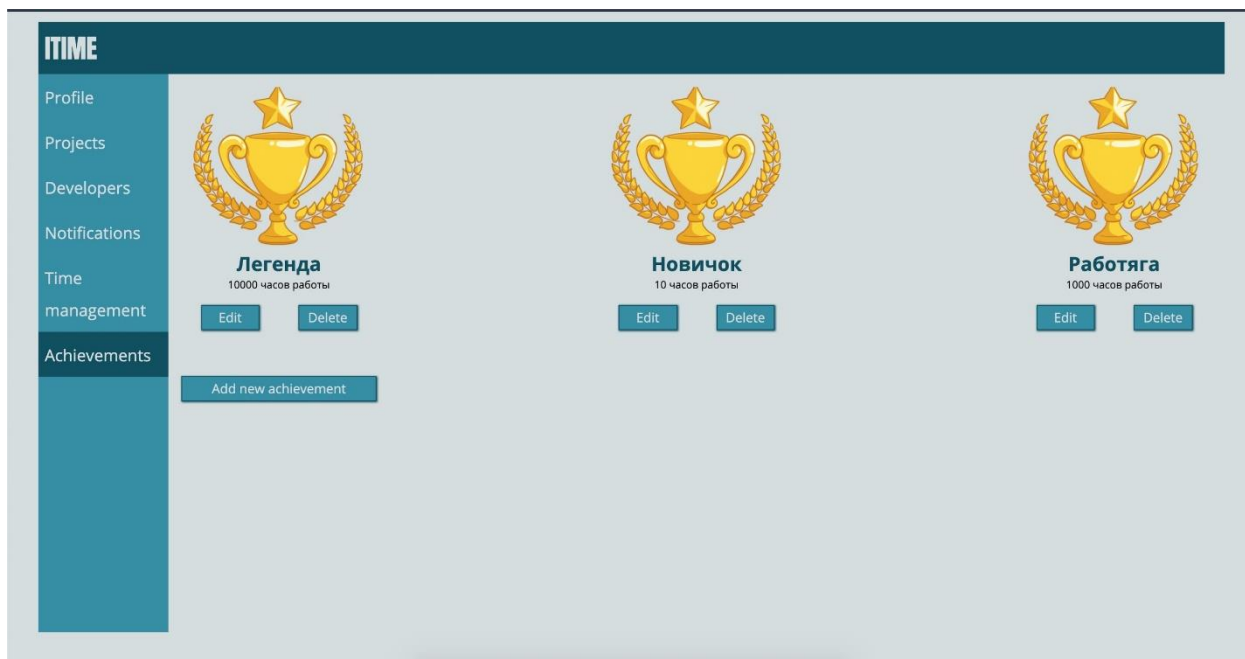


Рисунок 24 – Страница достижений

5 Тестирование

После реализации всех задач, был проведен запланированный набор тестов. Он включает 3 вида тестирования:

- дымовое тестирование,
- UI тестирование,
- юзабилити тесты

5.1 Дымовое тестирование

Для данного тестирования необходимо было проверить работоспособность сайта на следующих основных сценариях:

- создание проекта,
- создание задания проекта,
- удаление задания,
- изменение задания,
- авторизация,
- добавление времени,
- редактирование информации в личном кабинете,
- фильтрация разработчиков проекта.

Дымовое тестирование проводилось ручным способом, в браузере Google Chrome. Результаты, полученные в ходе тестирования представлены в таблице 1.

Таблица 1. Результаты дымового тестирования.

Сценарий	Результат
Создание проекта	Пройден
Создание задания проекта	Пройден
Удаление задания	Пройден
Изменение задания	Пройден

Авторизация	Пройден
Добавление времени	Пройден
Редактирование информации в личном кабинете	Пройден
Фильтрация разработчиков проекта	Пройден

По итогу дымового тестирования было установлено, что сайт проходит все основные утвержденные сценарии.

5.2 UI тесты

В результате UI тестирования было выполнено 27 тестов, охватывающие основные возможности сайта.

Таблица 2. Результаты UI тестирования.

Шаги теста	Ожидаемый результат	Статус
1. Нажимается «Login» на главной странице при введённых данных	Открылась страница пользователя при правильно введённых данных	Пройден
1. Нажимается кнопка «Change profile» на странице пользователя	Открылась страница изменения данных о пользователе	Пройден
1. Нажимается кнопка «Delete profile» на странице пользователя	Всплывает подтверждение об удалении пользователя	Пройден

1. Нажимается кнопка «Profile»	Открылась страница пользователя	Пройден
1. Нажимается кнопка «Projects»	Открылась страница со списком проектов	Пройден
1. Нажимается кнопка «Developers»	Открылась страница со списком проектов	Пройден
1. Нажимается кнопка «Notifications»	Открылась страница со списком проектов	Пройден
1. Нажимается кнопка «Time management»	Открылась страница со списком проектов	Пройден
1. Нажимается кнопка «Achievements»	Открылась страница со списком проектов	Пройден
1. Запускается начальная страница 2. В поля вводятся корректные данные 3. Нажимается кнопка «Login»	1. Открылась страница с авторизацией 2. Открылась страница личного кабинета	Пройден
1. Запускается начальная страница 2. В поля вводятся некорректные данные	1. Открылась страница с авторизацией 2. Появилось сообщение о том, что пользователя с такими данными не существует	Пройден

3. Нажимается кнопка «Login»		
1. Запускается начальная страница 2. Не во все поля вводятся данные 3. Нажимается кнопка «Login»	1. Открылась страница с авторизацией 2. Появилось сообщение о том, что необходимо ввести поле	Пройден
1. Нажимается кнопка «Profile»	Открылась страница личного кабинета	Пройден
1. Нажимается кнопка «Выход»	Открылась страница главной страницы	Пройден
1. Нажимается кнопка «Change profile» 2. В поля вводятся корректные данные 3. Нажимается кнопка «save»	1. Открылась страница с редактированием данных 2. Открылась страница личного кабинета с сохраненными измененными данными	Пройден
1. Нажимается кнопка «Change profile» 2. В одно или несколько полей вводятся некорректные данные или	1. Открылась страница с редактированием данных 2. Появилось сообщение о том, что необходимо ввести поле	Пройден

не во все поля вводятся данные		
3. Нажимается кнопка «save»		

5.3 Юзабилити тестирование

Для проведения юзабилити тестирования было случайно отобрано 3 человека, не пользовавшиеся заранее сайтом. Для данного тестирования необходимо проверить следующие основные сценарии взаимодействия пользователя с сайтом:

- авторизация
- создание проекта,
- создание задания проекта,
- удаление задания,
- изменение проекта,
- добавление времени,
- редактирование информации в личном кабинете,
- проверка получения достижений,
- выход из аккаунта.

Результаты тестирования отображены в таблице 3.

Сценарий	Пользователь 1	Пользователь 2	Пользователь 3
Авторизация	Пройден	Пройден	Пройден
Создание проекта	Пройден	Пройден	Пройден

Создание задания проекта	Пройден	Пройден	Пройден
Удаление задания	Пройден	Пройден	Пройден
Изменение проекта	Пройден	Пройден	Пройден
Добавление времени	Пройден	Пройден	Пройден
Редактирование информации в личном кабинете	Пройден	Пройден	Пройден
Проверка получения достижений	Пройден	Пройден	Пройден
Выход из аккаунта	Пройден	Пройден	Пройден

6 Заключение

В результате работы был разработан сайт для контроля времени работы сотрудников, а также были выполнены следующие задачи:

- 1) разработана Front-end часть сайта,
- 2) разработана Back-end часть сайта,
- 3) создана связь между Front-end и Back-end частями сайта,
- 4) разработана база данных.

Так же приложение отвечает всем заявленным требованиям.