

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Вычислительная техника
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ВТ

_____ О.В. Непомнящий

подпись

инициалы, фамилия

« _____ » _____ 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

код и наименование направления

Система учета и контроля рабочего времени

тема

Руководитель

подпись, дата

ст. преподаватель
должность, ученая степень

И. В. Матковский

инициалы, фамилия

Выпускник

подпись, дата

Д. Е. Ломанов

инициалы, фамилия

Консультант

подпись, дата

доцент, канд. техн. наук
должность, ученая степень

Л. И. Покидышева

инициалы, фамилия

Нормоконтролер

подпись, дата

доцент, канд. техн. наук
должность, ученая степень

В. И. Иванов

инициалы, фамилия

Красноярск 2019

РЕФЕРАТ

Выпускная квалификационная работа по теме «Система учета и контроля рабочего времени» содержит 35 страниц текста с 19 иллюстрациями, 2 таблицами и 14 использованными источниками.

POMODORO, АРХИТЕКТУРНЫЙ ПАТТЕРН, АЛГОРИТМ, БАЗА ДАННЫХ, .NET FRAMEWORK, ENTITY FRAMEWORK, РАСШИРЕНИЕ БРАУЗЕРА.

Цель работы – разработка и реализация приложения, предоставляющего набор функций для экономии и контроля рабочего времени.

В результате выполнения ВКР разработана система учета и контроля рабочего времени.

В выпускную квалификационную работу входит введение, 4 главы и заключение.

Во введении ставится цель, выполняется ее декомпозиция на задачи.

В первой главе рассматриваются аналоги, формируются основные требования к приложению, а также происходит выбор набора инструментов для разработки.

Во второй главе рассматривается структура системы и выбранный паттерн архитектуры, сущности базы данных, а также выбираются средства для реализации основных функций.

В третьей главе описана детализированная структура приложения и основные компоненты.

В четвертой главе описано руководство пользователя.

В заключении подводятся итоги выполненной работы.

СОДЕРЖАНИЕ

Введение	4
1 Анализ задания на ВКР	5
1.1 Сравнение существующих аналогов.....	5
1.1.1 Приложение Toggl	5
1.1.2 Приложение Pomodone.....	6
1.1.3 Приложение TMetric.....	6
1.1.4 Приложение «RescueTime».....	7
1.1.5 Сравнение существующих решений.....	7
1.2 Техника Pomodoro.....	8
1.3 Разработка технического задания	9
1.3.1 Требования к приложению	9
1.3.2 Выбор инструментов разработки	9
1.4 Выводы по разделу	10
2 Проектирование приложения	12
2.1 Архитектура приложения	12
2.2 Хранение данных	13
2.3 Реализация техники Pomodoro	13
2.4 Отслеживание активных окон приложением.....	14
2.5 Отслеживание активных вкладок браузера приложением	14
2.6 Выводы по разделу	15
3 Разработка приложения.....	16
3.1 База данных	16
3.2 Структура приложения.....	18
3.3 Алгоритм PomodoroTimerService	24
3.4 Алгоритм RecordActiveProcessService	25
3.5 Алгоритм RecordActiveTabService	26

3.6 Выводы по разделу	27
4 Руководство пользователя	28
4.1 Установка и запуск	28
4.2 Демонстрация возможностей программы	28
Заключение	32
Список сокращений	33
Список использованных источников	34

ВВЕДЕНИЕ

В эру цифровых технологий компьютер служит одновременно как рабочим местом, так и центром развлечений. Для увеличения эффективности труда требуются вспомогательные средства, улучшающие фокусировку на задачах, уменьшающие случаи отвлечения и позволяющие контролировать время, проведенное за рабочим компьютером.

На данный момент существует широкий спектр программ для учета времени. Такие программы используются представителями различных профессий с целью увеличения эффективности работы и экономии потраченного времени. Каждое приложение предлагает некие особенные опции, которые зачастую служат решающим аргументом в пользу выбора конкретного продукта. Сложившаяся ситуация имеет ряд проблем:

- неудобно использовать несколько приложений с понравившимися опциями;
- отсутствие полной картины расхода времени из-за несгруппированных отчетов нескольких используемых приложений;
- платный дополнительный функционал.

Актуальность работы обусловлена тем, что в современных реалиях экономия и грамотное использование временных ресурсов является катализатором успешных изменений в любой сфере.

Для реализации приложения необходимо решить следующие задачи:

1. Разработка технического задания.
2. Выбор инструментов разработки.
3. Разработка приложения.
4. Тестирование приложения.
5. Разработка инструкции разработчика.

1 Анализ задания на ВКР

Для того, чтобы выявить набор полезных функций и составить техническое задание необходимо рассмотреть существующие решения. Проанализировать их достоинства и недостатки.

1.1 Сравнение существующих аналогов

1.1.1 Приложение Toggl

Приложение для отслеживания времени по категориям и проектам. Основные особенности: удобство использования, дружелюбный интерфейс [1].

Основная функция – замер времени по некоторой категории. Есть возможность использовать таймер с уведомлениями.

Есть возможность просматривать отчет, содержащий информацию о затраченном времени по категориям с привязкой к календарю.

Бесплатная версия предлагает следующие возможности:

- добавлять в вашу команду до 5 человек;
- создавать неограниченное количество проектов;
- экспортировать отчеты в PDF и CSV;
- использовать категории.

Платная версия расширяет функционал:

- команда из более чем 5 человек;
- возможность делиться отчетами с заказчиками и коллегами;
- интеграция с другими приложениями;
- большая иерархия задач.

Доступные операционные системы: Android, IOS, Mac OS X 10.4, Linux, Windows.

1.1.2 Приложение Pomodoro

Приложение для увеличения эффективности труда путем упрощения соблюдения режима труда и отдыха. Используется техника Pomodoro [2]. Особенности данного продукта являются: приятный интерфейс и дизайн, простота использования, возможность интеграции с другими сервисами, отчеты [3].

Бесплатная версия предлагает возможность интеграции с двумя сервисами, ограниченный просмотр отчета.

Платная подписка от 2 евро включает просмотр отчета за 6 месяцев, возможность 3 интеграций из ограниченного списка, поддержка подзадач.

Операционная система: Android, IOS, Mac OS X 10.4, Linux, Windows.

1.1.3 Приложение TMetric

TMetric – это сервис учёта рабочего времени [4]. Основные возможности:

- отображение рабочего дня в виде удобной линейки;
- редактирование задач и рабочего времени без клавиатуры;
- быстрое переключение между задачами;
- детальные отчеты о рабочем времени, затраченном на проекты, задачи, клиентов;
- детальные отчёты о деньгах;
- теги для быстрого создания задач;
- аватары для проектов и клиентов;
- интеграция с популярными системами управления проектов;
- бесплатная версия;
- отслеживание времени;
- неограниченное количество проектов;
- ведение отчетов;

- интеграция с веб-приложениями;
- стоимость платной версии начинается от 4 долларов;
- финансовые возможности;
- управление задачами;
- отчеты об использовании приложений.

Операционные системы: IOS, macOS, Linux, Windows.

1.1.4 Приложение «RescueTime»

Приложение собирает статистику проведенного времени в различных приложениях и на сайтах для получения полной картины проведенной за компьютером сессии [5].

Бесплатная версия предоставляет следующие возможности:

- отслеживание времени, проведенного в приложениях и сайтах;
- установка целей;
- рассылка отчетов;
- максимум 3-х месячный отчет.

Платная версия расширяет функционал:

- уведомление по достижению целей;
- блокировка отвлекающих сайтов на время;
- детализированный отчет;
- быстрая обработка данных;
- просмотр отчета за все время.

1.1.5 Сравнение существующих решений

Сравнение будет происходить по следующим критериям:

- наличие платной подписки и ее стоимость;
- наличие функции замера времени;
- наличие функции отслеживания времени в активных приложениях;
- возможность генерировать отчет.

В таблице 1 приведено сравнение аналогов. Можно отметить, что полный спектр функций существующие решения предоставляют по платной подписке. Представленные решения позволяют создавать отчеты. А также их можно разделить на две группы: замер времени и отслеживание приложений и сайтов.

Таблица 1 – Сравнение существующих решений

Сервис/приложение	Платная подписка	Замер времени	Время в программах	Отчеты
Toggl	9\$	+	-	+
PomoDone	5\$	+	-	+
TMetric	4\$	+	-	+
RescueTime	4,5\$	-	+	+

Для увеличения производительности и уменьшения факторов отвлечения целесообразно использовать оба подхода, но использовать несколько приложений неудобно, так как информация будет храниться отдельно, что усложняет ее анализ.

1.2 Техника Pomodoro

Рассмотренные ранее приложения Toggl и PomoDone в качестве основной функции используют реализацию техники Pomodoro [2].

Автором техники является Франческо Чирилло, который предлагал разбивать задачи на 25-минутные периоды (рисунок 1), после каждого шел короткий перерыв. После четырех временных блоков следовал длинный пе-

рерыв. Техника направлена на поддержание концентрации и увеличении производительности.

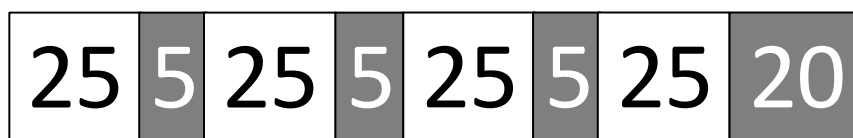


Рисунок 1 – Временные блоки техники Pomodoro

1.3 Разработка технического задания

1.3.1 Требования к приложению

На основе рассмотренных решений составлены следующие требования к разрабатываемому проекту:

1. Реализация таймеров на основе техники Pomodoro.
2. Учет времени, проведенного в программах и на сайтах.
3. Возможность создания и настройки профилей допустимых программ и сайтов.
4. Уведомление при использовании программы или сайта из черного списка.
5. Отображение расширенного отчета по потраченному времени (время по категориям/проектам/программам/сайтам с привязкой к календарю).

1.3.2 Выбор инструментов разработки

В качестве платформы разработки был выбран .NET Framework 4.7.2 [6], как одно из современных решений, предоставляющих широкие возможности для разработки приложения под операционную систему Windows.

В качестве технологии разработки приложения с графическим интерфейсом выбрана Windows Presentation Foundation (WPF) [7]. С помощью WPF можно создавать широкий спектр как автономных, так и запускаемых в брау-

зере приложений. Также WPF имеет широкие возможности для работы с графикой, так как в ее основе лежит графическая технология DirectX, которая предоставляет аппаратное ускорение графики, что повышает производительность WPF.

В качестве языка программирования был выбран один из самых быстроразвивающихся и востребованных языков, предоставляющий возможности для разработки широкого спектра приложений – C# 7.0 [8].

Для того чтобы приложение работало с данными независимо от типа хранилища, используется технология Entity Framework [9], которая позволяет абстрагироваться от конкретной реализации базы данных и оперировать данными в виде классов, вместо таблиц, индексов, первичных и внешних ключей.

В качестве хранилища данных выбрана база данных на основе технологии SQLite [10], как компактное встраиваемое решение, не использующее парадигму «клиент-сервер» и не требующие отдельных процессов для работы.

Для уменьшения связности компонентов приложения используется внедрение зависимостей с помощью популярной технологии Autofac [11], которая является решением с открытым исходным кодом и подробной документацией с понятными примерами.

В проекте используется интегрированная среда разработки Visual Studio 2017 Community [12], как наиболее удобное решение для работы с продуктами Майкрософт.

1.4 Выводы по разделу

В ходе анализа задания на ВКР произведен поиск и сравнение существующих решений, предоставляющих функции для отслеживания времени. В ходе сравнения были выявлены их достоинства и недостатки. В результате было разработано техническое задание, а также определен набор инструментов для разработки приложения, отображенных в таблице 2.

Таблица 2 – Инструменты разработки

Платформа	.NET Framework 4.7.2
Технология	Windows Presentation Foundation
Язык программирования	C# 7.0
ORM для базы данных	Entity Framework Core
База данных	SQLite
IoC-контейнер	Autofac
Интегрированная среда разработки	Visual Studio 2017 Community

2 Проектирование приложения

2.1 Архитектура приложения

Для того, чтобы отделить логику приложения от его визуальной части широко применяются архитектурные паттерны семейства MV*. Так как для реализации приложения используется технология WPF, которая поддерживает функцию привязки данных, из семейства MV* был выбран архитектурный паттерн Model-View-ViewModel (MVVM) [13], изображенный на рисунке 2.

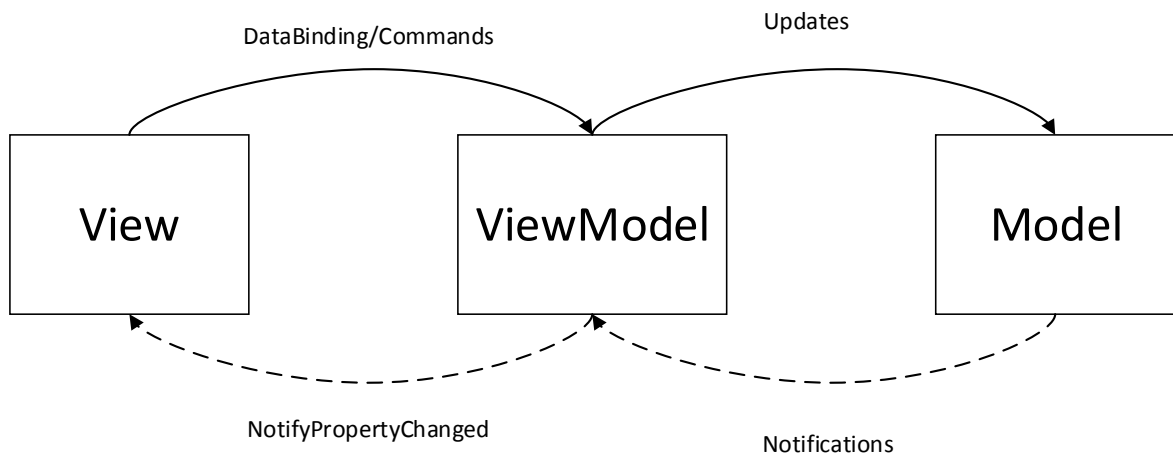


Рисунок 2 – Архитектурный паттерн MVVM

Для реализации возможности отслеживания активных вкладок в браузере используется расширение для браузера Google Chrome. Таким образом проект состоит из двух компонентов: WPF-приложения и расширения для браузера. Если WPF-приложение использует архитектурный паттерн MVVM, то в случае расширения в использовании паттерна нет необходимости, ввиду простоты логики и тривиальности структуры. На рисунке 3 отображена структура приложения.

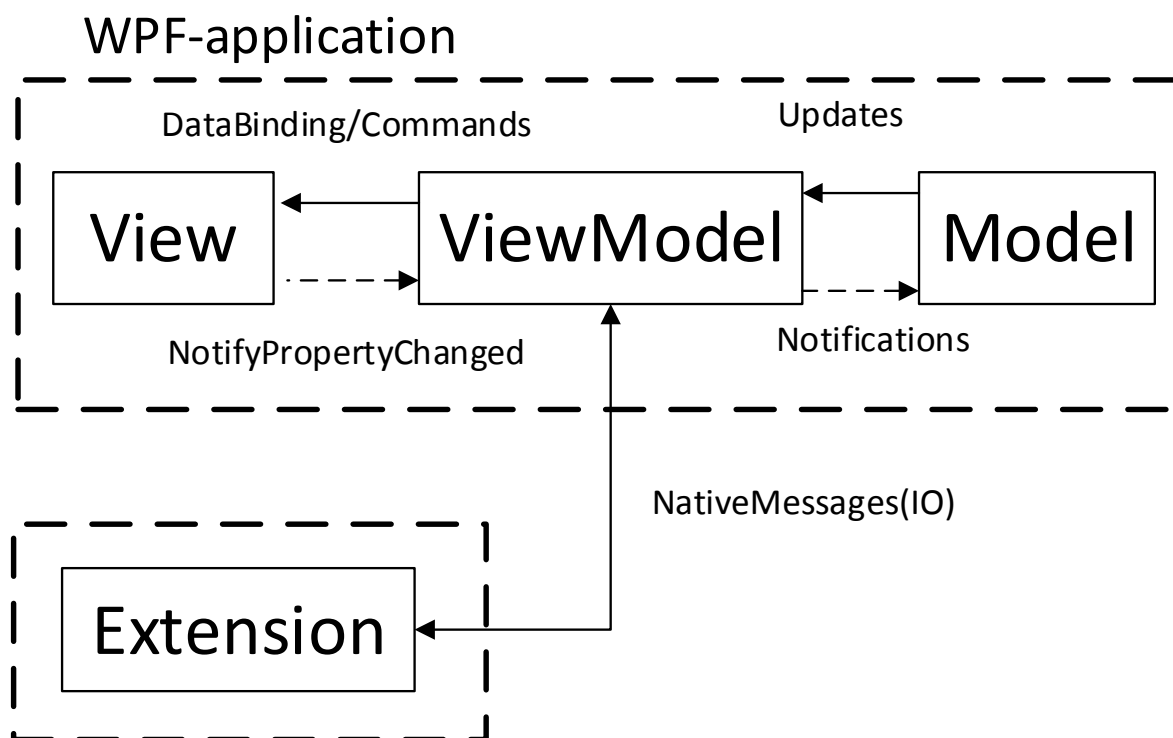


Рисунок 3 – Структура приложения

2.2 Хранение данных

Для реализации запланированных функций необходимо хранить следующие данные:

- временные промежутки и сопутствующую информацию;
- информацию об активных процессах;
- информацию об активных вкладках;
- профили и черные списки процессов и вкладок.

2.3 Реализация техники Pomodoro

Для реализации техники Pomodoro планируется использовать встроенные инструменты платформы .NET Framework – класс Timer, который находится в сборке System.dll и пространстве имен System.Timers. Реализацию техники Pomodoro планируется вынести в отдельный модуль.

2.4 Отслеживание активных окон приложением

Отслеживание приложением активных окон планируется осуществлять при помощи обработчика системных событий операционной системы Windows. Обработчик планируется прикрепить к событию с помощью библиотеки user32.dll которая предоставляет доступ к следующим функциям:

- SetWinEventHook – служит для подписки обработчика на событие;
- UnhookWinEvent – служит для отписки от события ранее установленного обработчика;
- GetWindowText – позволяет получить заголовок окна;
- GetWindowThreadProcessId позволяет получить идентификатор процесса окна.

Библиотека user32.dll предоставляет доступ сразу ко всем событиям операционной системы. Для реализации функции отслеживания активных окон приложения требуется единственное событие типа EVENT_SYSTEM_FOREGROUND.

2.5 Отслеживание активных вкладок браузера приложением

Отслеживание приложением активных вкладок браузера планируется осуществлять при помощи вспомогательного расширения в самом браузере, которое с помощью API будет иметь доступ к URL конкретной вкладки и ее заголовку. Далее информация будет передаваться в приложение. В браузере Google Chrome доступно следующее API [14] для осуществления данной функции:

- chrome.tabs – инструмент, позволяющий взаимодействовать с вкладками в браузере;
- Native Messaging – инструмент, позволяющий обмениваться сообщениями между расширением браузера и приложением через ОС.

2.6 Выводы по разделу

В ходе проектирования приложения был выбран архитектурный паттерн MVVM для обеспечения модульности приложения и удобства последующей модификации исходного кода. Определены основные сущности для хранения в базе данных, позволяющие реализовать главные функции приложения. Выбраны библиотеки и подходы для реализации основных функций приложения.

3 Разработка приложения

3.1 База данных

В ходе проектирования выявлены следующие сущности, которые необходимо хранить в базе данных.

Таблица TimeRecord хранит информацию для реализации функции отслеживания времени: название промежутка, дата начала, дата конца. Так как SQLite не поддерживает формат DateTime, решено хранить даты в формате строки. Таблицы Project и Tag связаны с TimeRecord связью «один ко многим» и необходимы для фильтрации. Эти таблицы содержат наименование и внешний ключ, указывающий на TimeRecord.

Таблицы ProcessType и ProcessRecord необходимы для реализации функции отслеживания активного окна. ProcessType представляет собой тип процесса, содержит название процесса и заголовок окна. Таблица ProcessRecord связана с ProcessType связью «один ко многим» и содержит информацию о периоде активности окна, а также справочную информацию.

Таблицы TabType и TabRecord необходимы для реализации функции отслеживания активной вкладки в браузере. TabType содержит название типа и URL. Таблица TabRecord связана с TabType связью «один ко многим» и, по аналогии с ProcessRecord, содержит информацию о периоде активности вкладки, а также ее URL.

Таблицы Profile, BannedProcessNote и BannedTabNote необходимы для реализации системы уведомлений, которые сигнализируют о том, что в данный момент активна вкладка и/или процесс, которые находятся в черном списке. Profile содержит только наименование, но с этой таблицей связью «один ко многим» связаны BannedProcessNote и BannedTabNote, которые содержат соответственно внешний ключ на ProcessType и TabType, а также на Profile. Диаграмма базы данных отображена на рисунке 4.

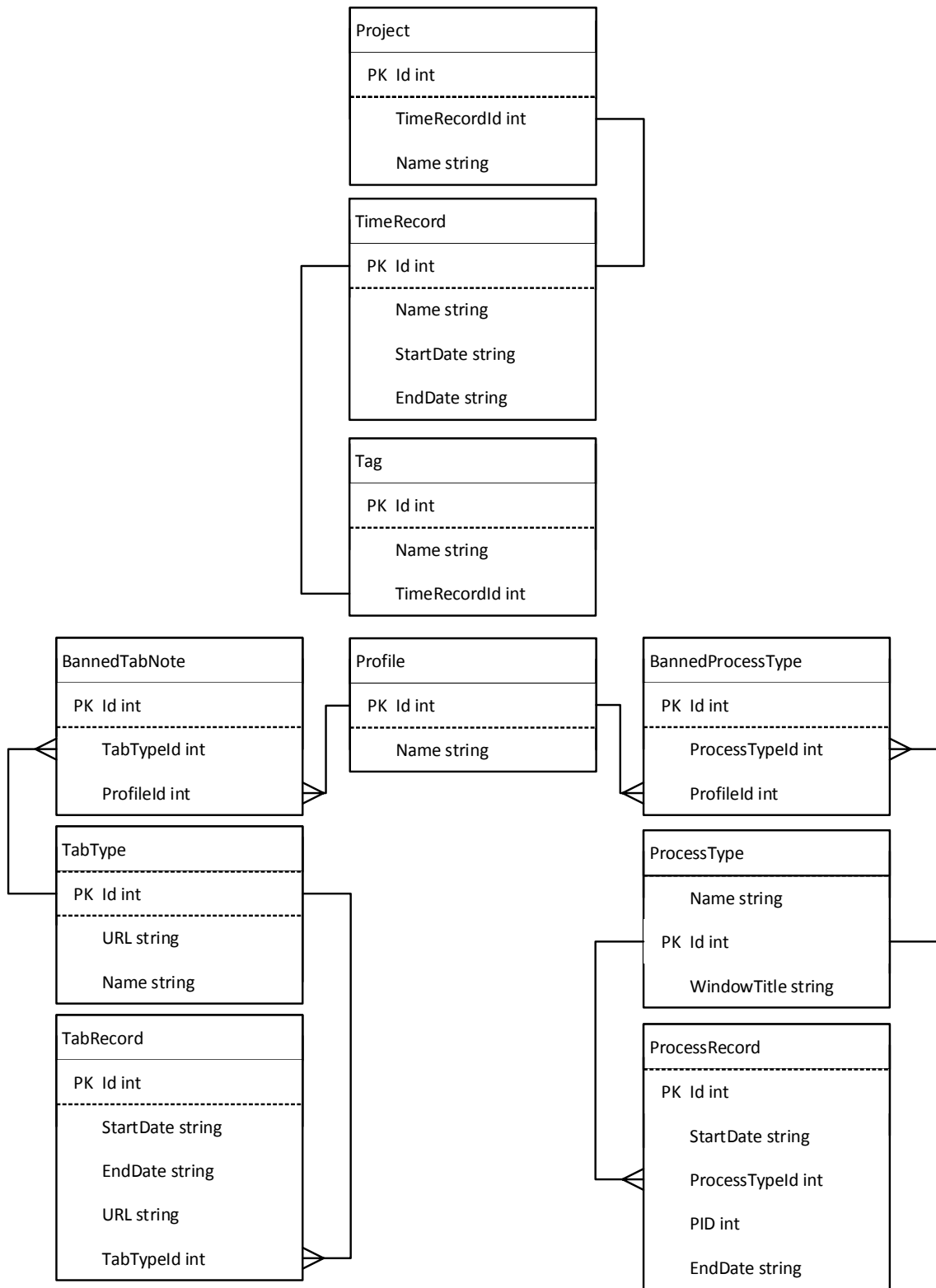


Рисунок 4 – Диаграмма базы данных

3.2 Структура приложения

В ходе разработки приложения была детализирована его структура, которая отображена на рисунке 5. Рассмотрим составные части основных модулей: View, ViewModel, Model, Extension.

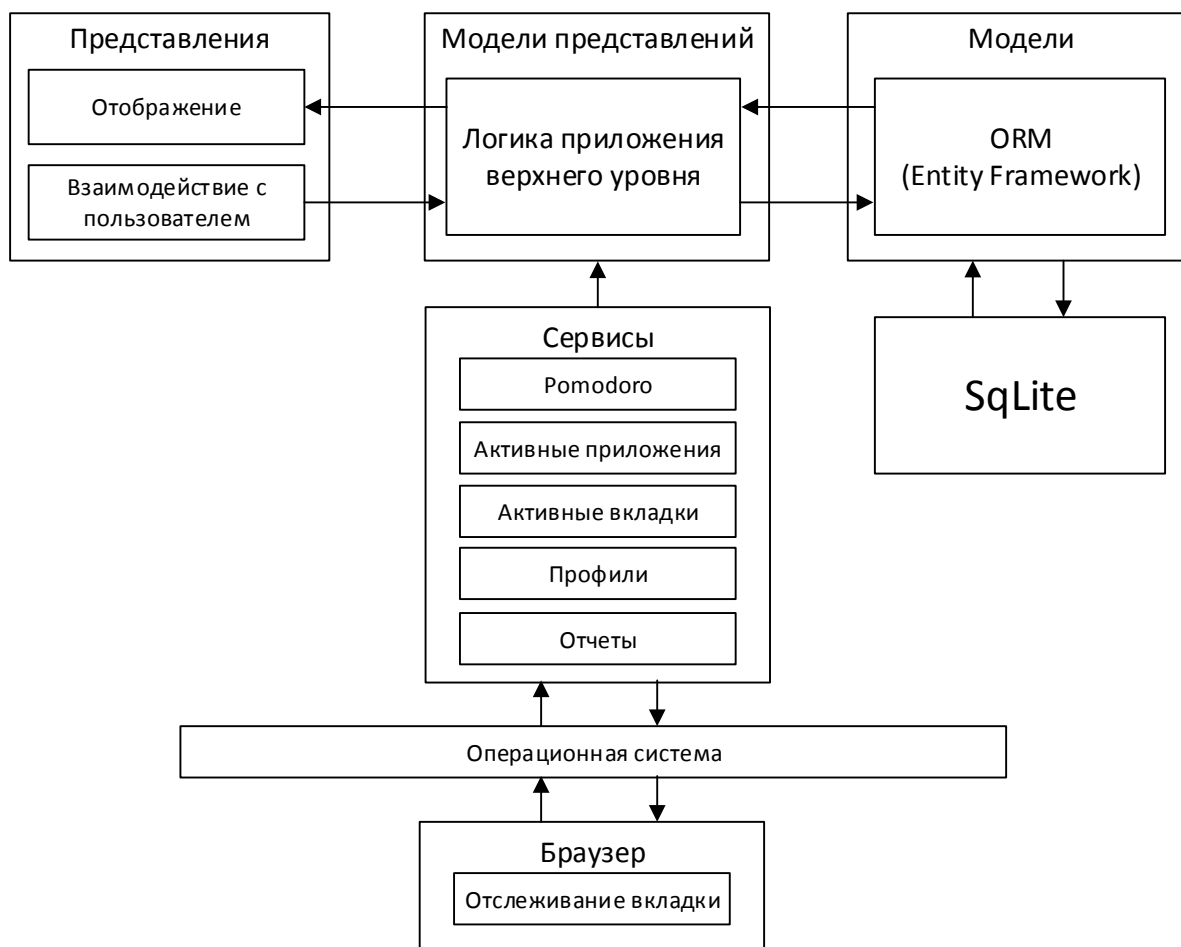


Рисунок 5 – Структура приложения

View состоит из следующих классов: MainView, EditView, SettingsView, ReportView, ProfileView. Эти классы представляют собой интерфейс приложения (рисунок 6).

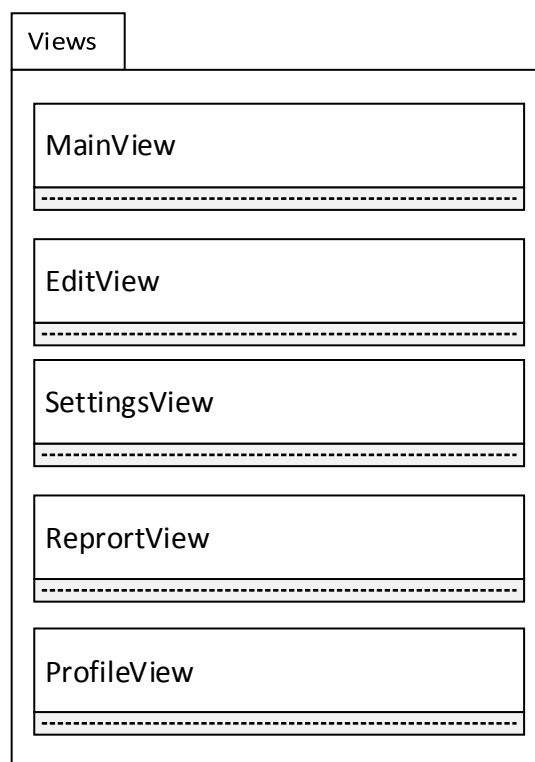


Рисунок 6 – Классы модуля View

Архитектурный модуль ViewModel состоит из следующих подмодулей: ViewModels, Services, Common, Utils.

ViewModels состоит из следующих классов (рисунок 7):

- MainViewModel содержит в себе логику представления MainView, а также логику верхнего уровня приложения;
- EditViewModel содержит логику редактирования выбранного промежутка времени, а также отображения дополнительной информации;
- SettingsViewModel содержит логику настройки таймера;
- ReportViewModel содержит логику отвечающую за фильтрацию данных, а также последующего вывода отфильтрованных данных в виде графиков.

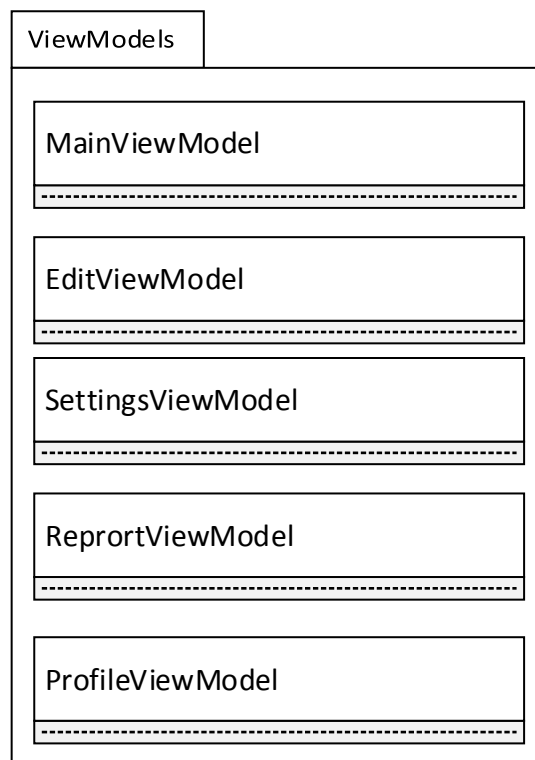


Рисунок 7 – Классы модуля ViewModel

Модуль Services состоит из следующих классов (рисунок 8):

- PomodoroTimerService содержит реализацию техники Pomodoro;
- RecordActiveProcessService содержит реализацию отслеживания активного окна;
- RecordActiveTabService содержит реализацию отслеживания активной вкладки в браузере;
- ReportService предоставляет функционал для фильтрации информации, хранящейся в базе данных;
- ProfileService предоставляет функционал для создания черных списков и определения, нахождения процесса/вкладки в черном списке.

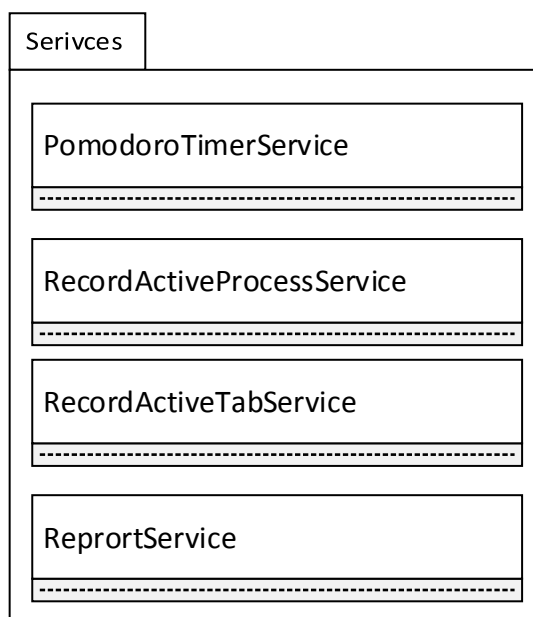


Рисунок 8 – Классы модуля Services

Модуль Common состоит из следующих классов (рисунок 9):

- TriggerState – класс, использующийся для стилизации кнопок;
- RelayCommand – класс использующийся для осуществлении привязки элементов представления и данных в модели представлений.

Utils состоит из одного класса – TimerHelper, который содержит вспомогательные функции для таймеров.

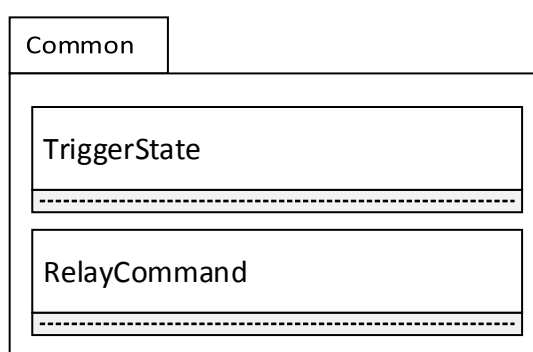


Рисунок 9 – Классы модуля Common

Модуль Model содержит класс AppContext, через который происходит взаимодействие с базой данных, а также классы, которые представляют таблицы базы данных (рисунок 10):

- TimeRecord – содержит информацию по замеренному промежутку времени: название, дата начала замера, дата окончания замера;
- Tag – элемент, использующийся в фильтрации промежутков времени;
- Project – элемент, позволяющий объединять несколько замеренных промежутков времени;
- ProcessType – тип процесса активного окна;
- ProcessRecord – содержит информацию об активности процесса: название, дата начала, дата окончания;
- TabType – тип вкладки;
- TabRecord – содержит информацию об активности вкладки: название URL, дата начала, дата окончания;
- Profile – элемент, объединяющий черные списки приложений и вкладок;
- BannedProcessNote – запись, содержащая информацию об приложении из черного списка;
- BannedProcessType – запись, содержащая информацию об вкладке из черного списка.

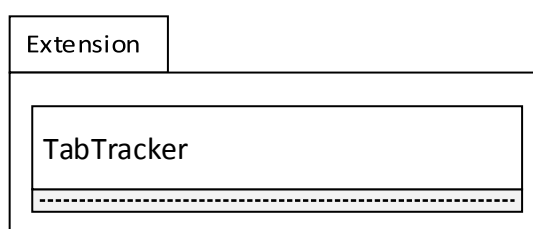


Рисунок 10 – Скрипт модуля Extension

Модуль Extension содержит скрипт TabTracker (рисунок 11).

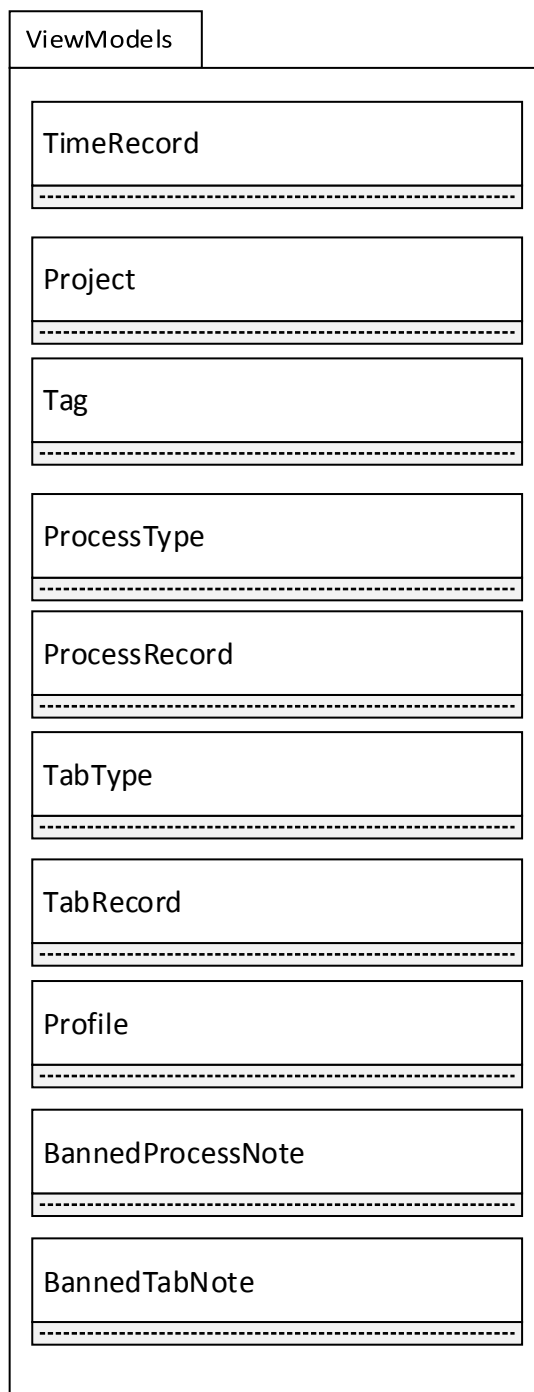


Рисунок 11– Классы модуля Model

3.3 Алгоритм PomodoroTimerService

Алгоритм заключается в последовательном запуске таймеров и уведомлении класса владельца об срабатывании таймеров. На рисунке 12 отображено как класс владелец MainViewModel запускает PomodoroTimerService. Далее сервис запускает таймер, по истечении которого передает информацию о замеренном промежутке времени в MainViewModel, после чего MainViewModel сохраняет информацию, используя AppContext.

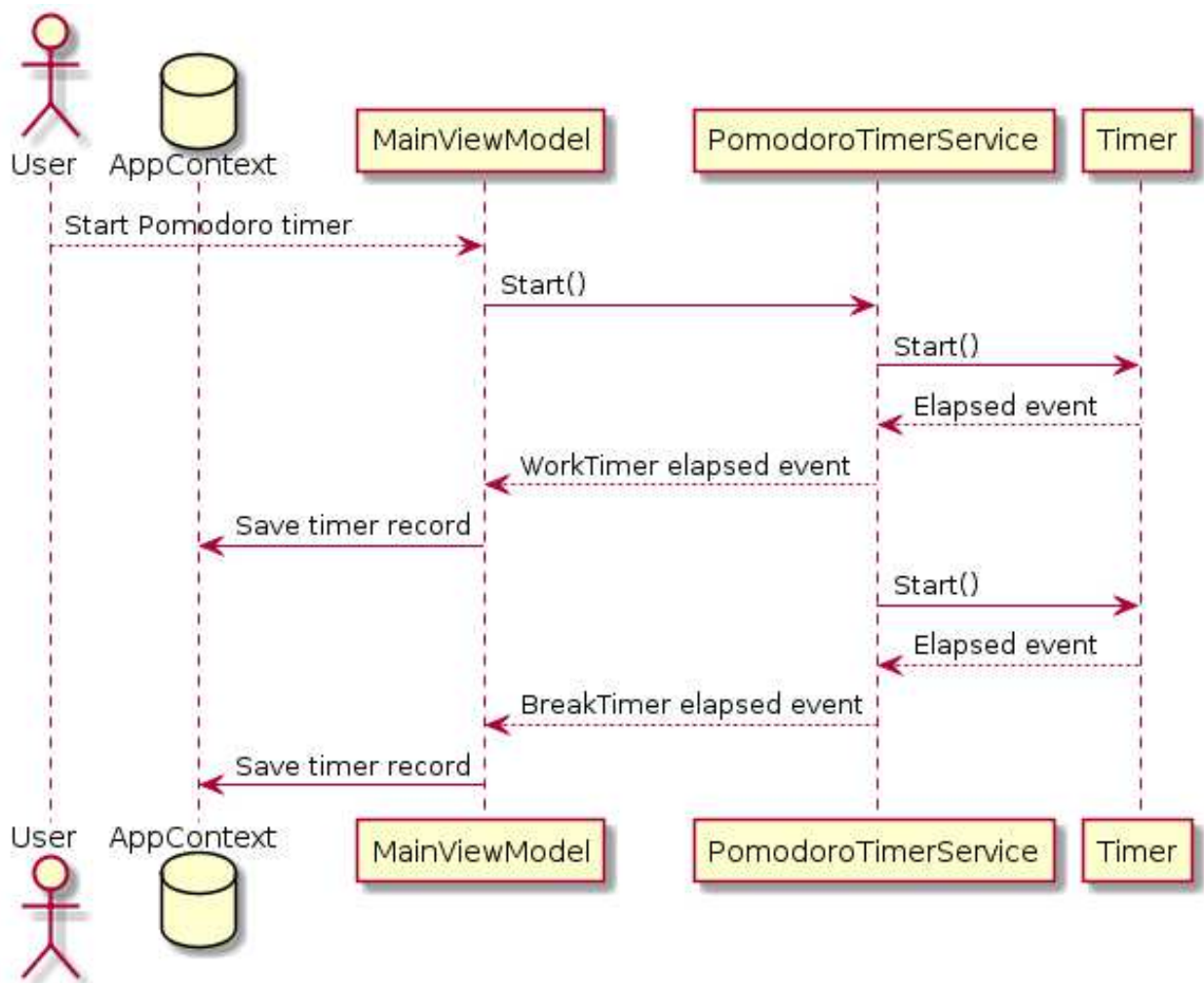


Рисунок 12 – Диаграмма последовательностей PomodoroTimerService

3.4 Алгоритм RecordActiveProcessService

Логика алгоритма заключается в подписке специального обработчика на события операционной системы Windows. В обработчике происходит фильтрация событий, необходимое событие – переключение окна на другое.

На рисунке 13 изображена диаграмма последовательностей, на которой отображено, как класс владелец MainViewModel запускает сервис RecordActiveProcessService. Сервис подписывает обработчик на события операционной системы, по событию переключения окон сервис отправляет информацию об активном процессе в класс владелец.

Класс владелец сохраняет информацию через AppContext.

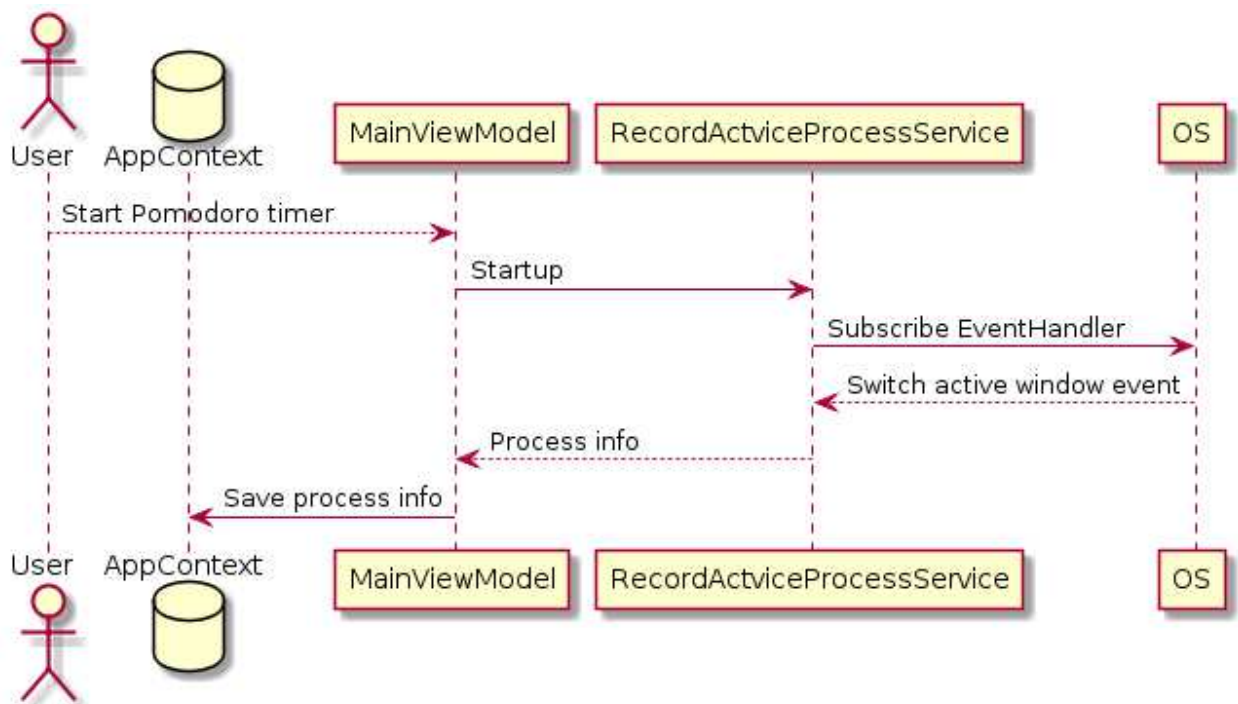


Рисунок 13 – Диаграмма последовательностей RecordActiveProcessService

3.5 Алгоритм RecordActiveTabService

Алгоритм заключается в следующем: расширение в браузере передает информацию об активной вкладке через средства операционной системы в приложение. На рисунке 14 отображена диаграмма последовательностей алгоритма. Класс владелец MainViewModel активирует RecordActiveTabService, который начинает прослушивать сообщения от расширения в браузере.

Как только вкладка изменяется, сервис передает информацию об активной вкладке в класс владелец, который сохраняет информацию в базу данных.

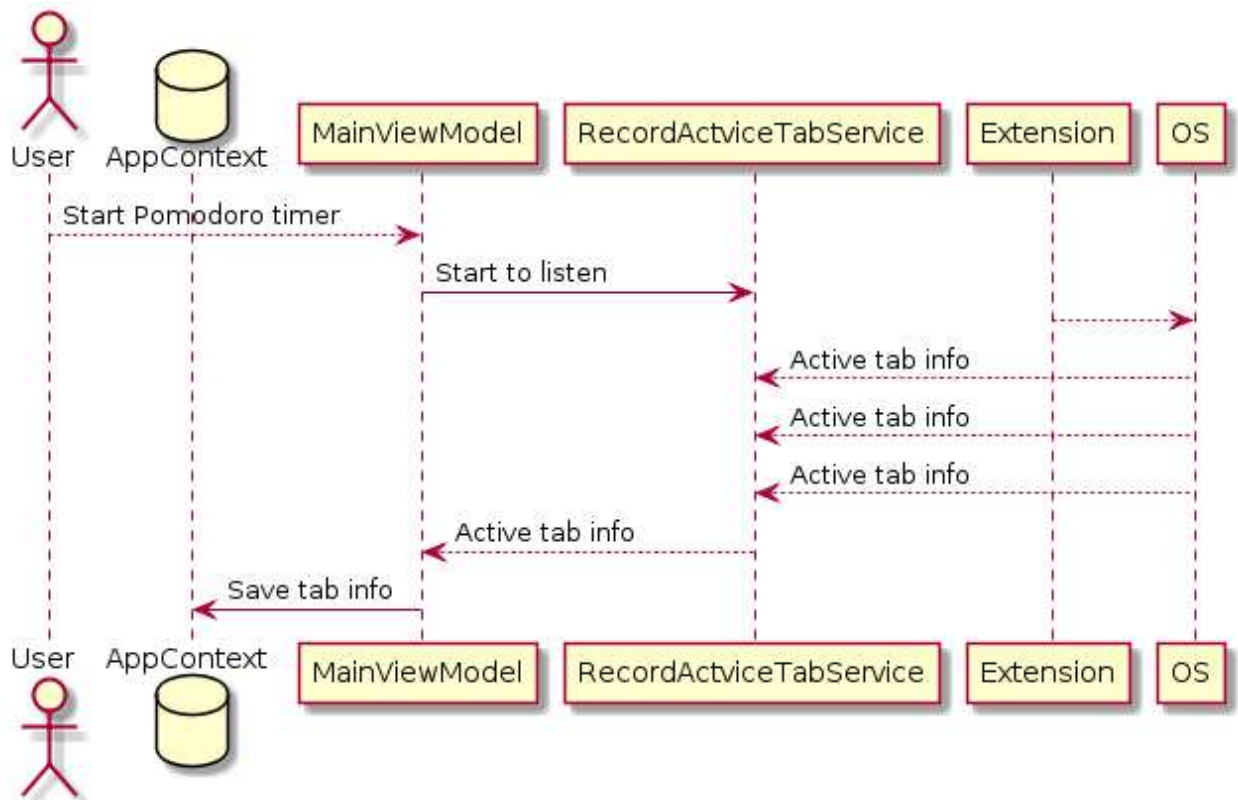


Рисунок 14 – Алгоритм RecordActiveTabService

3.6 Выводы по разделу

В данной главе была рассмотрена структура приложения, сущности для хранения в базе данных. Рассмотрены конкретные элементы модулей View, ViewModel, Model. Рассмотрены основные алгоритмы приложения.

4 Руководство пользователя

4.1 Установка и запуск

Для использования приложения необходимо запустить исполняемый файл и добавить расширение в Google Chrome. Также требуется версия операционной системы не ниже Windows 7 и .NET Framework 4.7.2.

4.2 Демонстрация возможностей программы

После запуска отображается главное меню (рисунок 15) на котором расположено меню настроек, панель для замера времени, состоящая из поля ввода названия для временного периода, таймера и кнопки Start/Stop.

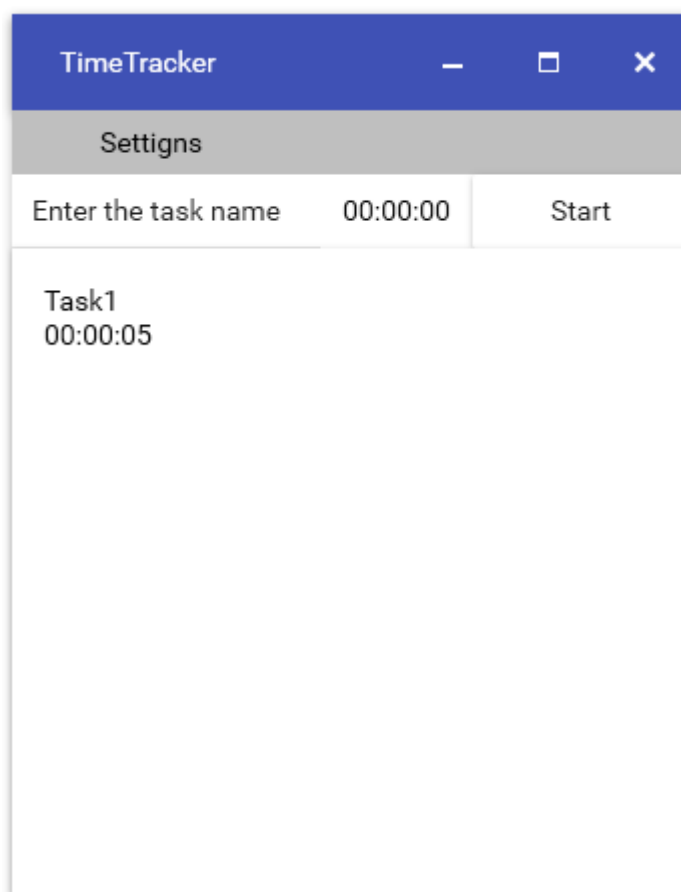


Рисунок 15 – Главное окно

В списке (рисунок 16) расположены замеренные интервалы, по клику открывается окно деталей, которое содержит список приложений и вкладок.

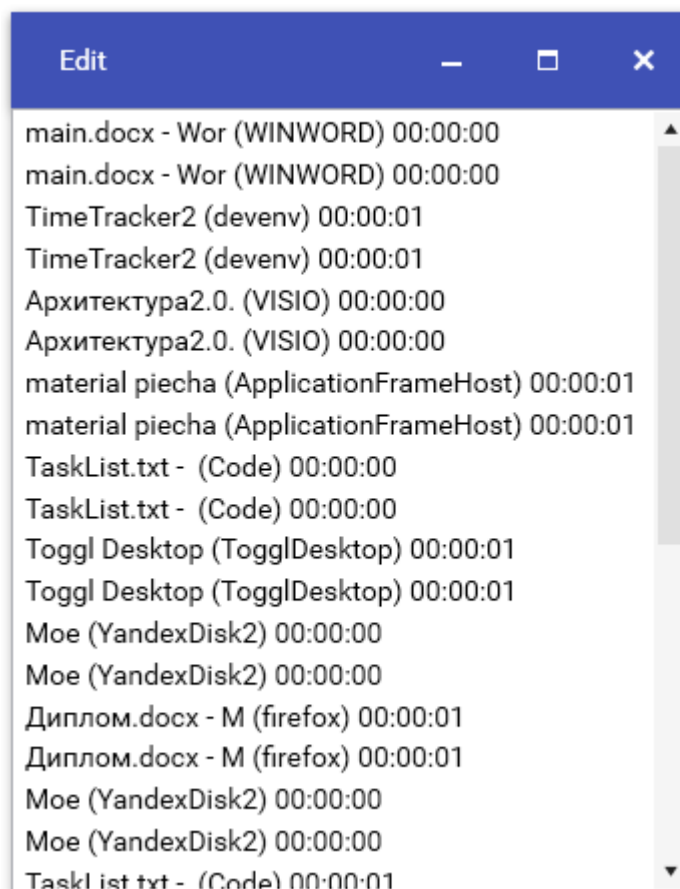


Рисунок 16 – Окно деталей

С главного окна можно попасть в окно настроек таймера (рисунок 17).

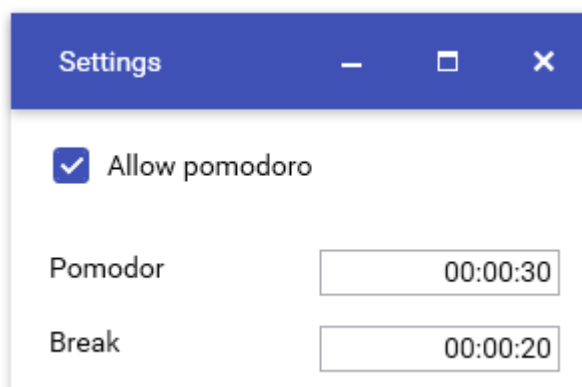


Рисунок 17 – Окно настроек

На рисунке 18 показано окно отчета, на котором можно выбрать необходимый диапазон дат и времени.

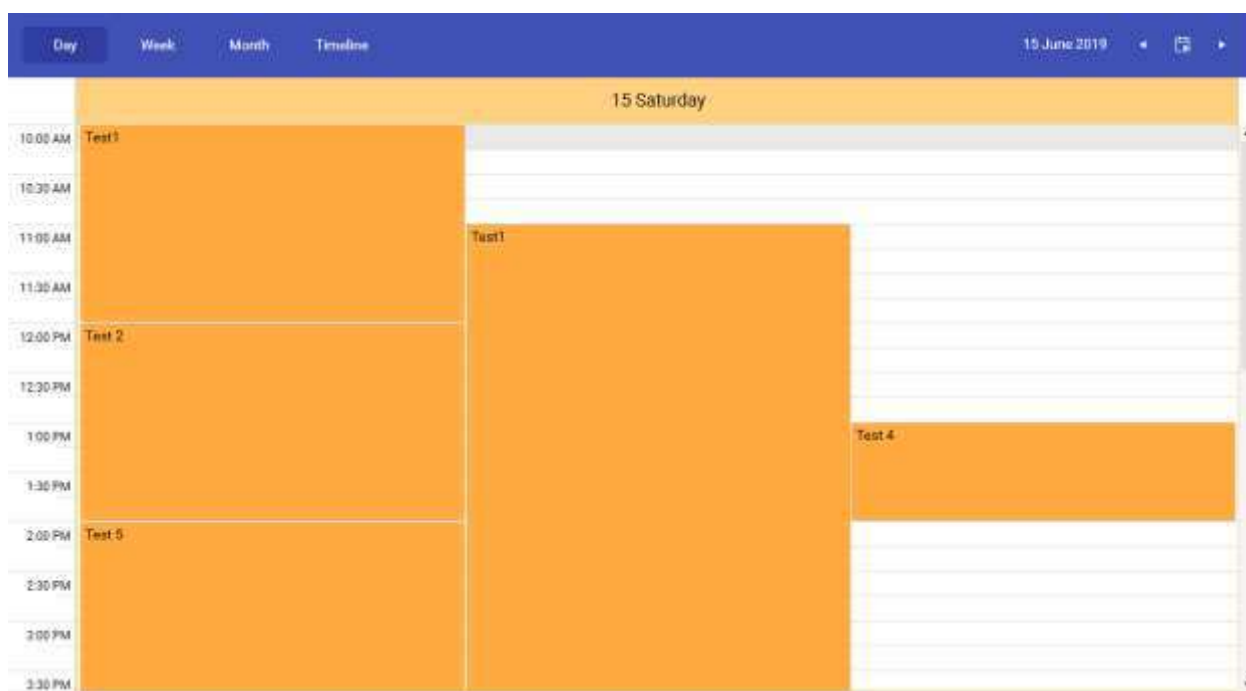


Рисунок 18 – Окно отчета

На рисунке 19 показано уведомление, всплывающее при активности приложений из черного списка.

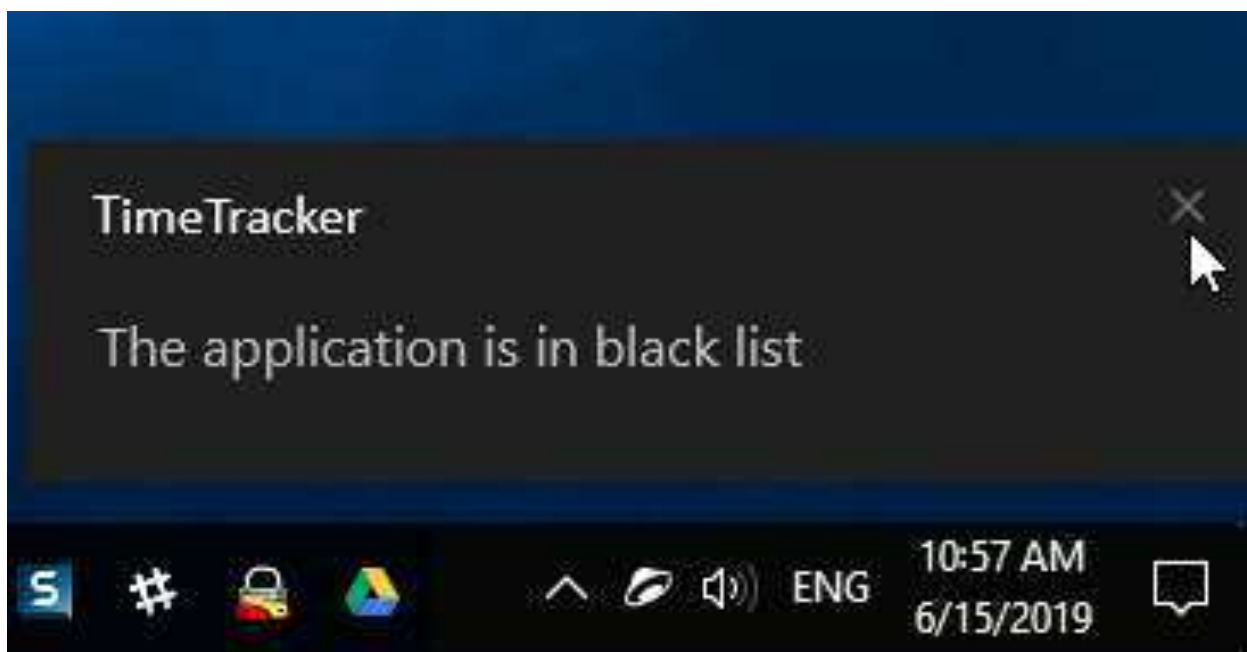


Рисунок 19 – Push-уведомление

ЗАКЛЮЧЕНИЕ

В ходе выполнения бакалаврской работы было разработано приложение под операционную систему Windows, предоставляющее возможность учета и контроля времени, отслеживания приложений и вкладок браузера, создавать профили с черными списками приложений/вкладок, а также возможность создавать отчеты.

Были реализованы все поставленные задачи, а именно: были изучены и проанализированы аналоги данного приложения, выявлены их особенности и недостатки, с помощью этого были поставлены требования к разрабатываемому продукту.

Для разработки приложения под ОС Windows была использована платформа .NET Framework 4.7.2, язык C#. ORM Entity Framework, SQLite, также IoC контейнер Autofac.

СПИСОК СОКРАЩЕНИЙ

API (Application Programming Interface) – программный интерфейс приложения

CSV (Comma Separated Values) – текстовый формат, предназначенный для представления табличных данных

MVVM (Model-View-ViewModel) – архитектурный паттерн построения приложений

ORM (Object-Relational Mapping) – технология, позволяющая связывать базы данных с концепциями объектно-ориентированного программирования

PDF (Portable Document Format) – межплатформенный открытый формат электронных документов

URL (Uniform Resource Locator) – систему унифицированных адресов электронных ресурсов

WPF (Windows Presentation Foundation) – система для построения клиентских приложений

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Сервис учета времени Toggl [Электронный ресурс] : – Режим доступа: <https://toggl.com/>.
2. Техника Pomodoro [Электронный ресурс] : – Режим доступа: <https://francescocirillo.com/pages/pomodoro-technique>.
3. Сервис учета времени PomоDone [Электронный ресурс] : – Режим доступа: <https://pomodoneapp.com/>.
4. Сервис учета времени TMetric [Электронный ресурс] : – Режим доступа: <https://tmetric.com/>.
5. Сервис учета времени RescueTime [Электронный ресурс] : – Режим доступа: <https://rescuetime.com/>.
6. .NET Framework [Электронный ресурс] : – Режим доступа: <https://www.microsoft.com/ru-RU/download/details.aspx?id=17851>.
7. WPF [Электронный ресурс] : – Режим доступа: <https://docs.microsoft.com/ru-ru/visualstudio/designers/getting-started-with-wpf?view=vs-2019>.
8. C# 7.0 [Электронный ресурс] : – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/whats-new/csharp-7>.
9. Entity Framework [Электронный ресурс] : – Режим доступа: <https://docs.microsoft.com/ru-ru/ef/>.
10. SQLite [Электронный ресурс] : – Режим доступа: <https://www.SQLite.org/download.html>.
11. Autofac [Электронный ресурс] : – Режим доступа: <https://autofac.org/>.
12. Visual Studio 2017 Community [Электронный ресурс] : – Режим доступа: <https://visualstudio.microsoft.com/ru/vs/community/>.
13. Архитектурный паттерн MVVM [Электронный ресурс] : – Режим доступа: <https://metanit.com/sharp/wpf/22.1.php>.

14. Google Chrome API [Электронный ресурс] : – Режим доступа:
https://developer.chrome.com/extensions/api_index.

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой ВТ
О.В. Непомнящий
подпись инициалы, фамилия
« 27 » 06 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код и наименование направления

Система учета и контроля рабочего времени
тема

Руководитель	<u>Иванов</u> 25.06.19 подпись, дата	ст. преподаватель должность, ученая степень	<u>И. В. Матковский</u> инициалы, фамилия
Выпускник	<u>И.В.</u> 25.06.19 подпись, дата		<u>Д. Е. Ломанов</u> инициалы, фамилия
Консультант	<u>И.В.</u> 25.06.19 подпись, дата	доцент, канд. техн. наук должность, ученая степень	<u>Л. И. Покидышева</u> инициалы, фамилия
Нормоконтролер	<u>Иванов</u> 26.06.19 подпись, дата	доцент, канд. техн. наук должность, ученая степень	<u>В. И. Иванов</u> инициалы, фамилия

Красноярск 2019