МИНИСТЕРСТВО ОБРАЗОВАНИЯ ОМСКОЙ ОБЛАСТИ

бюджетное профессиональное образовательное учреждение Омской области

«ОМСКИЙ АВТОТРАНСПОРТНЫЙ КОЛЛЕДЖ»

(БПОУ «Омский АТК»)

**Специальность: 09.02.03 «Программирование в компьютерных системах»**

**КУРСОВОЙ ПРОЕКТ**

**по МДК.01.01 Системное программирование**

**Тема: *Проектирование АИС контроля персонала***

Выполнил студент группы ПКС351

Блащенко И. С.

Проверил преподаватель

Курчевский В. Е.

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Омск 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ ОМСКОЙ ОБЛАСТИ

бюджетное профессиональное образовательное учреждение Омской области

«ОМСКИЙ АВТОТРАНСПОРТНЫЙ КОЛЛЕДЖ»

(БПОУ «Омский АТК»)

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ |
|  | Зам. директора |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.В. Сидоренко |
|  | «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ год |
|  |  |

##### **ЗАДАНИЕ**

**на курсовой проект**

**по МДК.01.01 Системное программирование**

студенту Блащенко Ивану Сергеевичу

группы ПКС351 курса 4

специальность 09.02.03 «Программирование в компьютерных системах»

**Тема: *Проектирование АИС контроля персонала***

**ВВЕДЕНИЕ**

Актуальность заданной темы, цели и задачи работы.

**1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ**

1.1 Основные понятия

1.2 Анализ существующих аналогов

1.3 Техническое задание

**2 ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ**

2.1 Назначение разработки

2.2 Требования к проекту

2.3 Выбор программных и технических средств

2.4 Проектирование системы

**3 ОРГАНИЗАЦИОННЫЙ РАЗДЕЛ**

3.1 Организация графика разработки

3.2 Система контроля версий

**ЗАКЛЮЧЕНИЕ**

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

Дата выдачи задания на курсовой проект «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

Срок выполнения курсового проекта «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

**Руководитель курсового проекта** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Задание рассмотрено и одобрено на заседании ЦМК «Информационные системы и программирование» от «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

Протокол №\_\_\_\_\_\_ Председатель ЦМК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Харченко В.М./

Задание получил «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г.

**Студент**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc5885036)

[1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ 5](#_Toc5885037)

[1.1 Основные понятия 5](#_Toc5885038)

[1.2 Анализ существующих аналогов 6](#_Toc5885039)

[1.3 Техническое задание 8](#_Toc5885040)

[2 ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ 9](#_Toc5885041)

[2.1 Назначение разработки 9](#_Toc5885042)

[2.2 Требования к проекту 9](#_Toc5885043)

[2.3 Выбор программных и технических средств 11](#_Toc5885044)

[2.4 Проектирование системы 16](#_Toc5885045)

[3 ОРГАНИЗАЦИОННЫЙ РАЗДЕЛ 20](#_Toc5885046)

[3.1 Организация графика разработки 21](#_Toc5885047)

[3.2 Система контроля версий 27](#_Toc5885048)

[ЗАКЛЮЧЕНИЕ 30](#_Toc5885049)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 31](#_Toc5885050)

**ВВЕДЕНИЕ**

На сегодняшний день для эффективной работы компании, нужно уделить особое внимание контролю персонала. Функция контроля является одной из важнейших функций управления и поэтому она должна реализовываться уже с момента создания организации. Данные системы устанавливаются не только в крупных компаниях, но также и в небольших офисах. Данные системы позволяют фиксировать время прибытия и ухода сотрудников, время отдыха и установить, чем занят персонал в рабочее время. И тем самым, с помощью использования данной системы снижается риск некачественной работы сотрудниками тех или иных компаний. По этой причине компания может понести вполне конкретные экономические потери. И если руководитель не может знать, как работают его сотрудники, то его бизнес не может быть эффективным и продуктивным. В такой ситуации учет рабочего времени и контроль сотрудников – мера необходимая. Поэтому на сегодняшний день все больше компаний заостряют свое внимание на системах контроля персонала.

Цель проекта: проектирование автоматизированной системы для контроля за персоналом.

Задачи, которые требует:

1. Проанализировать информацию по данной теме,
2. Сформулировать требования к системе,
3. Спроектировать базу данных для системы,
4. Спроектировать визуальную часть системы,
5. Спроектировать программную составляющую приложения.
6. **ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ**
   1. **Основные понятия**

С# – язык программирования, сочетающий объектно-ориентированные и контекстно-ориентированные концепции. Разработан в 1998—2001 годах группой инженеров под руководством Андерсa Хейлсбергa в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET. Компилятор с C# входит в стандартную установку самой .NET, поэтому программы на нём можно создавать и компилировать даже без инструментальных средств вроде Visual Studio.

SQL – язык управления базами данных для реляционных баз данных. Сам по себе SQL не является Тьюринг-полным языком программирования, но его стандарт позволяет создавать для него процедурные расширения, которые расширяют его функциональность до полноценного языка программирования.

Язык был создан в 1970х годах под названием “SEQUEL” для системы управления базами данных (СУБД) System R. Позднее он был переименован в “SQL” во избежание конфликта торговых марок. В 1979 году SQL был впервые опубликован в виде коммерческого продукта Oracle V2.

Первый официальный стандарт языка был принят ANSI в 1986 году и ISO — в 1987. С тех пор были созданы еще несколько версий стандарта, некоторые из них повторяли предыдущие с незначительными вариациями, другие принимали новые существенные черты.

* 1. **Анализ существующих аналогов**
     1. Система учета рабочего времени Bitcop

Система bitcop (рис. 1.1) в большей степени направлена на учет рабочего времени персонала и имеет следующие возможности:

1. Учет рабочего времени,
2. Скрытый режим работы,
3. Скриншоты экранов,
4. Мониторинг сайтов/программ.

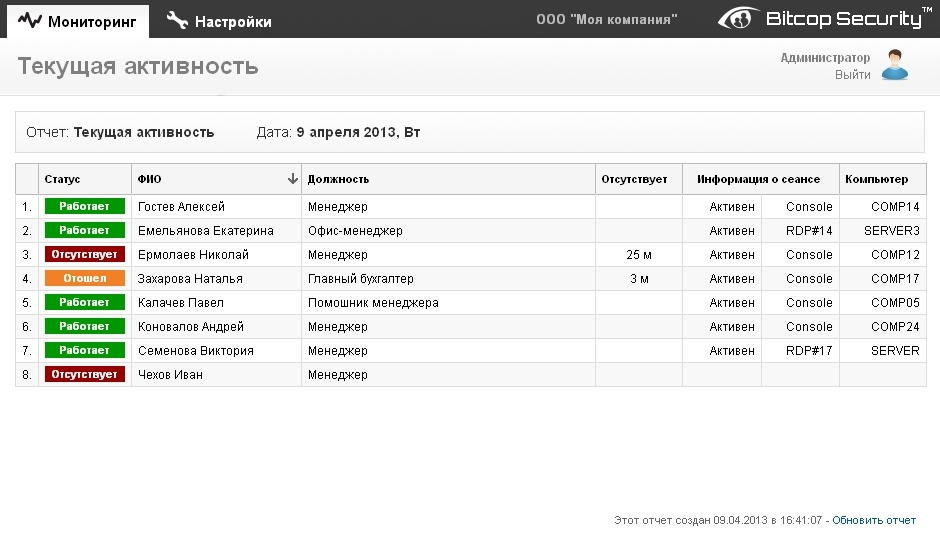
****

Рисунок 1.1 – Внешний вид приложения bitcop

* + 1. Система учета рабочего времени crocotime

Система crocotime (рис. 1.2), направленная на автоматический мониторинг рабочего времени сотрудников, обладает следующими возможностями:

1. Мониторинг работы за компьютером,
2. Скриншот рабочего стола,
3. Автоматические табели,
4. Фиксация нарушений.

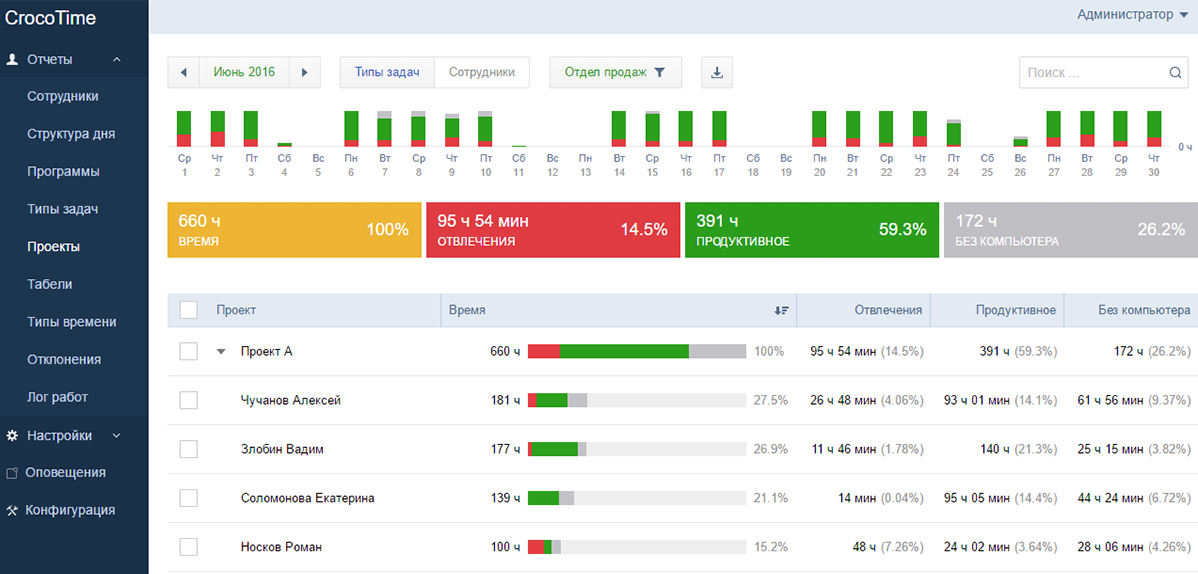
****

Рисунок 1.2 – Внешний вид приложения crocotime

* + 1. Система слежения NeoSpy

Система NeoSpy (рис. 1.3) ориентирована на слежение за сотрудниками и обладает следующими возможностями:

1. Создание скриншотов экрана,
2. Контроль запускаемых приложений,
3. Контроль нажатий,
4. Контроль рабочего времени.

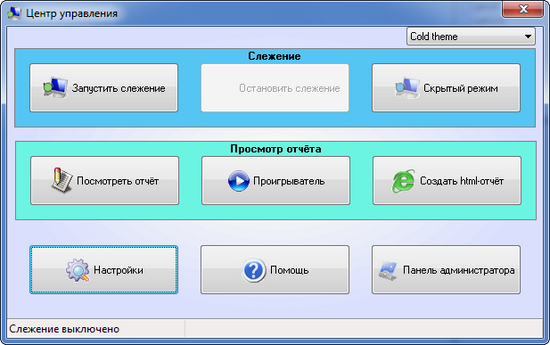


Рисунок 1.3 – Внешний вид системы NeoSpy

* 1. **Техническое задание**

В итоге, проанализировав программы – аналоги и систематизировав все возможности данных систем, можно выделить следующее, что должна уметь система:

1. Мониторинг работы за компьютером,
2. Учет времени работы/паузы,
3. Контроль запускаемых приложений,
4. Скриншоты экранов,
5. Скрытый режим работы (работа в «трее»),
6. Автоматические табели.
7. **ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ**

**2.1 Назначение разработки**

Цель проекта: проектирование автоматизированной системы для контроля за персоналом и записи на смены.

Задачи, которые требует:

1. Проанализировать информацию по данной теме,
2. Предоставить требования к системе,
3. Спроектировать базу данных для системы,
4. Спроектировать визуальную часть системы,
5. Спроектировать программную составляющую приложения.

**2.2 Требования к проекту**

2.2.1 Требования к графическому интерфейсу

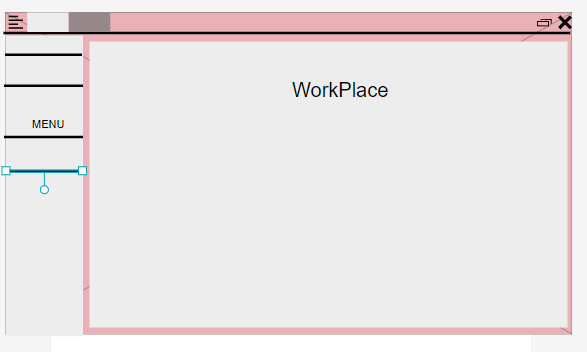


Рисунок 2.1 – Внешний вид моего приложения

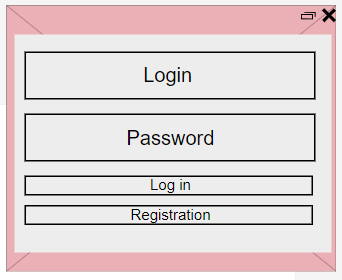


Рисунок 2.2 – Окно входа

2.2.2 Требования к функциональным характеристикам

Система должна иметь:

1. Систему авторизации/регистрации,
2. Функцию записи времени работы/паузы,
3. Функцию контроля прихода/ухода персонала,
4. Возможность просматривать статистику по рабочему времени,
5. Функцию записи на смены.

2.2.3 Требования к техническим характеристикам

Рабочая станция (компьютер) должны иметь следующие минимальные требования:

|  |  |
| --- | --- |
| ОС | Windows 7 |
| Процессор | AMD Athlon x4 840 Quad Core 3.10 GHz |
| ОЗУ | 4 Гб |
| Видеокарта | Необязательна |

2.2.4 Требования к безопасности

В приложениях у пользователей всегда одним из главных требований является безопасность их данных. Для этого существует множество способов. В своем проекте я хочу реализовать следующие из них:

1. Главным должен быть конечно же сам код, лишенный каких-либо изъянов,
2. Использование криптографических библиотек (BouncyCastle, HashLib, Strea, Cryptor и т.д.) шифрования, либо создание своего аналога данных библиотек,
3. Ограничение прав пользователей для предотвращения утечки данных,
4. Ограничение вводимых данных для невозможности использования SQL injection.
   1. **Выбор программных и технических средств**

Adobe Photoshop (рис 2.3) — многофункциональный графический редактор, разработанный и распространяемый фирмой Adobe Systems. В основном работает с растровыми изображениями, однако имеет некоторые векторные инструменты. Продукт является лидером рынка в области коммерческих средств редактирования растровых изображений и наиболее известным продуктом фирмы Adobe.

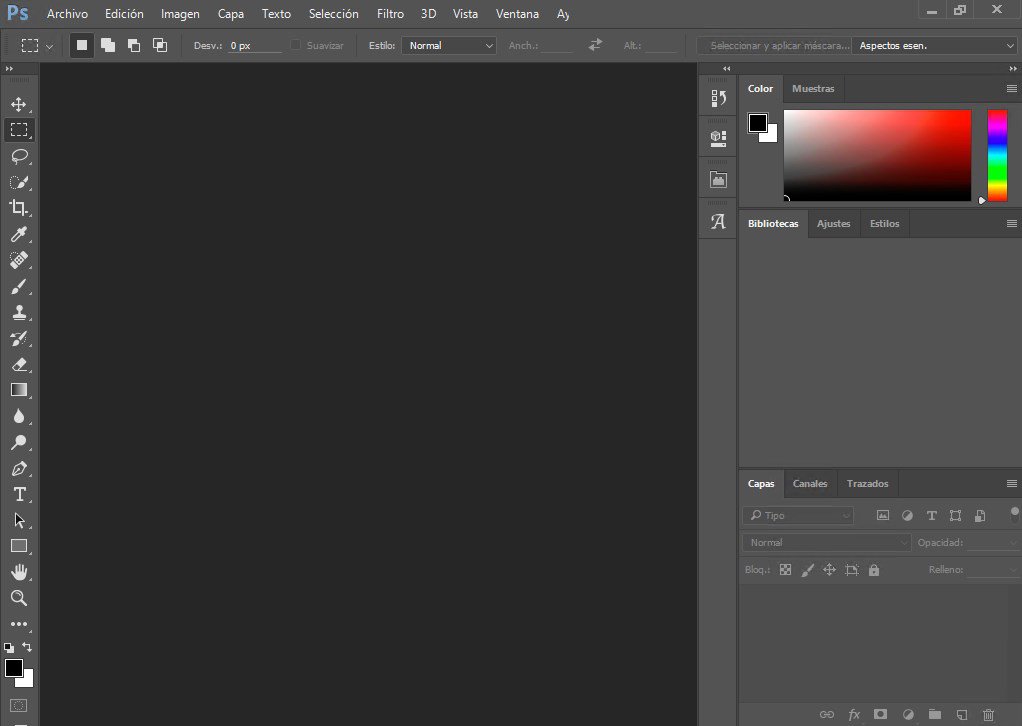


Рисунок 2.3 – Adobe Photoshop

В настоящее время Photoshop доступен на платформах macOS, Windows, в мобильных системах iOS, Windows Phone и Android. Также существует версия Photoshop Express для Windows Phone 8 и 8.1. Сейчас (2014) в США идёт бета-тестирование стриминговой версии продукта для Chrome OS. Ранние версии редактора были портированы под SGI IRIX, но официальная поддержка была прекращена, начиная с третьей версии продукта. Для версий 8.0 и CS6 возможен запуск под Linux с помощью альтернативы Windows API — Wine.

Для работы полиграфистов, дизайнеров, фотографов, мультипликаторов, архитекторов и других творческих профессий важно иметь полный набор инструментов для коррекции мультимедийных файлов. Adobe Photoshop – необычный редактор, в котором можно не только корректировать и ретушировать изображения, но и преобразовывать формы, сохранять GIF-анимации, использовать слои и эффекты. Эти функции позволяют легко трансформировать, клонировать, улучшать изображения, применять различные фильтры для создания красочной, продающей картинки.

Лицензированная версия Adobe Photoshop оснащена большим количеством функциональных возможностей для создания качественных изображений:

1. Улучшение качества отсканированных, отпечатанных снимков, устранение дефектов цветопередачи во время съемки,
2. Создание многослойного изображения с возможностью редактирования каждого в отдельности для передачи глубины цвета, работы со сложной графикой,
3. Монтирование фотографий и картинок, отдельных элементов в коллажи,
4. Восстановление старых снимков, ретушь фотографий,
5. Работа с эскизами и чертежами,
6. Большой выбор инструментов для обработки текстовых элементов (шрифты, эффекты, фильтры),
7. Обработка 2D и 3D моделей, добавление к ним текстур, работа с формами,
8. Создание интересных графических элементов для полиграфии, дизайна, оформления сайтов и так далее,
9. Подготовка к печати на принтере или в интернете и мгновенная публикация файлов,
10. Работа со стандартными форматами для растровых и векторных изображений,
11. Раскрашивание черно-белых файлов.

Adobe Photoshop – основной инструмент компьютерной графики, который используется для всех видов коммерческой обработки медиа файлов. Суть Photoshop в том, чтобы скорректировать изображение, улучшить его восприятие, придать фото и видео четкости и гармоничности. Эта программа позволяет усилить эффект от картинки, создать сложные и красивые конструкции из простых элементов.

Почему стоит использовать Adobe Photoshop:

1. Поддерживает все основные форматы медиа-файлов и инструментарии для их редактирования,
2. Имеет полноценный функционал для коррекции и создания картинок,
3. Поддерживает полноценную работу графических планшетов,
4. Может расширяться за счет плагинов,
5. Существует бесплатная версия, но для профессионального использования рекомендуем купить лицензированный ключ Adobe Photoshop.

MS Visual studio (рис. 2.4) – это программная среда по разработке приложений для ОС Windows, как консольных, так и с графическим интерфейсом.

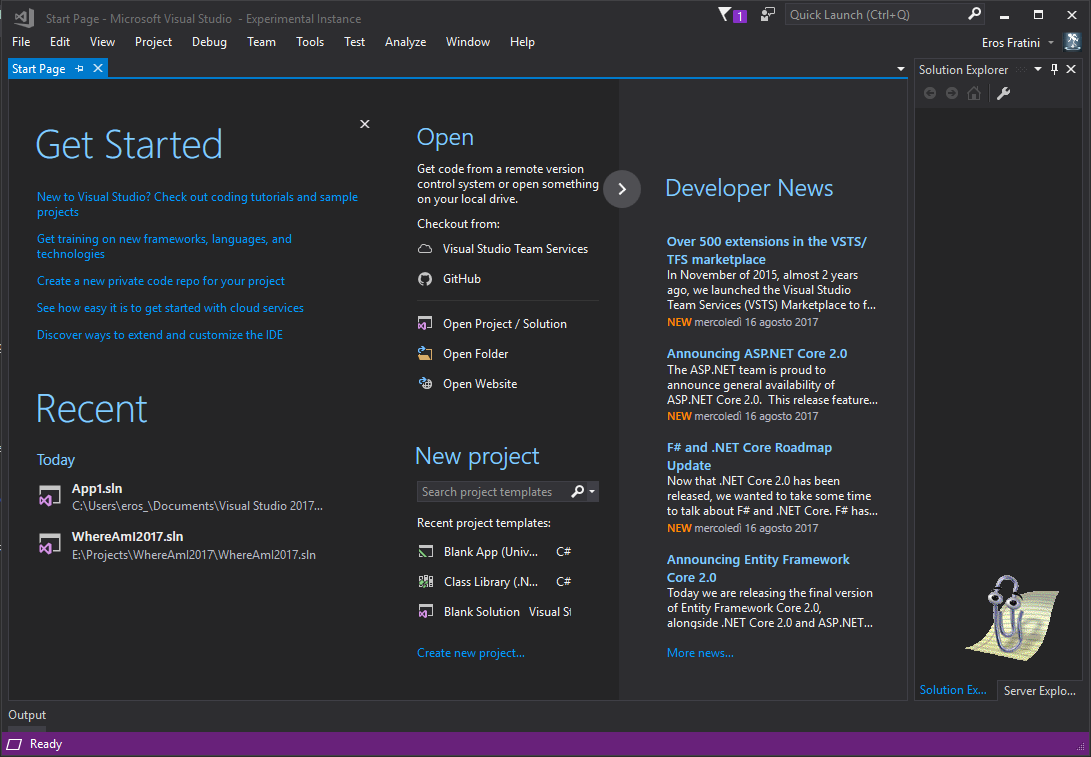


Рисунок 2.4 – Visual Studio

Функциональная структура среды включает в себя:

1. редактор исходного кода,
2. отладчик кода,
3. редактор форм, предназначенный для упрощённого конструирования графических интерфейсов,
4. веб-редактор,
5. дизайнер классов,
6. дизайнер схем баз данных.

Visual Studio также позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (Subversion и VisualSourceSafe), добавление новых наборов инструментов (для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования или инструментов для прочих аспектов процесса разработки программного обеспечения).

Преимущества:

1. Встроенный веб – сервер,
2. Поддержка множества языков при разработке,
3. Возможности отладки,
4. Возможность управления проектом.

SQL Server Management Studio (SSMS) (рис. 2.5) — это интегрированная среда для доступа, настройки, управления, администрирования и разработки всех компонентов SQL Server. Служба SSMS сочетает в себе обширную группу графических инструментов с рядом отличных редакторов сценариев для обеспечения доступа к службе SQL Server для разработчиков и администраторов всех профессиональных уровней.

Служба SSMS объединяет в единую среду функции Корпоративного Диспетчера, Анализатора Запросов и Диспетчера Анализа, включенные в прошлые выпуски службы SQL Server. В дополнение, служба SSMS работает со всеми компонентами службы SQL Server, такими как Службы Reporting Services и

Службы Integration Services. Разработчики получают знакомую среду, а администраторы баз данных — единую полнофункциональную программу, объединяющую простые в использовании графические средства и богатые возможности для создания сценариев.

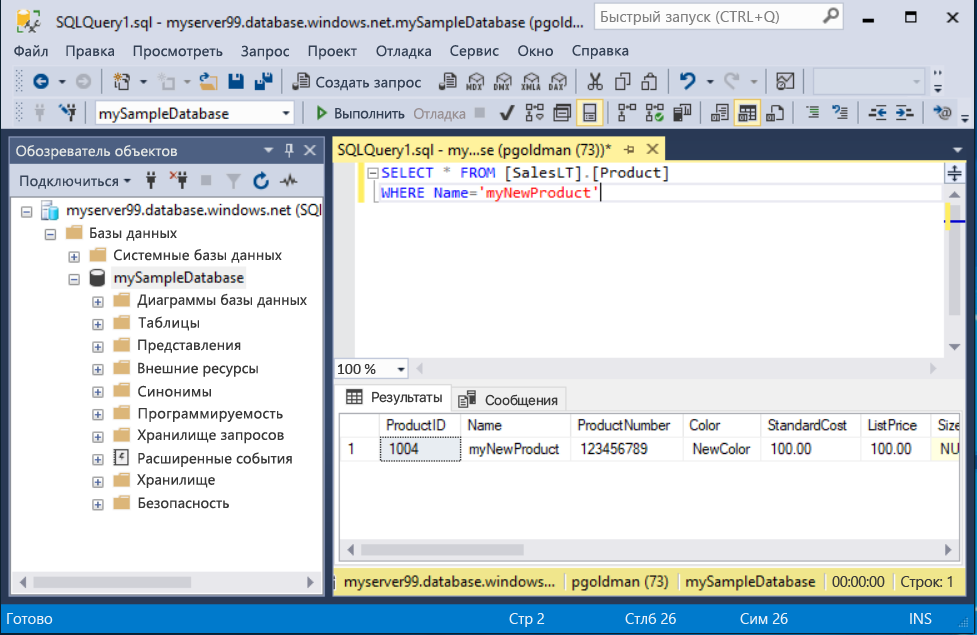


Рисунок 2.5 – MS SQL server management studio

**2.4 Проектирование системы**

ER – диаграмма (рис. 2.6) используется для разработки данных и представляет собой стандартный способ определения данных и отношений между ними.

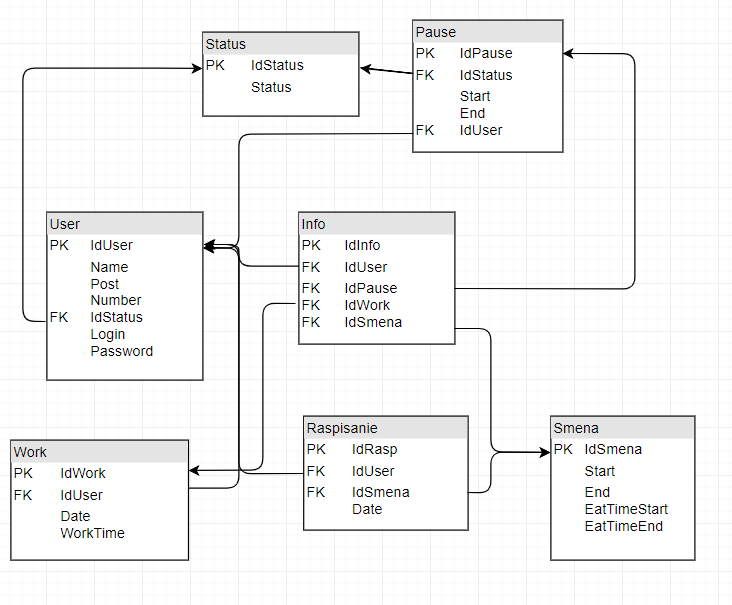


Рисунок 2.6 – ER-диаграмма

Диаграмма прецедентов (рис. 2.7) применяются для моделировании вида системы с точки зрения прецедентов (или вариантов использования).

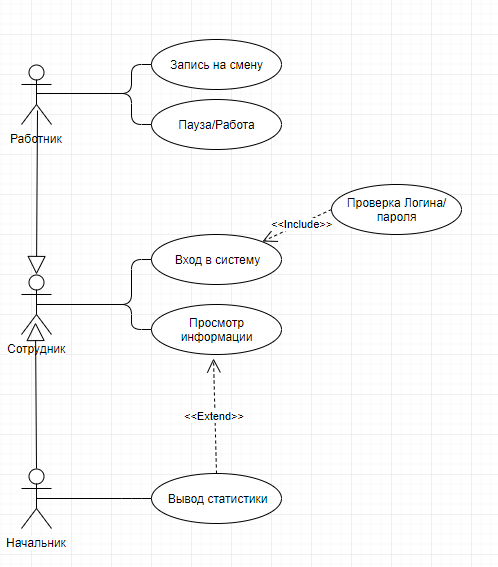


Рисунок 2.7 – Диаграмма прецедентов

Диаграмма состояний (рис. 2.8) полезна для описания алгоритма действий, но она не дает представления о поведении определенного объекта в рамках отдельного варианта использования или системы в целом, что необходимо при объектно-ориентированном программировании.



Рисунок 2.8 – Диаграмма состояний

Диаграмма развертывания (рис. 2.9) – диаграмма, на которой представлены узлы выполнения программных компонентов реального времени, а также процессов и объектов.

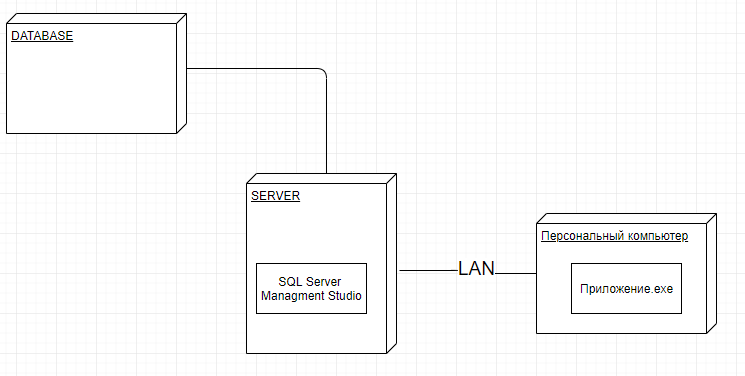


Рисунок 2.9 – Диаграмма развертывания

Диаграмма последовательности (рис. 2.10) предназначена для моделирования взаимодействия объектов Системы во времени, а также обмена сообщениями между ними.

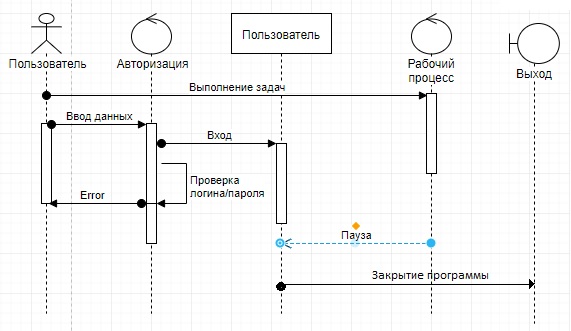


Рисунок 2.10 – Диаграмма последовательности

Блок - схема (рис. 2.11) - графическое представление алгоритма.

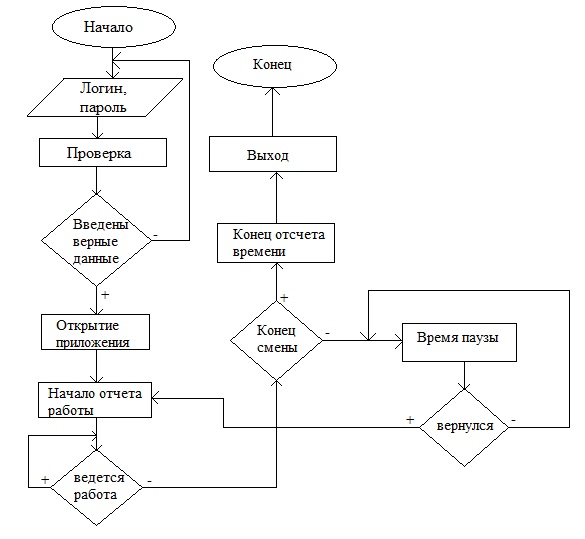


Рисунок 2.11 – Блок – схема

IDEF – диаграмма (рис. 2.12) предписывает построение иерархической системы диаграмм - единичных описаний фрагментов системы.

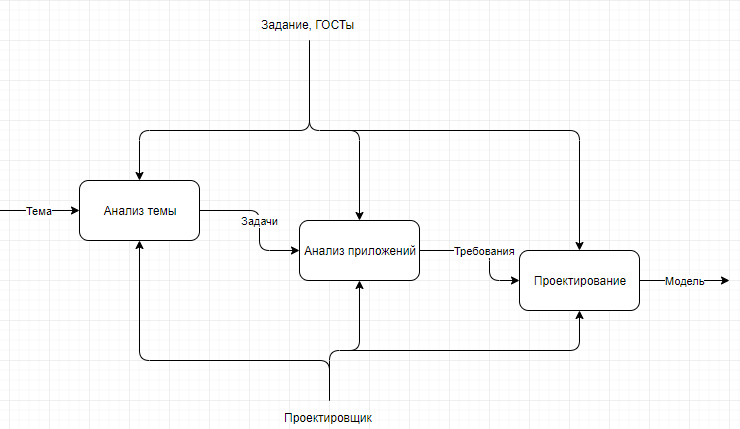


Рисунок 2.12 – IDEF - диаграмма

1. **ОРГАНИЗАЦИОННЫЙ РАЗДЕЛ**

**3.1 Организация графика разработки**

Если у вас возникает задача организовать персональную работу или работу небольшой команды без дополнительных усилий и за минимальное время, то обязательно посмотрите на Trello — это интересная система, построенная на карточках и наследующая принципы японской системы [канбан](https://lifehacker.ru/2010/06/03/kanban-dlja-odnogo-uproshhajj-planirovanie-vsled-za-japoncami/).

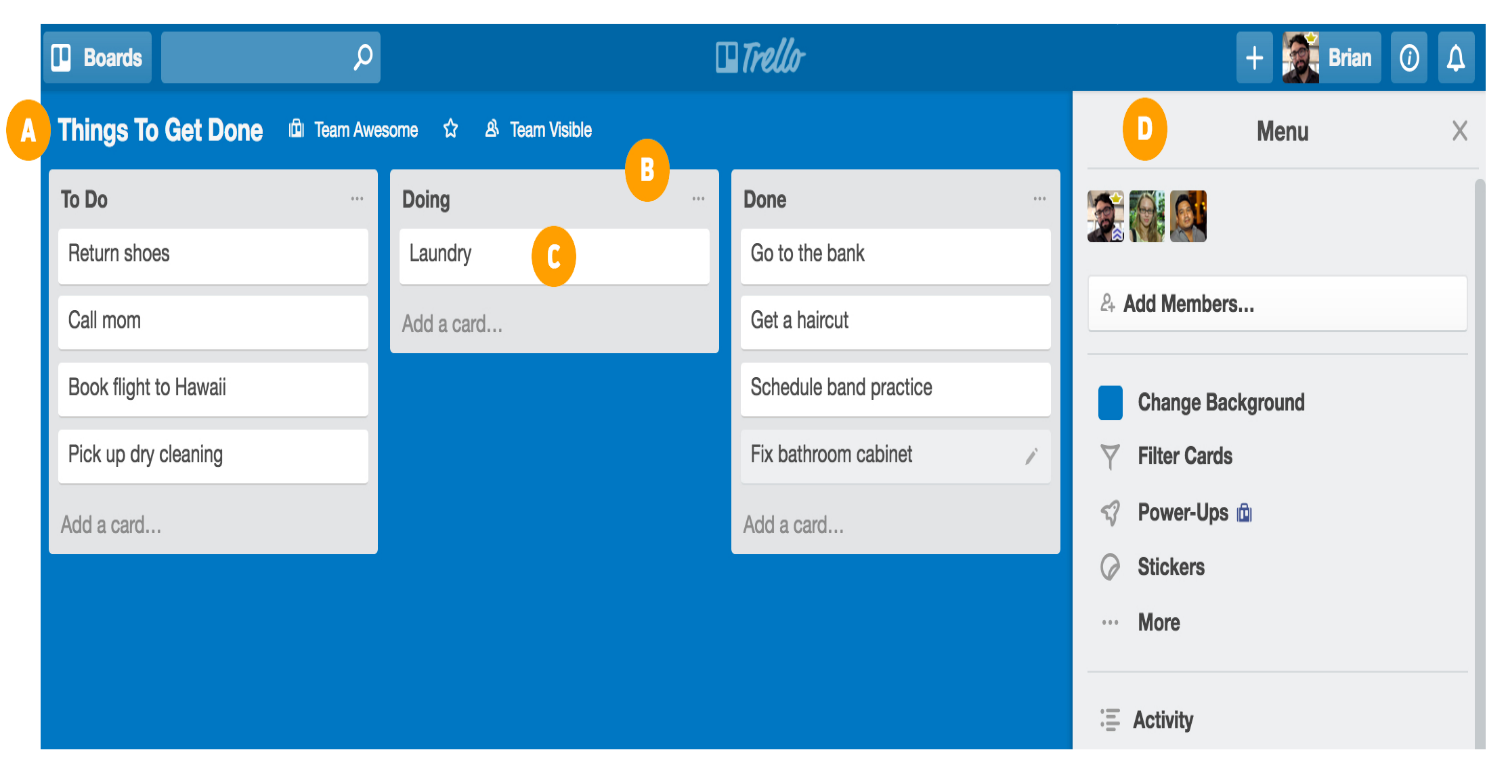


Рисунок 3.1 – Trello

Основным преимуществом Trello я бы назвал возможность видеть несколько одновременно запущенных проектов и их состояние в текущий момент времени. Если вы руководите группой разработчиков или иных исполнителей, работающих над проектами с конечной датой выполнения или фиксированной целью, то данная система может дать вам представление о ходе проектов в любой момент времени.

Trello — это пачки карточек. Каждая пачка показывает состояние любого проекта. Например, если вы отбираете кандидатов на работу, то в первой колонке будут карточки кандидатов, во второй — кандидаты, которых вы выбрали для собеседования, в третьей — с кем назначили встречи, в четвертой — с кем встретились, а в пятой — небольшой пул тех, о ком вы всерьез задумываетесь, как о своем будущем сотруднике.

Карточки имеют множество возможностей. Вы в них можете проводить обсуждения, голосования, загружать файлы данных, задавать дедлайны, назначать текстовые и цветовые метки. Чтобы к любой задаче назначить исполнителя, нужно в карточку выбрать его из списка вашей рабочей команды или просто справа перетащить на задачу аватар коллеги.

Важно заметить, что все члены рабочей группы видят в реальном времени изменения, вносимые в проект, и могут наблюдать состояния друг друга так же в реальном времени — онлайн или офлайн, смотрит на проект или нет в настоящий момент.

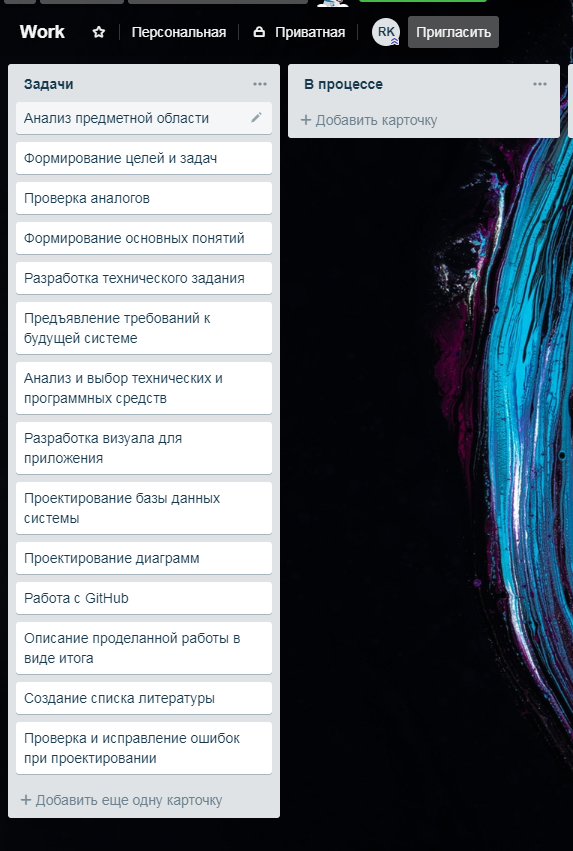


Рисунок 3.2 – Задачи

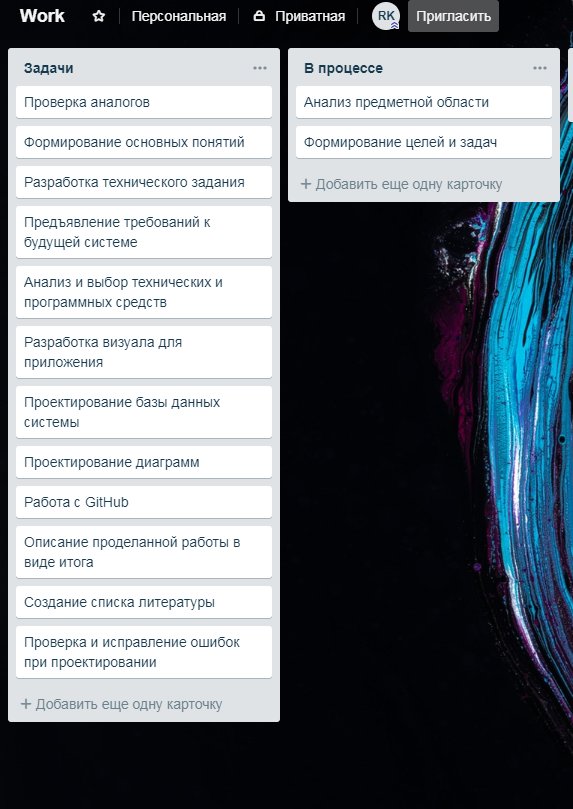


Рисунок 3.3 – Задачи в процессе

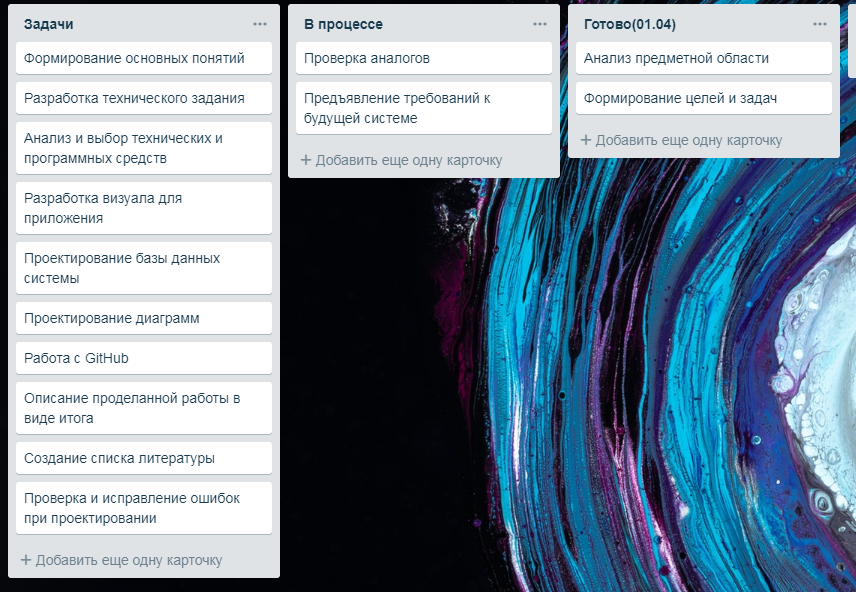


Рисунок 3.4 – Выполненные задачи на 01.04

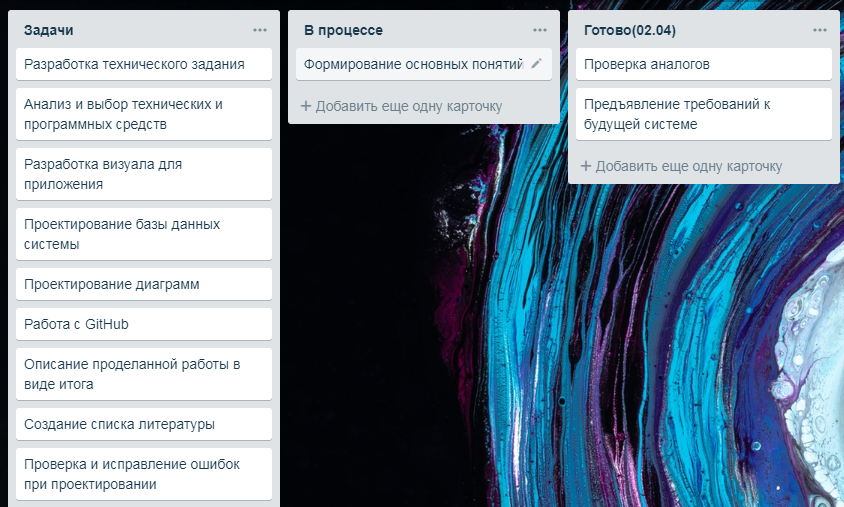


Рисунок 3.5 – Выполненные задачи на 02.04

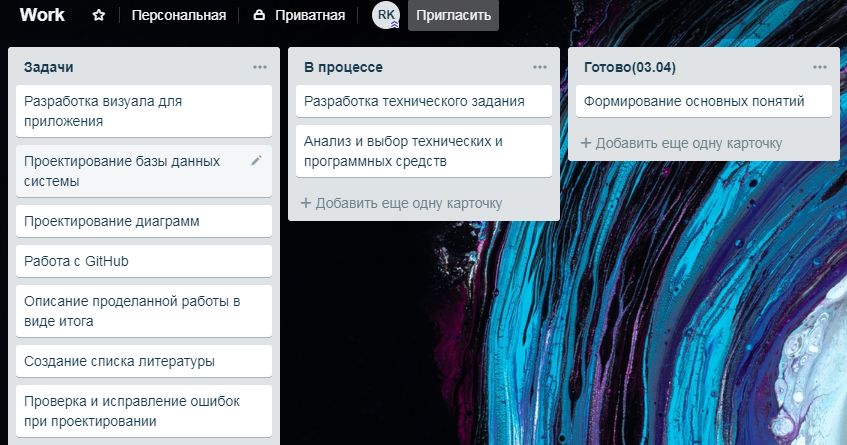


Рисунок 3.6 – Выполненные задачи на 03.04

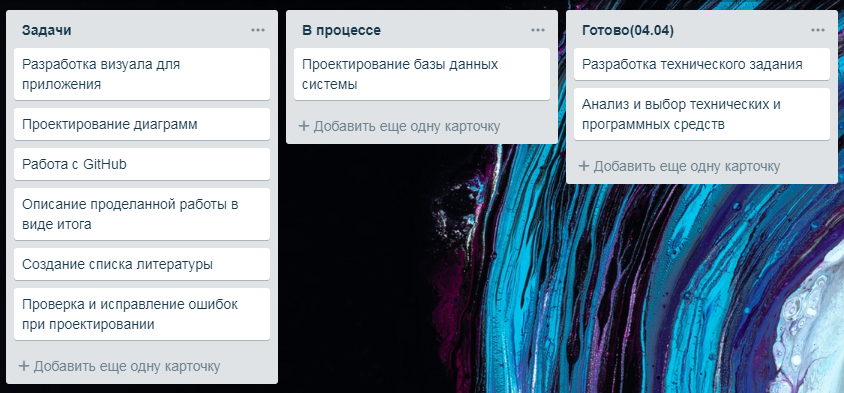


Рисунок 3.7 – Выполненные задачи на 04.04

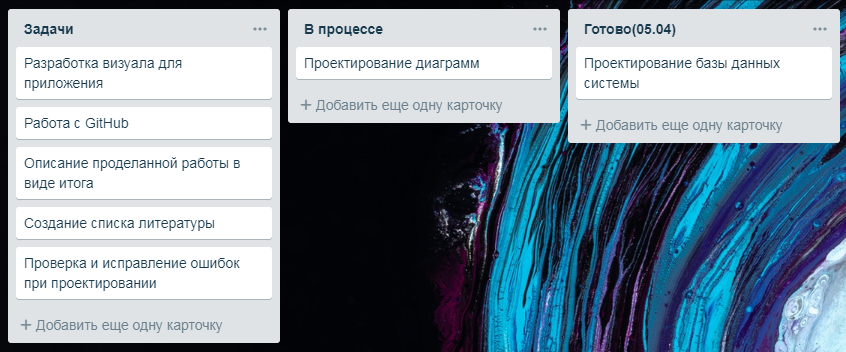


Рисунок 3.8 – Выполненные задачи на 05.04

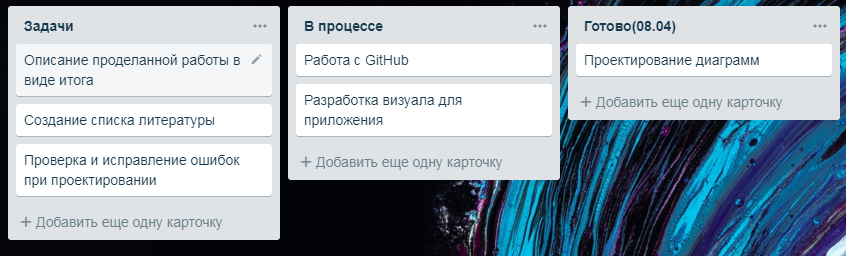


Рисунок 3.9 – Выполненные задачи на 08.04

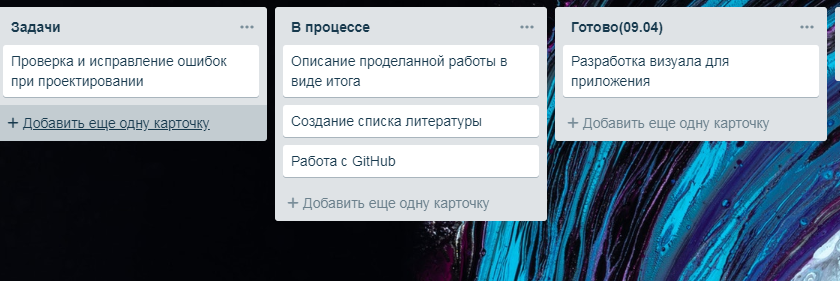


Рисунок 3.10 – Выполненные задачи на 09.04

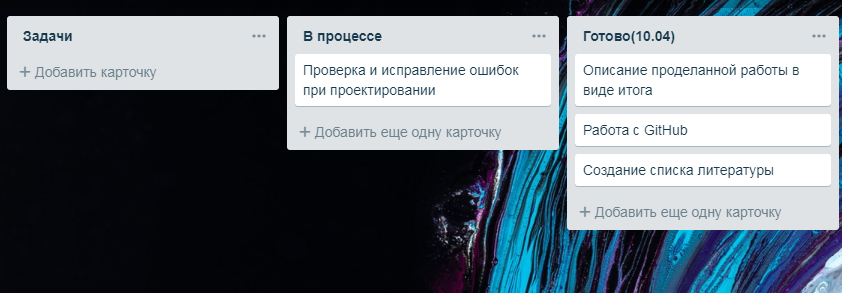


Рисунок 3.11 – Выполненные задачи на 10.04

**3.2 Система контроля версий**

GitHub — крупнейший [веб-сервис](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81) для [хостинга](https://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3) [IT-проектов](https://ru.wikipedia.org/wiki/IT) и их совместной разработки.

Веб-сервис основан на системе контроля версий [Git](https://ru.wikipedia.org/wiki/Git) и разработан на и [Erlang](https://ru.wikipedia.org/wiki/Erlang) компанией GitHub, Inc (ранее Logical Awesome). Сервис бесплатен для проектов с [открытым исходным кодом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) и (с 2019 года) небольших частных проектов, предоставляя им все возможности, а для крупных корпоративных проектов предлагаются различные платные тарифные планы.

Слоган сервиса — «Social Coding» — на русский можно перевести как «Пишем код вместе». На футболках же печатают совсем другую фразу: «Fork you!» («Ответвись!»). С одной стороны, она созвучна с англоязычным ругательством и намекает на неформальную атмосферу. С другой, эти слова напоминают, что создавать новые [форки](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BA) с Git можно легко и безболезненно — традиционно, к созданию веток разработчики проектов с открытым исходным кодом относятся негативно — а также созвучна названию одной из возможностей Github — очереди форков.

Создатели сайта называют GitHub «[социальной сетью](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) для разработчиков». Кроме размещения кода, участники могут общаться, комментировать правки друг друга, а также следить за новостями знакомых. С помощью широких возможностей Git программисты могут объединять свои [репозитории](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%B9) — GitHub предлагает удобный интерфейс, для этого он отражает вклад каждого участника в виде [дерева](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%BE_(%D1%82%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%BE%D0%B2)).

Для проектов есть личные страницы, небольшие [Вики](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8) и [система отслеживания ошибок](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BE%D1%82%D1%81%D0%BB%D0%B5%D0%B6%D0%B8%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BE%D1%88%D0%B8%D0%B1%D0%BE%D0%BA). Прямо на сайте можно просмотреть файлы проектов с [подсветкой синтаксиса](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D1%81%D0%B2%D0%B5%D1%82%D0%BA%D0%B0_%D1%81%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81%D0%B0) для большинства [языков программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F).

На платных тарифных планах можно создавать приватные репозитории, доступные ограниченному кругу пользователей.

Есть возможность прямого добавления новых файлов в свой репозиторий через веб-интерфейс сервиса.

Код проектов можно не только скопировать через [Git](https://ru.wikipedia.org/wiki/Git), но и скачать в виде обычных архивов с сайта.

Кроме [Git](https://ru.wikipedia.org/wiki/Git), сервис поддерживает получение и редактирование кода через [SVN](https://ru.wikipedia.org/wiki/SVN) и [Mercurial](https://ru.wikipedia.org/wiki/Mercurial).

На сайте есть [pastebin](https://ru.wikipedia.org/wiki/Pastebin)-сервис [gist.github.com](https://gist.github.com/) для быстрой публикации фрагментов кода.

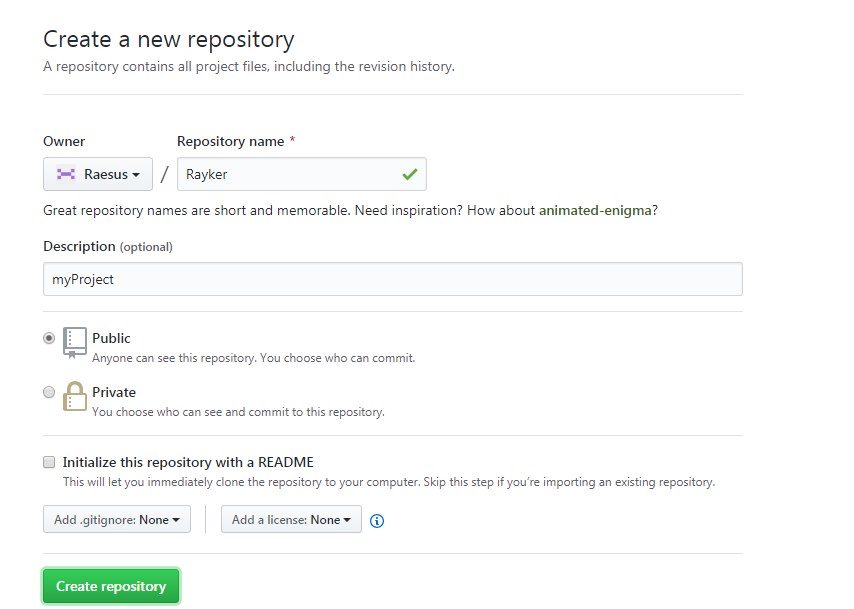


Рисунок 3.12 – Создание репозитория в GitHub

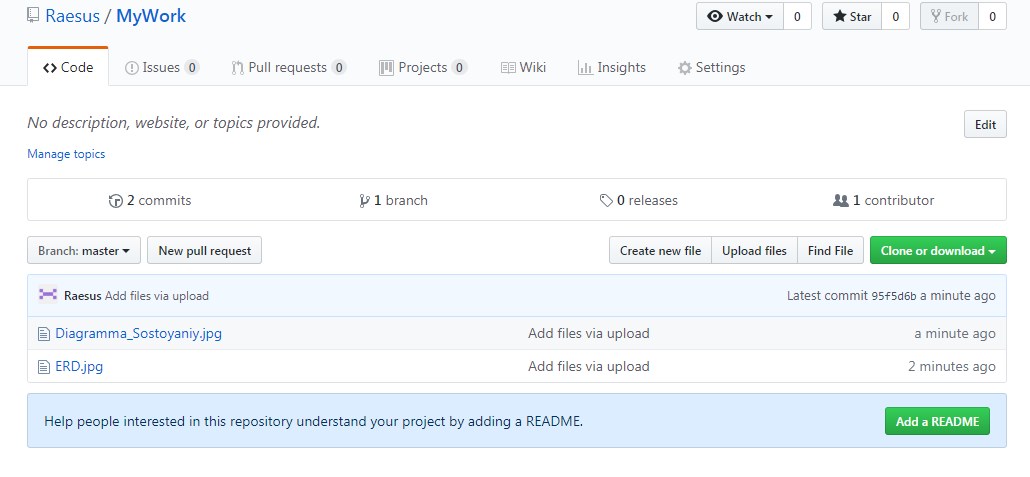


Рисунок 3.13 – Первые части работы в GitHub

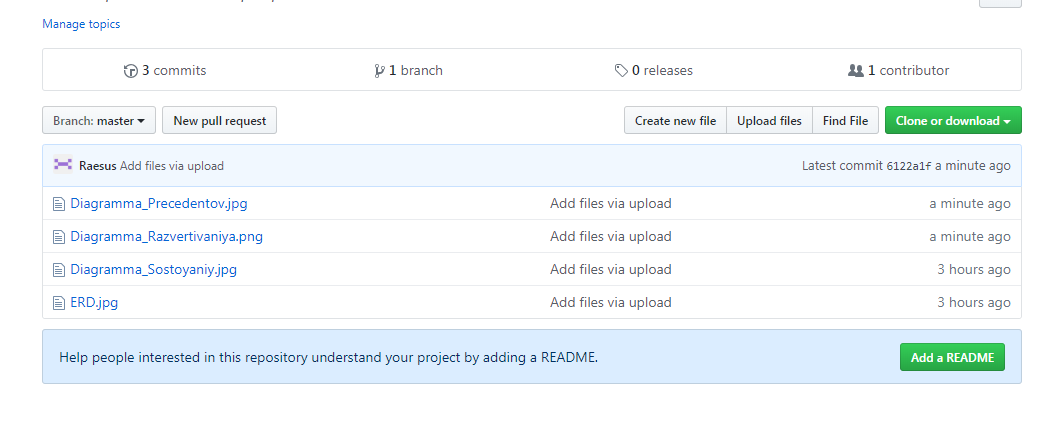


Рисунок 3.14 – Вторая часть работы в GitHub

### **ЗАКЛЮЧЕНИЕ**

В курсовом проекте была проведена проекционная работа автоматизированной системы контроля за персоналом. В рамках данной работы я выполнил следующие задачи:

1. Проанализировал информацию по данной теме,
2. Предоставил требования к системе,
3. Спроектировал базу данных для системы,
4. Спроектировать визуальную часть системы,
5. Спроектировать программную составляющую приложения.

Выполнив данные задачи, я смог прийти к поставленной мной цели, а именно спроектировать автоматизированную систему по контролю за персоналом.

Итогом послужила созданная модель, полностью удовлетворяющая всем требованиям, предоставленным с начала. Я уверен в том, что данная система имеет место быть в наше время, где каждый работодатель хочет получать от своих «ресурсов» КПД. И моя спроектированная система, как ни кстати, поможет главам компаний выполнять работу в отведенное время, а работникам будет, так сказать, надзирателем, чтобы «не расслаблялись».

### **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению [Электронный ресурс] / Межгосударственный стандарт. – М.: Стандартинформ, 2010. – URL: http://www.gostrf.com/normadata/1/4294850/4294850109.pdf (07.04.2019).
2. Статья про системы мониторинга персонала [Электронный ресурс] – URL: https://www.anti-malware.ru/analytics/Market\_Analysis/employee-monitoring-software-market-2017#part1 (07.04.2019).
3. NeoSpy официальный сайт [Электронный ресурс] – URL: https://neospy.net (07.04.2019).
4. Crocotime официальный сайт [Электронный ресурс] – URL: https://crocotime.com/ru/ (07.04.2019).
5. Bitcop официальный сайт [Электронный ресурс] – URL: https://bitcop.ru (07.04.2019).
6. Microsoft SQL server management [Электронный ресурс] – URL: https://docs.microsoft.com/ru-ru/sql/database-engine/use-sql-server-management-studio?view=sql-server-2014 (09.04.2019).
7. Статья об MS Visual Studio [Электронный ресурс] – URL: <https://alekseygulynin.ru/visual-studio/> (09.04.2019).
8. Статья об Adobe Photoshop [электронный ресурс] – URL: <https://www.softmagazin.ru/articles/vozmozhnosti_adobe_photoshop_obzor_versiy_i_dostoinstv/> (09.04.2019).
9. Draw.io [Электронный ресурс] – URL: [https://www.draw.io/](https://www.draw.io/#G1RWUsEOt3O0O6Td25LuyujHXWuvL860p-) (07.04.2019).
10. Электронная библиотека Википедия [Электронный ресурс] – URL: [https://ru.wikipedia.org/wiki/](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8) (09.04.2019).