

УДК 004.078

**В.И. Пачинин, И.Н. Коренская, А.А. Харлов**  
**ПРОГРАММНОЕ СРЕДСТВО**  
**КОНТРОЛЯ РАБОЧЕГО ВРЕМЕНИ**

Институт информационных технологий БГУИР

Разработанное программное средство осуществляет контроль рабочего времени, необходимость в котором возникает при усовершенствовании процесса распределения задач, в том числе предотвращении ситуаций его переработки и недоработки. В данном докладе рассматривается информационная модель базы данных, архитектура программного средства контроля рабочего времени, а так же описывается алгоритм авторизации.

*Ключевые слова:* программное средство, мониторинг, рабочее время, модель, фреймворк, токен, алгоритм, разработка, API, JSON.

Основой для реализации задач мониторинга рабочего времени является необходимость контроля отработанного рабочего времени, осуществляемая как со стороны работодателей, так и самими сотрудниками. На организацию производственного процесса существенно влияет дисциплина и мотивация сотрудников, следовательно, контроль персонала необходим для успешного решения производственных задач. Особенно актуальным в настоящее время становится разработка мониторинга для работников с гибким графиком. Информация об использовании системы учета рабочего времени положительно влияет на дисциплину коллектива, продуктивность его работы и безопасность.

Одной из главных задач разработанного программного средства (ПС) контроля рабочего времени является разработка информационной модели базы данных (БД), в которой содержатся сведения о количестве часов, затраченных на выполняемую задачу, а также представление отчетов в удобном для пользователя виде.

При проектировании и разработке ПС использовались следующие принципы:

- исключение дублирования информации и повышение ее достоверности за счет отождествления ранее введенной информации;
- расширяемость/масштабируемость – обеспечение возможности добавления, удаления или редактирования функций при неизменных остальных функциональных частях;
- иерархичность, реализуемая формированием многоуровневой структуры системы с учетом делегирования полномочий для соответствующего уровня управления;
- стандартизируемость – проектирование и разработка на основе согласованных международных стандартов и предложений, осуществление реализации открытости на базе функциональных стандартов (профилей) в области информационных технологий.

Для удобства использования ПС созданы мобильный и веб-клиенты. Во избежание дублирования кода, непременно приводящего к ошибкам в системе, вся бизнес-логика реализуется с помощью REST API, с которым работают оба клиента.

ПС контроля рабочего времени включает такие компоненты как: БД, веб-сервер, веб-клиент, мобильный клиент.

Архитектура программного средства представлена на рис 1.

Обмен данными между клиентом и сервером осуществляется по протоколу HTTPS с использованием сообщений в формате JSON.

В качестве СУБД для ПС была выбрана система управления реляционными БД, разработанная корпорацией Microsoft – Microsoft SQL Server. Основным используемым языком запросов является Transact-SQL, который создан совместно Microsoft и Sybase. Использование данной СУБД обусловлено её надежностью, целостностью, автоматическими обновлениями и производительностью.

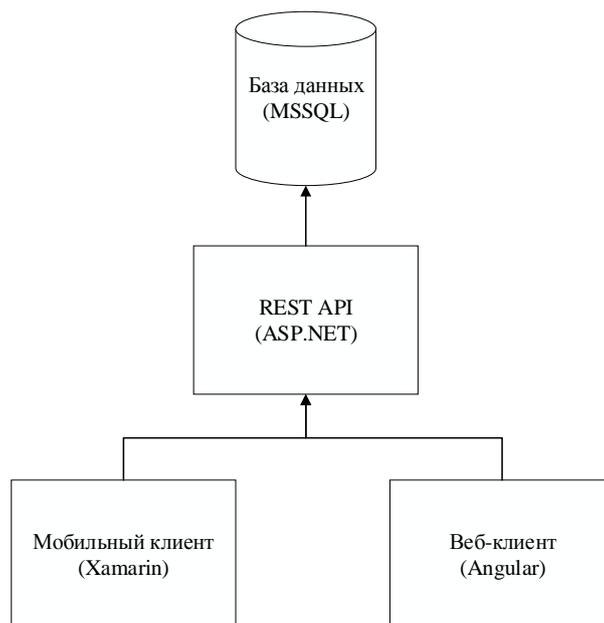


Рис. 1 – Архитектура программного средства

Веб-сервер ПК разработан с применением архитектурного стиля REST, который представляет собой согласованный набор ограничений, учитываемых при проектировании распределённой системы. Создание серверной части проводилось на платформе ASP.NET, которая является составной частью платформы Microsoft .NET и развитием более старой технологии Microsoft ASP. [1]

Для написания клиентской мобильной части был использован фреймворк для кроссплатформенной разработки мобильных приложений Xamarin с использованием языка C#. Он предоставляет возможность применения любого языка программирования с полным доступом ко всем возможностям SDK платформы и механизму создания UI, получая на выходе приложение, которое имеет высокую производительность. [2]

Для реализации веб-клиента был выбран подход SPA (Single Page Application – одностраничное веб-приложение) и один из лидирующих фреймворков Angular. Среди его преимуществ можно выделить такие, как: декларативный стиль кода, высокая скорость разработки, модульность, связывание данных, развитое сообщество.

Информационная модель базы данных представлена на рис. 2.

Приведем описание таблиц информационной модели БД:

- Users – хранит в себе данные о зарегистрированных пользователях ПК, имеет связи с таблицами Roles для определения роли пользователя, Tokens для определения доступных токенов для авторизации, TimerEntries;
- Roles – содержит в себе доступные роли пользователей;
- Tokens – применяется для реализации авторизации пользователей на сервере, хранит токены, которые были созданы во время логина;
- TimerEntries – служит для хранения информации об отработанных часах, включая время начала и окончания работы таймера и краткое описание проделанной работы, имеет связь с таблицей Projects для возможности связать часы с проектом;
- Projects – список проектов компании, доступный для выбора во время начала работы таймера, возможно установление связи проекта с клиентом (таблица Clients);
- Clients – хранит данные клиентов компании;
- ProjectMembers – применяется для управления участниками проекта.

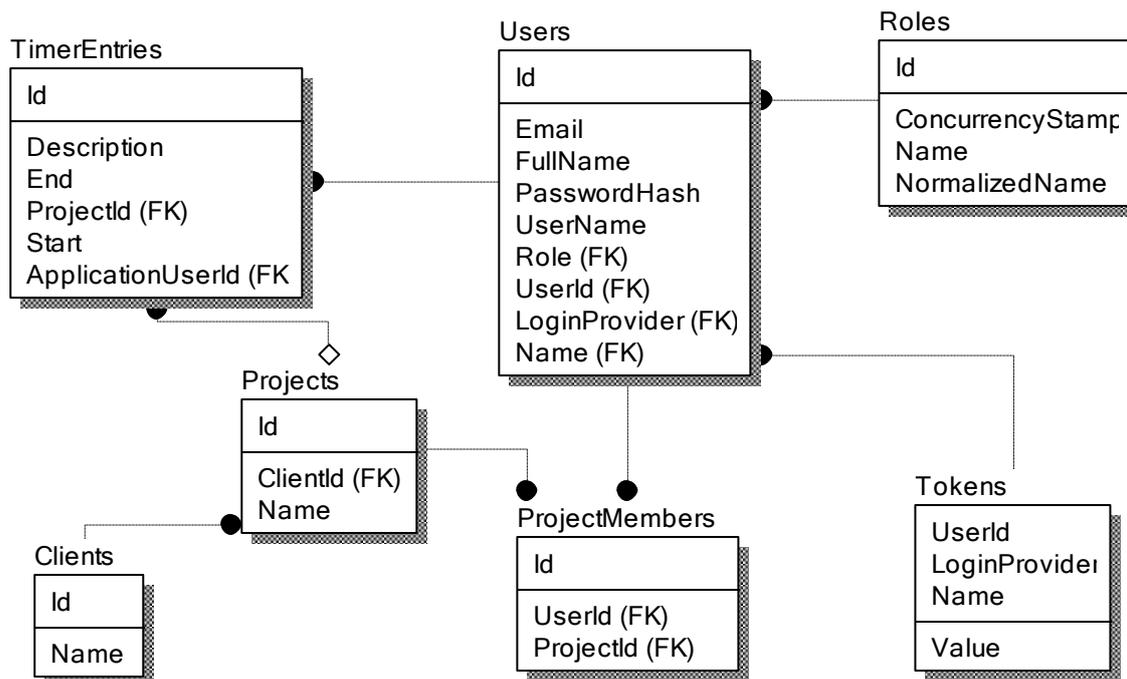


Рис. 2 – Информационная модель базы данных

Рассмотрим одно из наиболее важных требований безопасности при работе с ПС, а именно, доступ к некоторым частям системы, который должен быть осуществлен только после аутентификации пользователя. Для авторизации пользователей используется JWT: JSON Web Token (JWT) - открытый стандарт (RFC 7519) для создания токенов доступа, основанный на JSON формате. Он используется для передачи данных авторизации в клиент-серверных приложениях. Токены создаются сервером, подписываются секретным ключом и передаются клиенту, который в дальнейшем использует данный токен для подтверждения своей личности.

Перечислим преимущества JWT над cookies:

- При использовании cookies сервер хранит информацию о выданных сессиях, а применение JWT не требует хранения дополнительных данных о выданных токенах: только проверяется подпись.
- Сервер может не заниматься созданием токенов, а предоставить это внешним сервисам.
- В JSON токенах можно хранить дополнительные сведения о пользователях. Как следствие – более высокая производительность. В случае с cookies иногда требуется осуществлять запросы для получения дополнительной информации. При использовании JWT данная информация может быть передана в самом токене.
- JWT делает возможным предоставление одновременного доступа к различным доменам и сервисам.

Для каждого действия в контроллерах назначаются группы, которым разрешен просмотр содержимого данной страницы. При попытке перехода на страницу сайта, подсистема безопасности производит проверку наличия доступа текущего пользователя. Данная проверка называется авторизацией. Если разрешение доступа не получено, то пользователь должен пройти процесс аутентификации при помощи своих данных входа.

Схема алгоритма аутентификации и авторизации пользователей представлена на рис. 3.

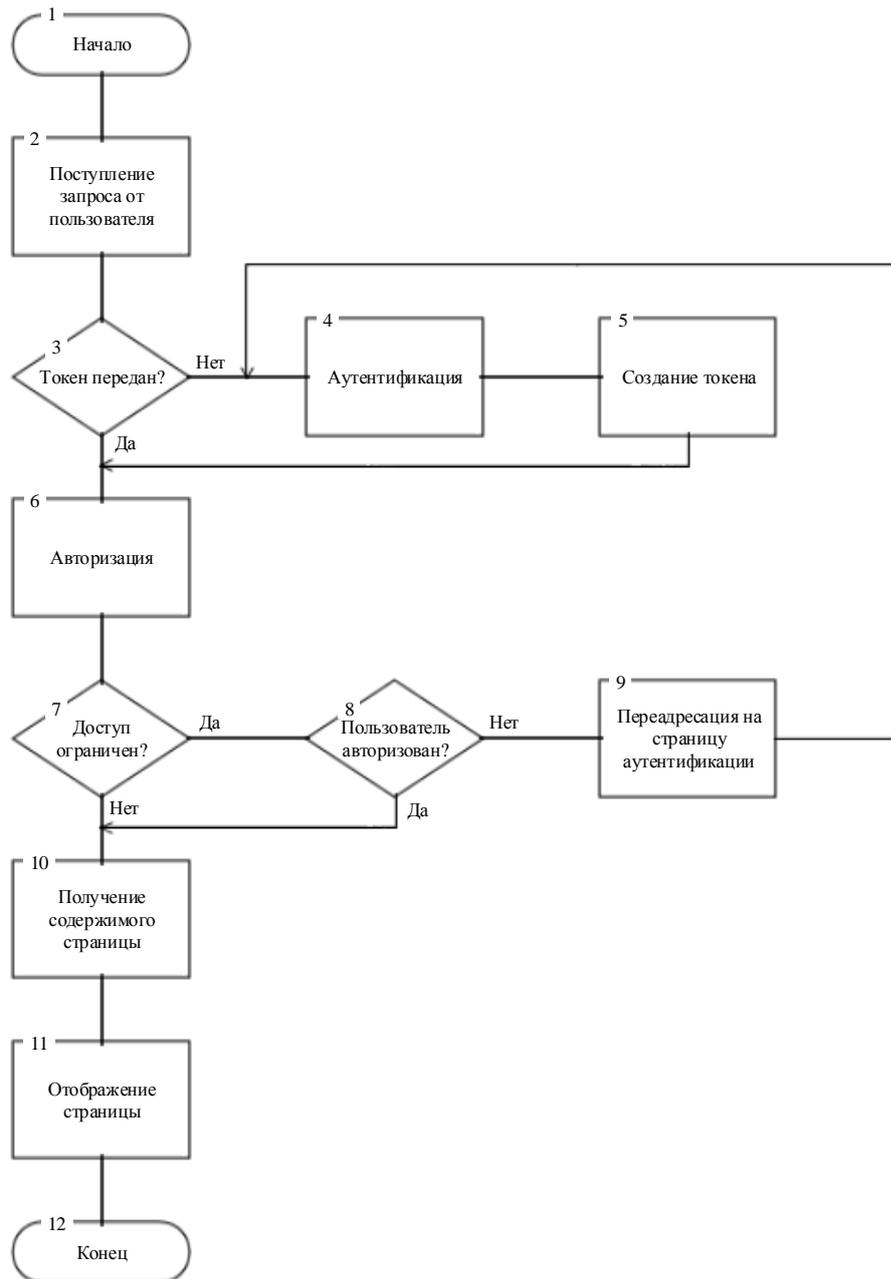


Рис. 3 – Схема алгоритма аутентификации и авторизации пользователей

По умолчанию анонимные пользователи могут обращаться к главной странице. При запросе веб-страницы, к которой закрыт анонимный доступ, выполняются следующие действия:

- Запрос отправляется веб-серверу. Так как идентичность пользователя в данный момент времени отсутствует, то предлагается ввести логин и пароль для входа, используя специальную веб-страницу.
- Пользователь предоставляет свое удостоверение, которое верифицируется приложением.
- Если удостоверение пользователя подтверждается, ему предоставляется доступ к веб-странице. Если же оно оценивается как нелегитимное, то пользователю предлагается ввести логин и пароль.

Разработанное ПС выполняет следующие функции:

- добавление/изменение/удаление клиентов компании;
- создание и редактирование записей таймера;

**СЕКЦИЯ 5.1 ТЕХНИЧЕСКАЯ КИБЕРНЕТИКА (СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ)**

- отображение статистики пользователя с возможностью сравнения с предыдущим периодом;
- добавление/изменение/удаление проектов;
- просмотр и экспорт отчетов;
- отображение истории таймера в мобильном приложении;
- создание новых записей таймера в мобильном приложении.

Использование результатов разработки позволяет выявить грубые нарушения рабочего режима, доработать процесс планирования и распределения задач, повысить эффективность работы коллектива и дисциплинировать сотрудников.

**Библиографический список**

1. Nagel, C. Professional C# 6 and .NET Core 1.0 / C. Nagel – Wrox – 2016. – 1536 pp.
2. Xamarin [Электронный ресурс]. – Режим доступа: <https://developer.xamarin.com/>. – Дата доступа: 18.12.2017.

**V.I. Pachynin, I.N. Korenskaya, A.A. Kharlov**

**TIME TRACKING SOFTWARE**

Institute of information technologies BSUIR

The developed software allows controlling of the working time for both managers and employees. The control of working hours is necessary to prevent overtimes and shortcomings, allows to improve the process of task distribution. Describes the algorithm of authorization, software architecture and information model of the database.

*Keywords:* software, tracking, working time, model, framework, token, algorithm, development, API, JSON.