Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Кубанский государственный технологический университет»

(ФГБОУ ВО КубГТУ)

Институт Компьютерных систем и информационной безопасности

Кафедра Информационных систем и программирования

Специальность 10.05.03 Информационная безопасность автоматизированных систем

Профиль Защищенные автоматизированные системы управления

**КУРСОВАЯ РАБОТА**

по дисциплине Технологии и методы программирования

(наименование дисциплины)

на тему: «Табло на станции метро»

(тема курсовой работы)

Выполнил студент 2 курса группы 18-К-АС1

Гереханов Р.И.

(Ф.И.О.)

Допущен к защите\_\_\_\_\_\_\_

Руководитель (нормоконтролер) работы \_\_\_\_\_\_\_\_\_О.Б. Попова

Защищен \_\_\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_\_\_\_

(дата)

Члены комиссии Н.В. Кушнир\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

К.Е. Тотухов\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Краснодар

2020

ФГБОУ ВО «Кубанский государственный технологический университет»

(ФГБОУ ВО КубГТУ)

Институт Компьютерных систем и информационной безопасности

Кафедра Информационных систем и программирования

Специальность 10.05.03 Информационная безопасность автоматизированных систем

Профиль Защищенные автоматизированные системы управления

УТВЕРЖДАЮ

Зав. кафедрой \_\_\_\_\_\_\_\_\_М.В. Янаева

«12» февраля 2020 г.

**ЗАДАНИЕ**

на курсовую работу

Студенту: Гереханову Р.И. группы 18-К-АС1 курса 2

(Ф.И.О.) (№ группы и курса)

Тема проекта: «Табло на станции метро»

План работы:

1. Изучение предметной области

2. Проектирование

3.  Описание реализованных диаграмм

Объем работы:

а) пояснительная записка 29 с.

Рекомендуемая литература

1.  Йордон. «Объектно-ориентированный анализ и проектирование систем»

2.  Роберт А. Максимчук. «UML для простых смертных»

3.  «Автоматизация проектирования вычислительных систем.» ред. М.Брейер

Срок выполнения: с «15» февраля по «11» мая 2020г.

Срок защиты: с «11» мая по «14» июня 2020 г.

Дата выдачи задания «15» февраля 2020г.

Дата сдачи работы на кафедру «01» июня 2020 г.

Руководитель работы \_\_\_\_\_\_\_\_\_ к.т.н., доцент Попова О.Б.

(должность, подпись,)

Задание принял студент Гереханов Р.И. Ф.И.О.

**Реферат**

Курсовая работа: 29 страница, 16 рисунков, 9 используемых источников.

Ключевые слова: ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ПРОЕКТИРОВАНИЕ, МОДЕЛЬ, КЛАСС, ТАБЛО, МЕТРО, UML, BPMN, ГАНТ, EPC, FURPS+, IDEF0, DFD, ДИАГРАММЫ.

Объектом исследования является программное обеспечение табло на станции метро, которое способно отображать текущее время, момент прибытия и отправления поезда, отображение конечной станции прибывающего поезда, а также отображать рекламные сообщения.

Цель работы состоит в разработке проекта программного обеспечения «Табло на станции метро» с использованием диаграмм разного вида, в полной мере описывающих как внутреннее устройство исследуемой системы, так и всевозможные взаимодействия между её компонентами.

В результате были получены диаграммы, обладающие исчерпывающей информацией о программном обеспечение домофона. К ним относятся: диаграмма Ганта, UML-диаграмма, IDEF0-диаграмма, DFD-диаграмма, EPC-диаграмма, BPMN «As-Is» и BPMN «To be», документ FURPS+.

**Содержание**

[Введение 5](#_Toc43399887)

[2 Диаграмма Ганта 7](#_Toc43399894)

[3 Создание модели As-Is в стандарте IDEF0 8](#_Toc43399895)

[4 Диаграмма потоков данных (DFD) 13](#_Toc43399896)

[5 UML 14](#_Toc43399897)

[6 Проектирование диаграммы вариантов использования. 15](#_Toc43399898)

[7 Проектирование диаграммы последовательности. 16](#_Toc43399899)

[8 EPC 19](#_Toc43399900)

[9 BPMN 20](#_Toc43399901)

[10 FURPS+ 22](#_Toc43399902)

[11 Тестирование программы 23](#_Toc43399903)

[Заключение 26](#_Toc43399904)

[Приложение А – Проверка на антиплагиат 28](#_Toc43399905)

[Приложение Б – Диаграмма Ганта 29](#_Toc43399906)

**Введение**

В настоящее время табло на любом метрополитене имеет большую важность для абсолютно всех жителей городов. Современное табло на станции метро должно содержать всю ту информацию, которая позволит абсолютно любому гражданину – от мала до велика, легко ориентироваться в данном общественном транспорте. С течением времени требования к табло изменились не сильно, оно по-прежнему должно содержать расписание, время прибытия/отбытия поездов и ничего лишнего, что могло бы сильно отвлекать внимание. Применение табло в различных метрополитенах стало неотъемлемой частью человеческой жизни.

Однако, несмотря на повсеместное использование данной технологии, она никогда не смогла бы обеспечить надлежавший уровень без тщательного исследования предметной области и проведения различных тестов, учитывающих всевозможные взаимодействия с компонентами системы.

Таким образом, в данной курсовой работе будут проектироваться специальные диаграммы, позволяющие в полной мере раскрыть все аспекты исследуемой системы. В результате табло, созданное на основании всех проведённых исследований, будет соответствовать основным требованиям для обеспечения надлежащего уровня функционирования табло на станции метро.

**1 Формулировка задачи**

# Программное обеспечение табло для информационной службы метрополитена.

# Табло расположены на каждой станции метро. Они работают под управлением единого пункта управления (ПУ) информационной службы метро. Табло отображает текущее время (часы, минуты, секунды) и время, прошедшее с момента отправления последнего поезда (минуты, секунды). Момент прибытия и отправления поезда определяется при помощи датчиков, устанавливаемых на путях. Все табло метро синхронизованы, текущее время отсчитывается и устанавливается из центральной службы времени, находящейся на ПУ.

# На табло высвечивается конечная станция назначения прибывающего поезда. Эти данные содержатся в расписании движения поездов, которое хранится в памяти табло и периодически обновляется с ПУ.

# В «бегущей строке» табло отображается рекламная информация. Память табло хранит до 10 рекламных сообщений. Сообщения отображаются друг за другом с небольшими паузами, циклически. Содержание рекламных сообщений поступает с ПУ.

# Дополнительная функция табло – по запросу с ПУ оно пересылает данные о нарушениях расписания (преждевременных отправлениях поездов или опозданиях).

# 2 Диаграмма Ганта

Диаграмма Ганта — это горизонтальная столбчатая диаграмма с временной шкалой, которая используется для иллюстрации плана работ по проекту с привязкой ко времени. Ее придумал американский инженер Генри Гант в 1910 году, и в течение 20 века она широко использовалась для планирования проектов. Задачам, перечисленным слева от диаграммы, соответствуют ленты, ориентированные вдоль оси времени. Эти ленты изображают рабочий процесс в проекте. Диаграммы Ганта обычно содержат даты начала и завершения задач, контрольные точки, зависимости между задачами и исполнителей.

С помощью диаграмм Ганта руководители проектов наглядно изображают составляющие части проекта и разбивают большой, цельный проект на задачи меньшего размера для удобства управления. Получившиеся задачи размещаются вдоль временной шкалы диаграммы Ганта, после чего на нее добавляют зависимости между задачами, исполнителей и контрольные точки. На этапе реализации больших, упорядоченных и детально проработанных проектов рабочий процесс и ход работы благодаря диаграммам Ганта становятся наглядными. Эти диаграммы обеспечивают согласованность между командами: у всех перед глазами одни цели и для всех установлены одни сроки. Главным преимуществом рассматриваемой диаграммы является способность упрощать составные проекты. С помощью этого средства можно весьма наглядно и удобно для обобщения представить большое количество данных.

Диаграмма Ганта для проекта «Табло на станции метро» находится в «Приложении Б».

**3 Создание модели As-Is в стандарте IDEF0**

Чтобы оценить возможности, разрабатываемой системы необходимо построить её базовую модель, которую можно представить в виде диаграммы As-Is.

Диаграмма As-Is – это функциональная модель системы «как есть», позволяющая узнать где находятся слабые места, в чём будут состоять преимущества и недостатки, протекающих в ней бизнес-процессов относительно конкурентов. Применение данной модели позволит чётко зафиксировать какие информационные объекты принимают участие в жизненном цикле системы, какая информация будет поступать на вход и что будет получаться на выходе. Модель As-Is, строится с использованием нотации IDEF0.

IDEF0 – это графическая нотация, предназначенная для описания бизнес-процессов. Система, описываемая в данной нотации, проходит через декомпозицию или, иными словами, разбиение на взаимосвязанные функции. Для каждой функции существует правило сторон:

– стрелкой слева обозначаются входные данные;

– стрелкой сверху – управление;

– стрелкой справа – выходные данные;

– стрелкой снизу – механизм.

Учитывая всё вышеперечисленное на рисунке 1 была составлена модель As-Is проекта «Табло на станции метро».

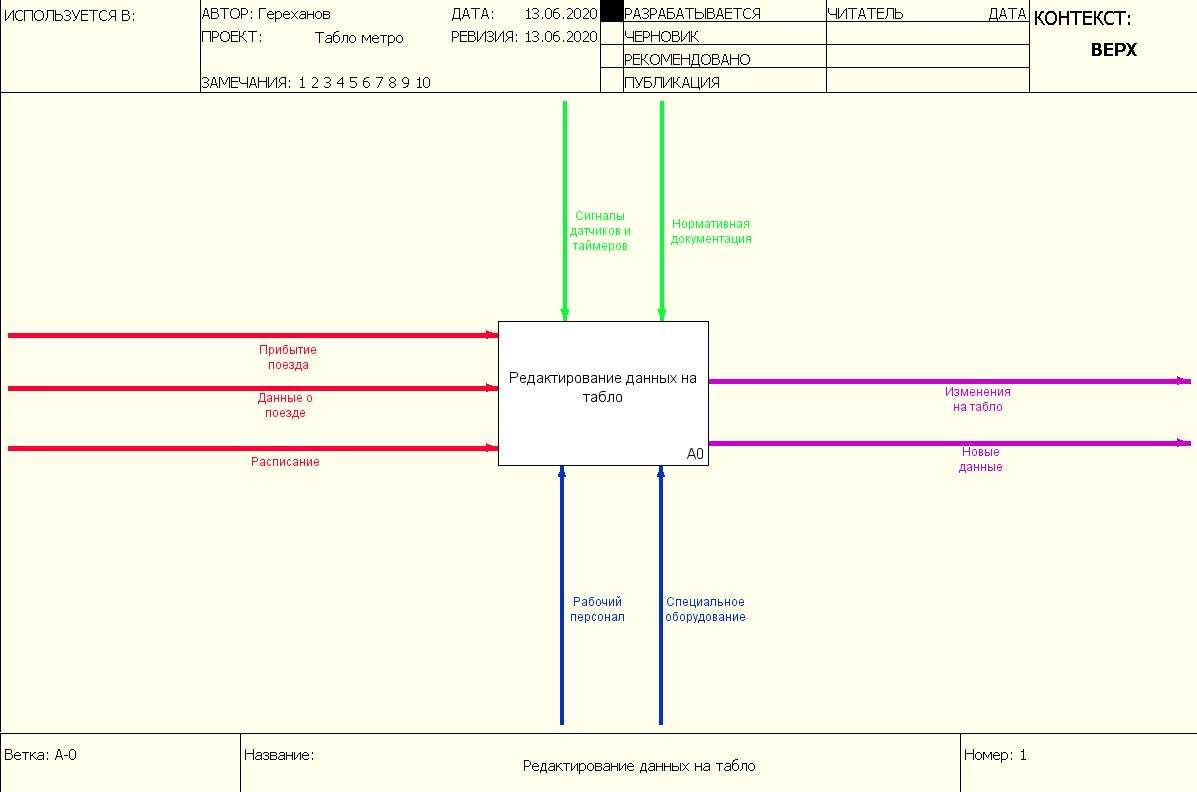


Рисунок 1 – Модель проекта «Табло на станции метро»

Входными данными для данной системы является прибытие поезда, данные о поезде, а также расписание.

Управление происходит благодаря всевозможной нормативной документации и сигналам датчиков и таймеров.

Механизмом реализации работы системы являются рабочий персонал и специальное оборудование.

Результатом деятельности системы являются изменения на табло и измененные данные.

Полученная модель системы может быть представлена в более подробном виде путём разбиения на большее количество составных элементов.

На рисунке 2 можно видеть модель системы после декомпозиции.

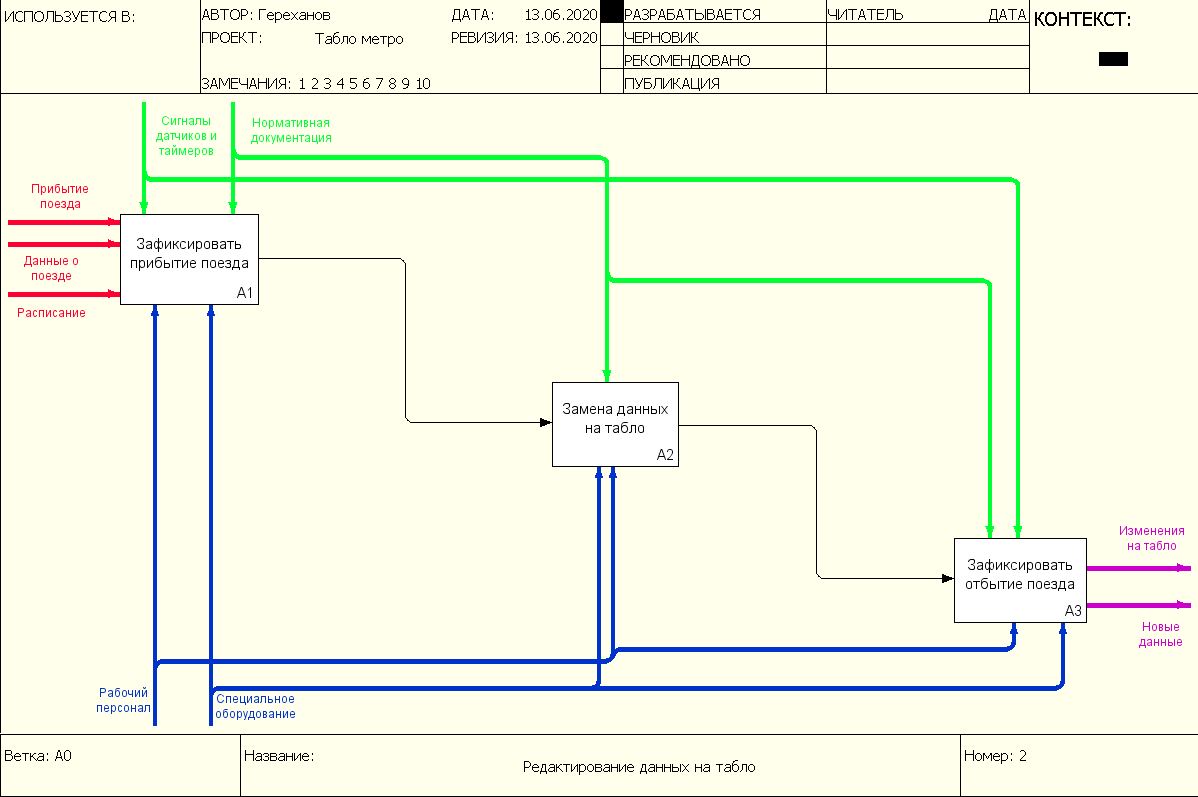


Рисунок 2 – Декомпозиция проекта «Табло на станции метро»

Работа рассматриваемой системы разбивается на три основные функции:

– зафиксировать прибытие поезда;

– замена данных на табло;

– зафиксировать отбытие поезда.

Полученная модель системы может быть представлена в более подробном виде путём разбиения на большее количество составных элементов.

На рисунке 3 можно видеть декомпозицию фиксации прибытия поезда.

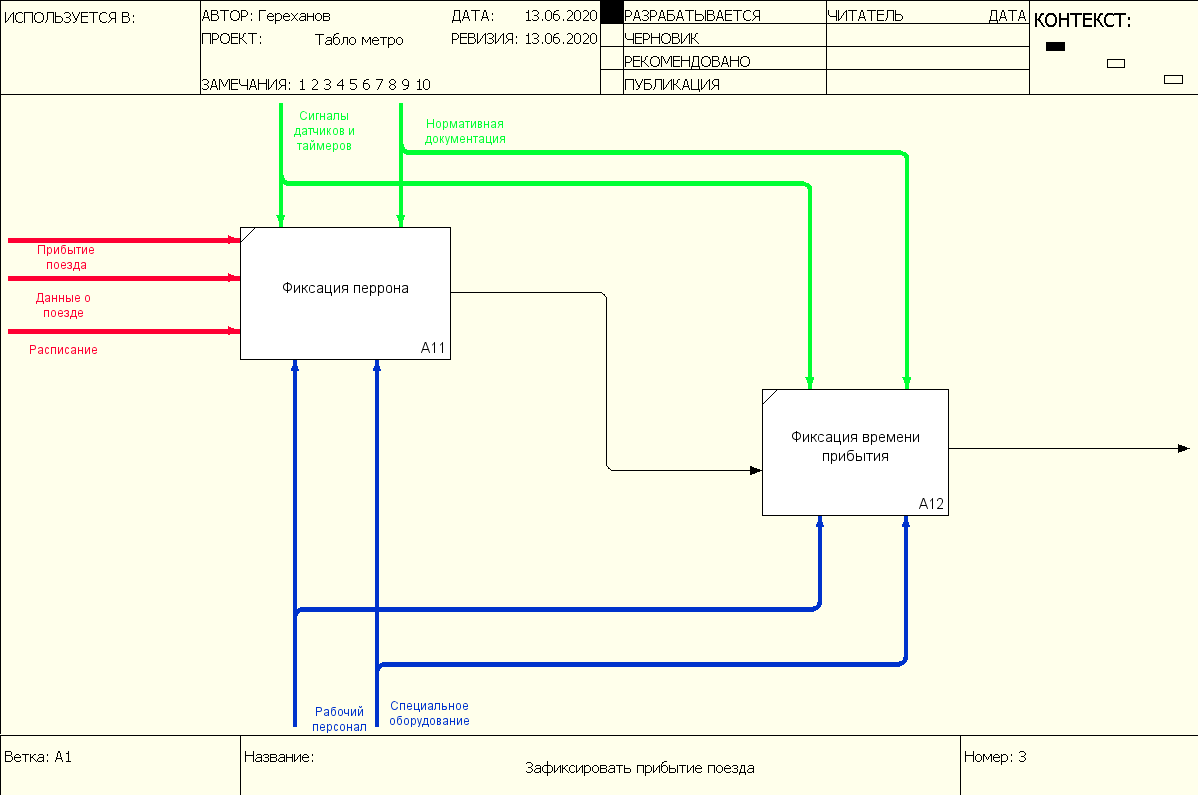


Рисунок 3 – Декомпозиция обработки заказа клиента

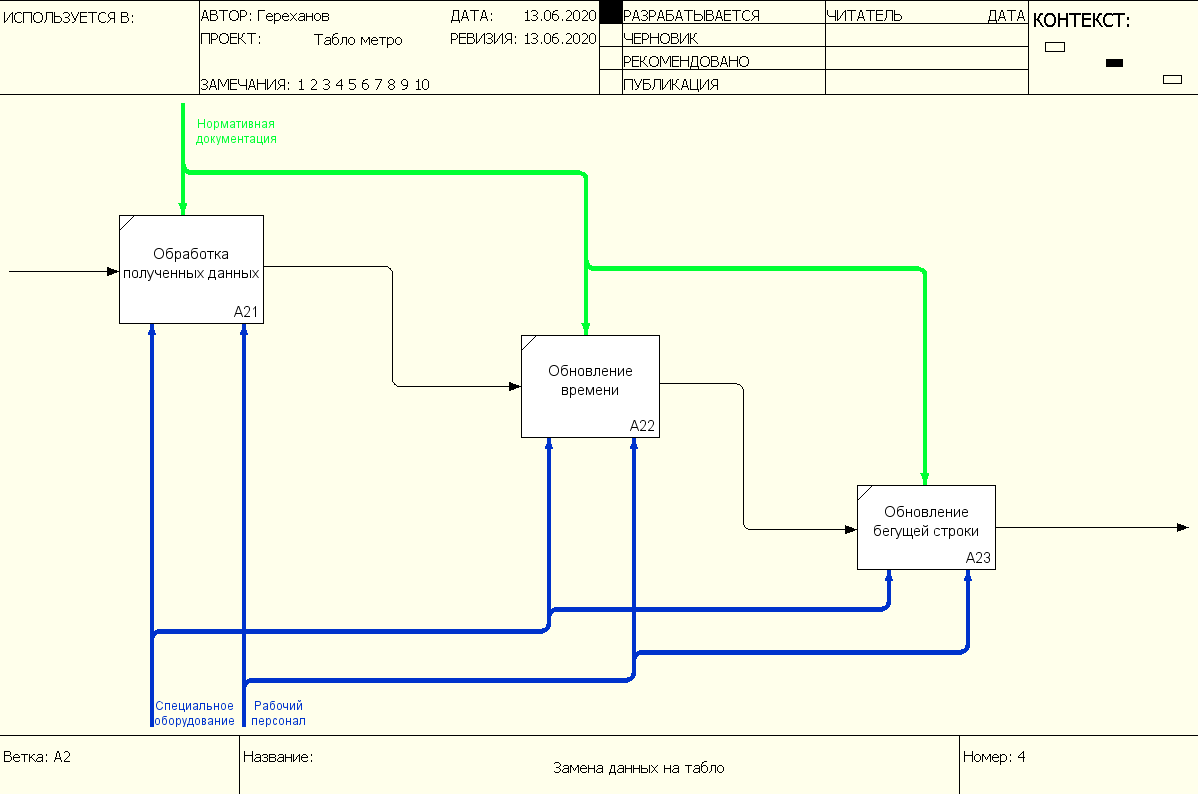


Рисунок 4 – Декомпозиция замены данных на табло

На рисунке 5 можно видеть декомпозицию фиксации отбытия поезда.

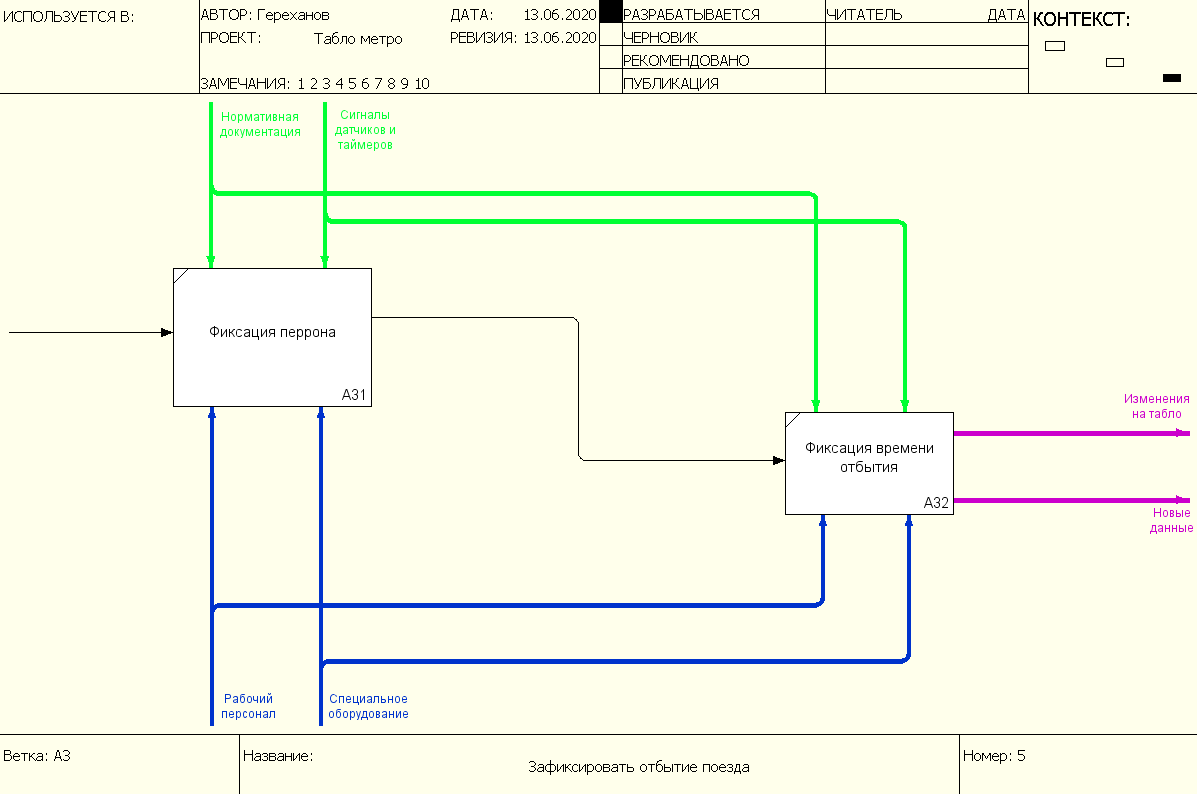


Рисунок 5 – Декомпозиция подготовки документации

**4 Диаграмма потоков данных (DFD)**

Диаграмма потоков данных DFD (DataFlowDiagrams) – «это методология графического структурного анализа, описывающая внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ. Диаграмма DFD – это один из основных инструментов структурного анализа и проектирования информационных систем, существовавших до широкого распространения UML.»

В результате декомпозиции системы «Табло на станции метро» была получена следующая диаграмма DFD (рис. 6).

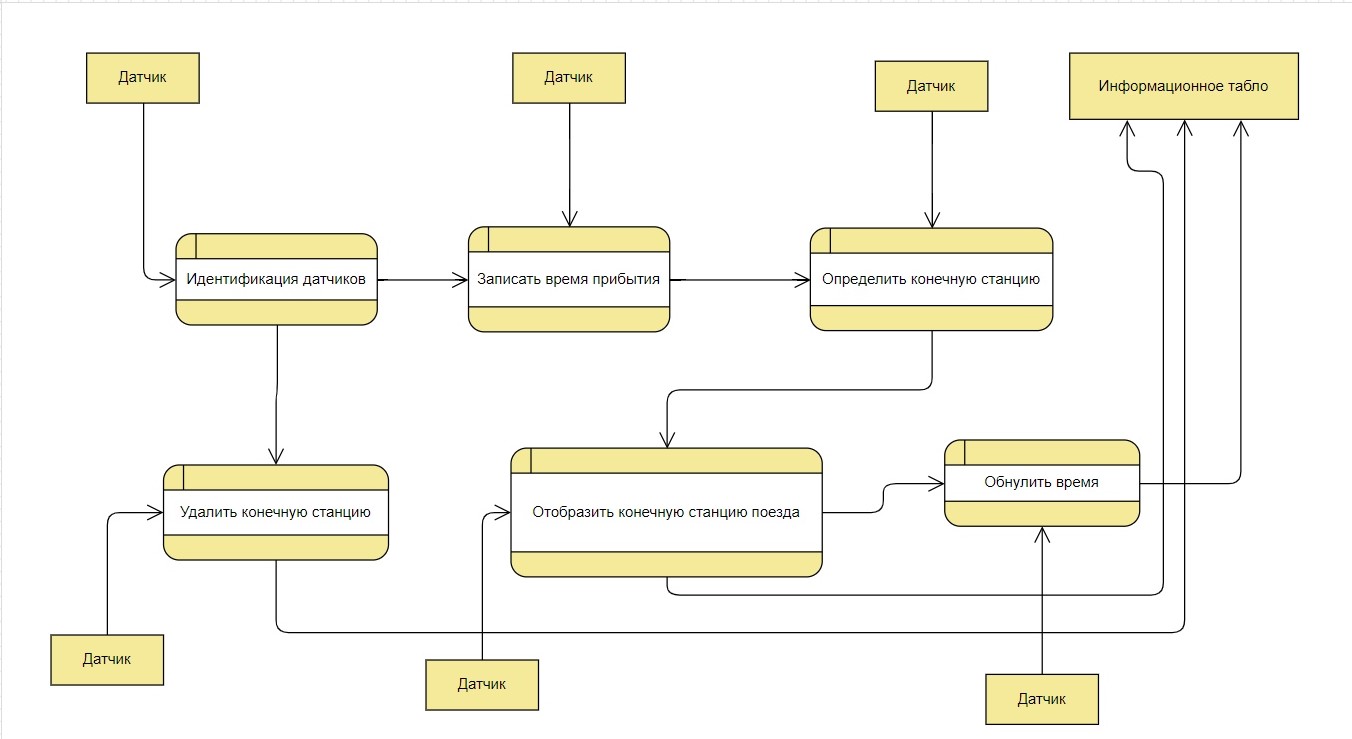


Рисунок 6 – Диаграмма DFD-системы «Табло на станции метро»

Система содержит процессы: идентификация датчиков, запись времени прибытия поезда, определение конечной станции, отображение конечной станции поезда, удаление конечной станции, обнуление времени.

**5 UML**

UML (англ. Unified Modeling Language — «унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

UML является языком широкого профиля, это — открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода.

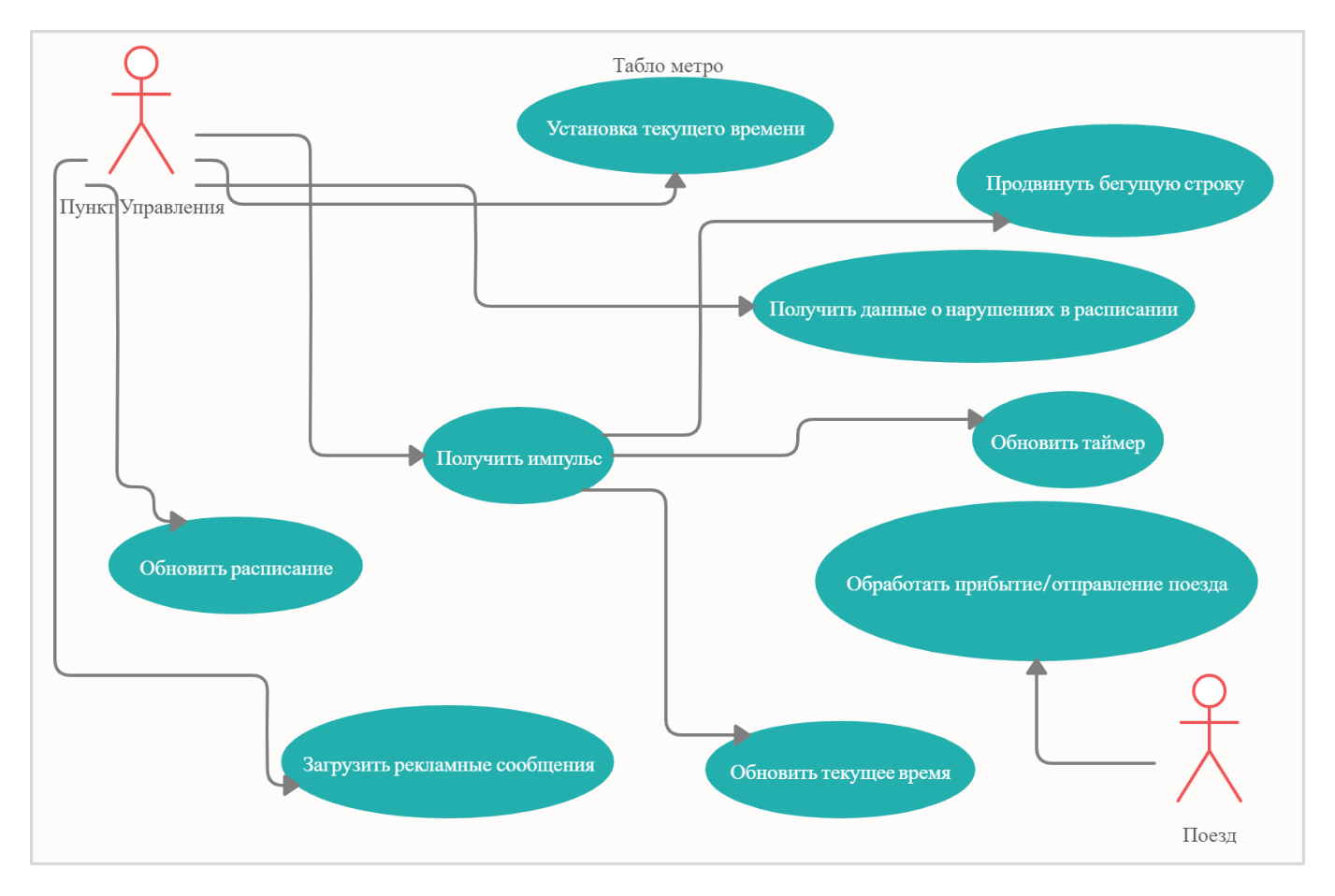


Рисунок 7 – UML-диаграмма системы «Табло на станции метро»

**6 Проектирование диаграммы вариантов использования.**

Диаграмма вариантов использования описывает функциональные требования системы с точки зрения способов её применения. Данная модель позволяет связать то, что нужно разработчикам от системы, с тем, как система удовлетворяет эти потребности. Данная диаграмма должна состоять из актеров, вариантов использования и связей между ними.

Ее суть состоит в том, что система представляется в виде актеров, которые взаимодействуют с системой. Они могут взаимодействовать исходя из различных вариантов использования. Актером может быть любой объект, который может взаимодействовать в исследуемой системой извне. Вариант использования – это набор функций, которая система может предоставить актеру.

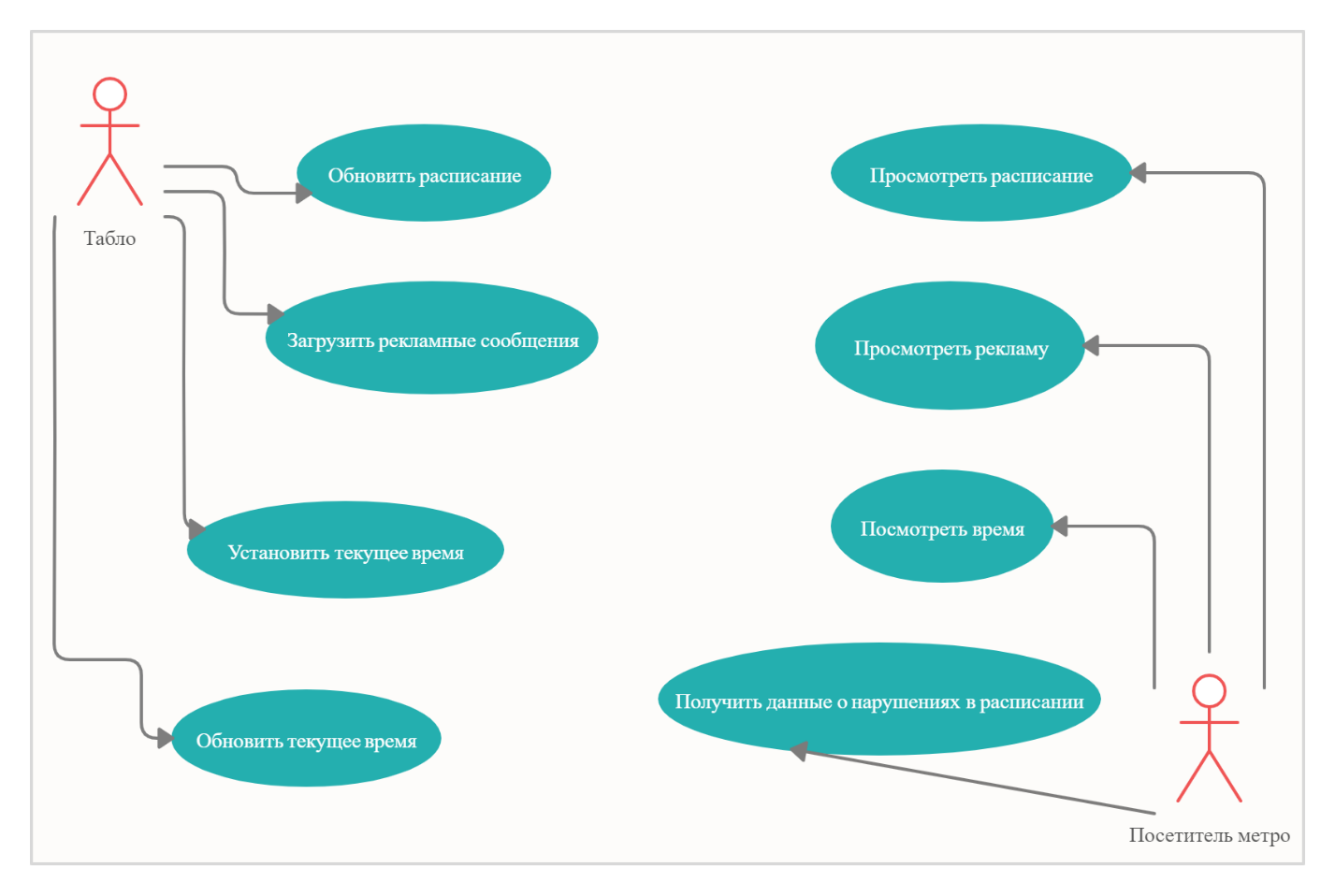


Рисунок 8 – Диаграмма вариантов использования

**7 Проектирование диаграммы последовательности.**

Диаграмма последовательности отражает взаимодействие определенного набора объектов на некоторой временной оси.

Основными ее элементами являются обозначения объектов, линии жизни объектов и стрелки, показывающие обмен сигналами или сообщениями между ними.

На этой диаграмме изображаются только те объекты, которые непосредственно участвуют во взаимодействии. Для диаграммы последовательности ключевым моментом можно назвать именно динамику взаимодействия объектов во времени. Графически каждый объект изображается прямоугольником и располагается в верхней части своей линии жизни.

Линия жизни объекта изображается пунктирной вертикальной линией. Она служит для обозначения периода времени, в течение которого объект существует в системе. Отдельные объекты, выполнив свою роль в системе, могут быть уничтожены путем обрыва линии жизни.

Ниже представлены основные диаграммы последовательности для проекта «Табло на станции метро».

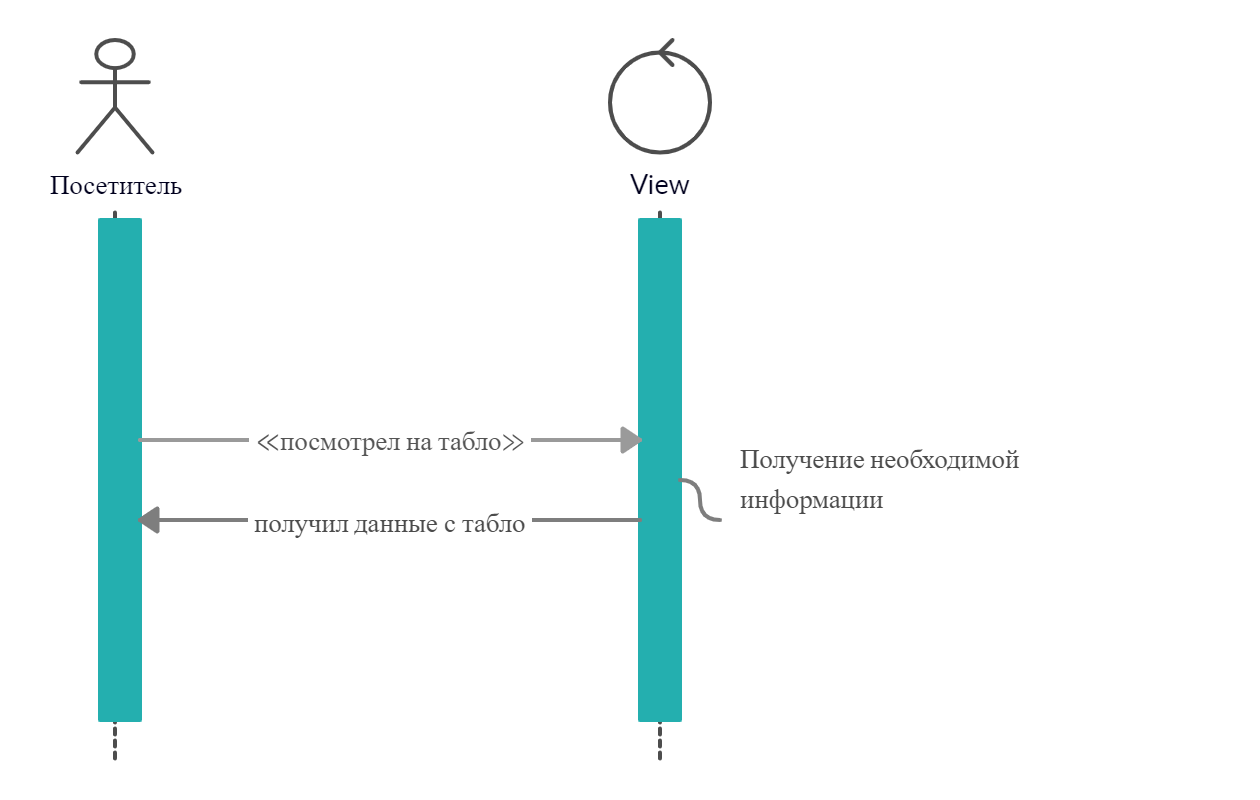


Рисунок 9 – Диаграмма последовательности - просмотр табло

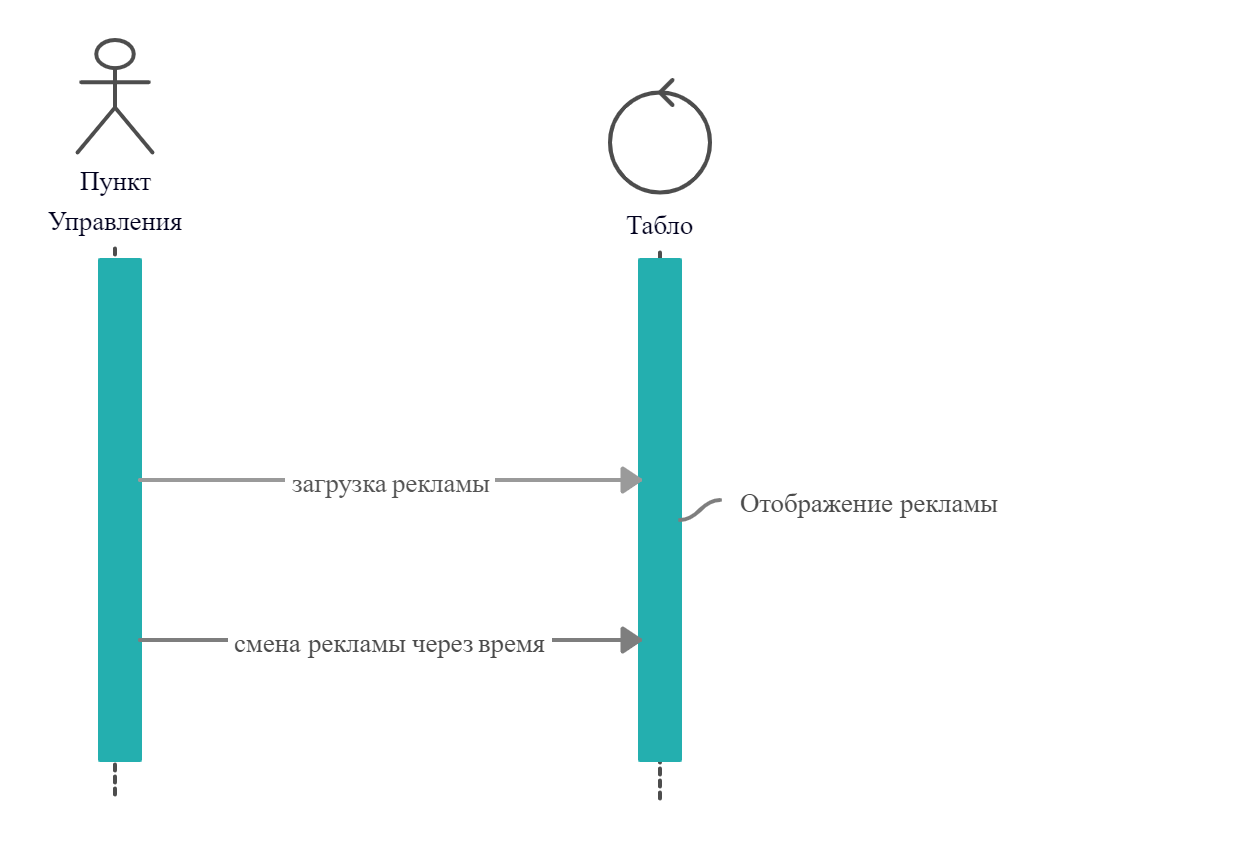


Рисунок 10 – Диаграмма последовательности - реклама

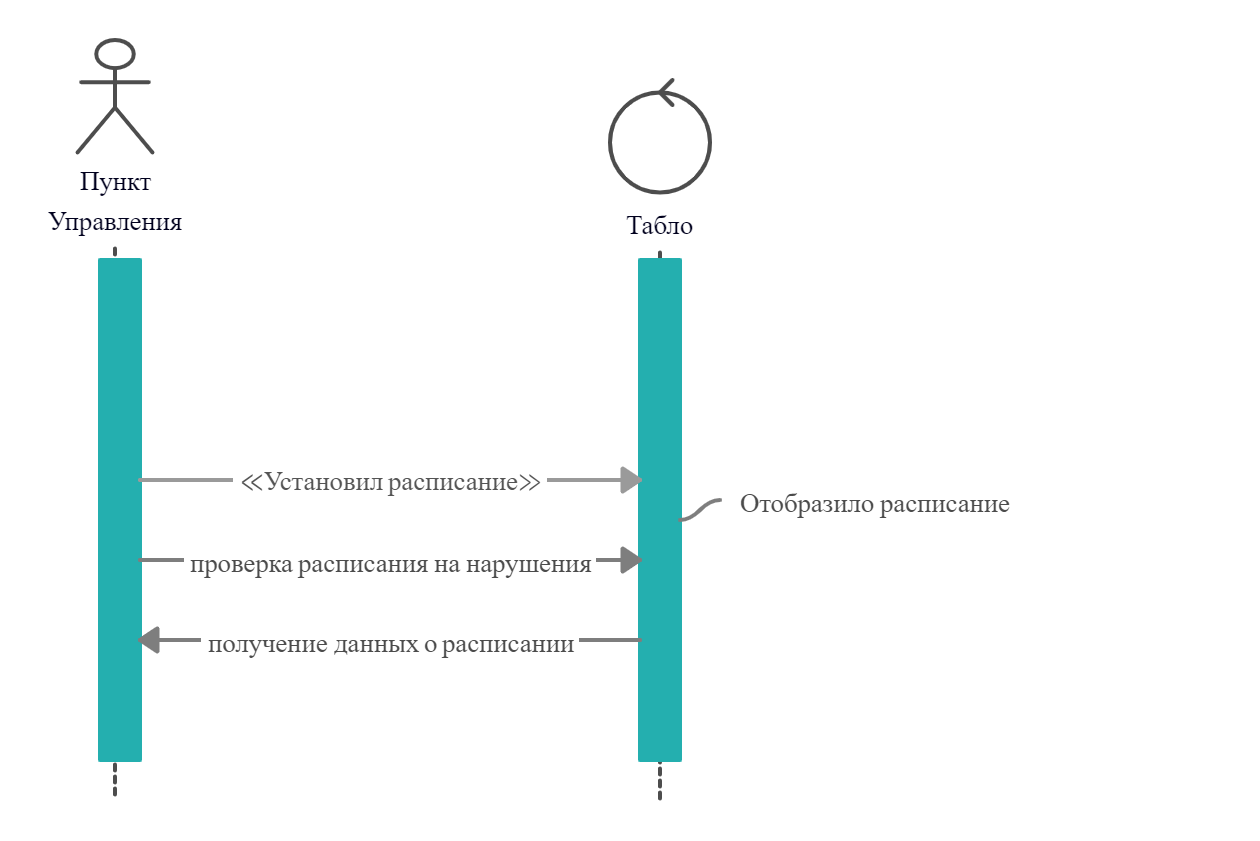


Рисунок 11 – Диаграмма последовательности - расписание

Таким образом при помощи диаграммы последовательности были описаны цепочки действий, происходящие между объектами системы при всевозможных развитиях событий.

# 8 EPC

Событийная цепочка процессов (EPC-диаграмма, англ. event-driven process chain) — тип блок-схемы, используемой для бизнес-моделирования. EPC может быть использована для настройки системы планирования ресурсов предприятия (ERP), и для улучшений бизнес-процессов.

Организации используют EPC-диаграммы для планирования потоков работ бизнес-процессов. Существует ряд инструментов для создания EPC-диаграмм, некоторые из этих средств поддерживают инструменто-независимый формат обмена данными EPC — язык разметки EPML. EPC-диаграммы используют символы нескольких видов, чтобы показать структуру потока управления (последовательность решений, функции, события и другие элементы) бизнес-процесса.

EPC-метод был разработан Августом-Вильгельмом Шеером в рамках работ над созданием ARIS в начале 1990-х годов. Используется многими организациями для моделирования, анализа и реорганизации бизнес-процессов.

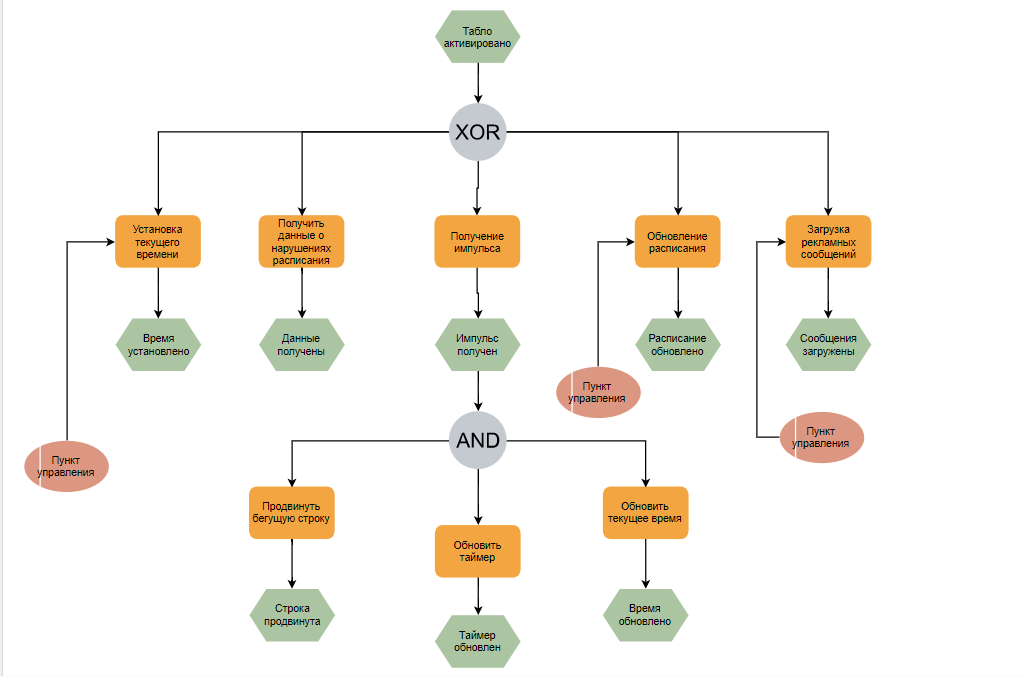


Рисунок 12 – EPC-диаграмма системы «Табло на станции метро»

# 9 BPMN

[Спецификация](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F) BPMN описывает условные обозначения и их описание в XML для отображения [бизнес-процессов](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81) в виде диаграмм бизнес-процессов. BPMN ориентирована как на технических специалистов, так и на бизнес-пользователей. Для этого язык использует базовый набор интуитивно понятных элементов, которые позволяют определять сложные [семантические](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D0%BA%D0%B0_(%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D1%8F)) конструкции. Кроме того, спецификация BPMN определяет, как диаграммы, описывающие бизнес-процесс, могут быть трансформированы в исполняемые модели.

Основная цель BPMN — создание стандартного набора условных обозначений, понятных всем бизнес-пользователям. Бизнес-пользователи включают в себя [бизнес-аналитиков](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D1%82%D0%B8%D0%BA), создающих и улучшающих процессы, [технических разработчиков](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D1%87%D0%B8%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F), ответственных за реализацию процессов и [менеджеров](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%BD%D0%B5%D0%B4%D0%B6%D0%B5%D1%80), следящих за процессами и управляющих ими. Следовательно, BPMN призвана служить связующим звеном между фазой дизайна бизнес-процесса и фазой его реализации.

На рисунке 9 представлена диаграмма BPMN системы табло на станции метро.

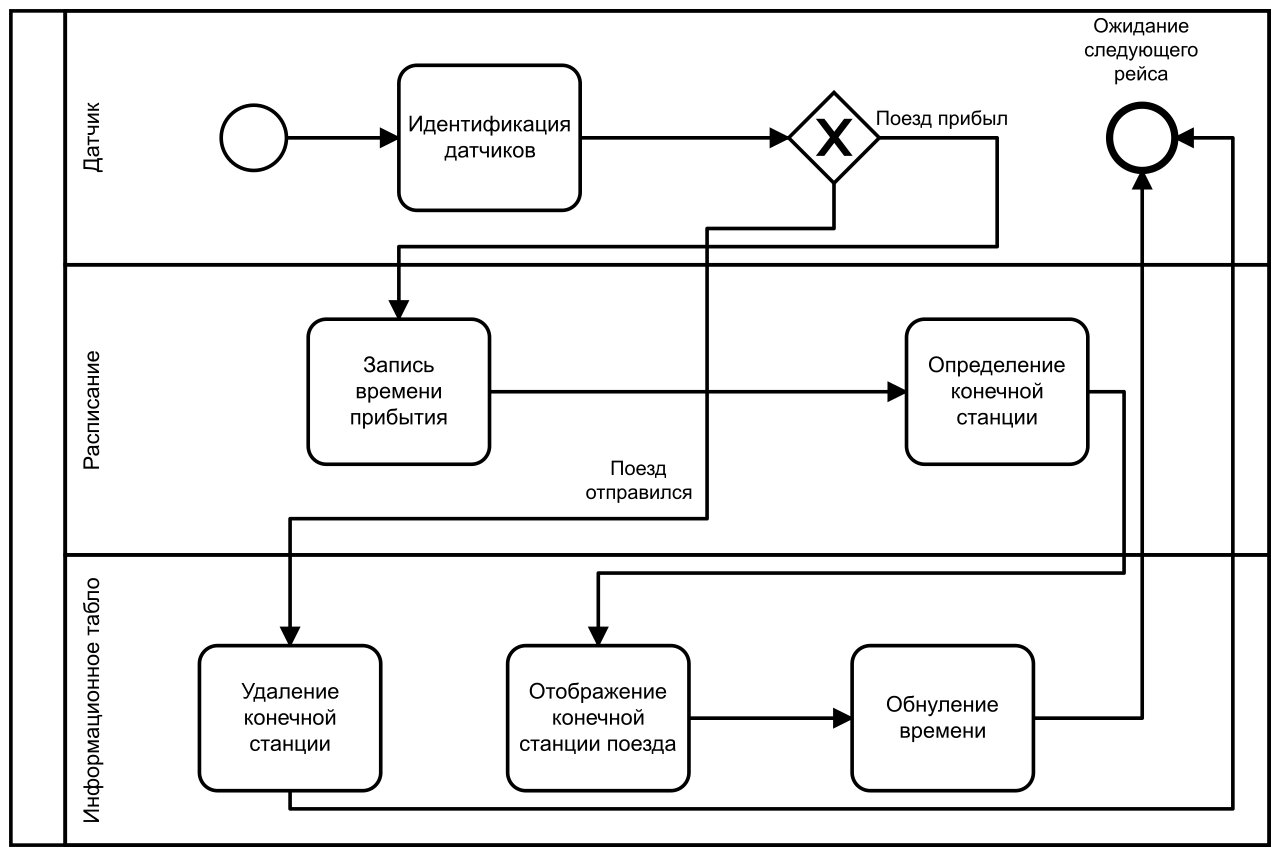


Рисунок 13 – Диаграмма BPMN «As-Is» и «To be»

# 10 FURPS+

Классификация требований к системе FURPS+ была разработана Робертом Грэйди (Robert Grady) из Hewlett-Packard и предложена в 1992 году. Сокращение FURPS расшифровывается так:

* Functionality, функциональность
* Usability, удобство использования
* Reliability, надежность
* Performance, производительность
* Supportability, поддерживаемость

+ необходимо помнить о таких возможных ограничениях, как:

* ограничения проектирования, design
* ограничения разработки, implementation
* ограничения на интерфейсы, interface
* физические ограничения, physical

FURPS+ для системы табло на станции метро:

F – табло метро отображает время прибытия поезда и другую информацию, нужную для комфортного ориентирования посетителей метрополитена.

U – приятный дизайн и понятный для пользователя интерфейс.

R – 1 сбой/1 год; сбой происходит без потери данных.

P – время отклика системы на запрос пользователя не превышает 1 с, 100% эффективность работы.

S – работоспособность системы не зависит от стороннего ПО; легкая инсталляция 1 человеком;

+ - никаких ограничений

# 11 Тестирование программы

Ниже представлено стартовое окно программы. На нем сразу можно узнать текущее время, дату и список станций. Сообщение «Ошибка» означает, что расписание еще не задано пунктом управления.

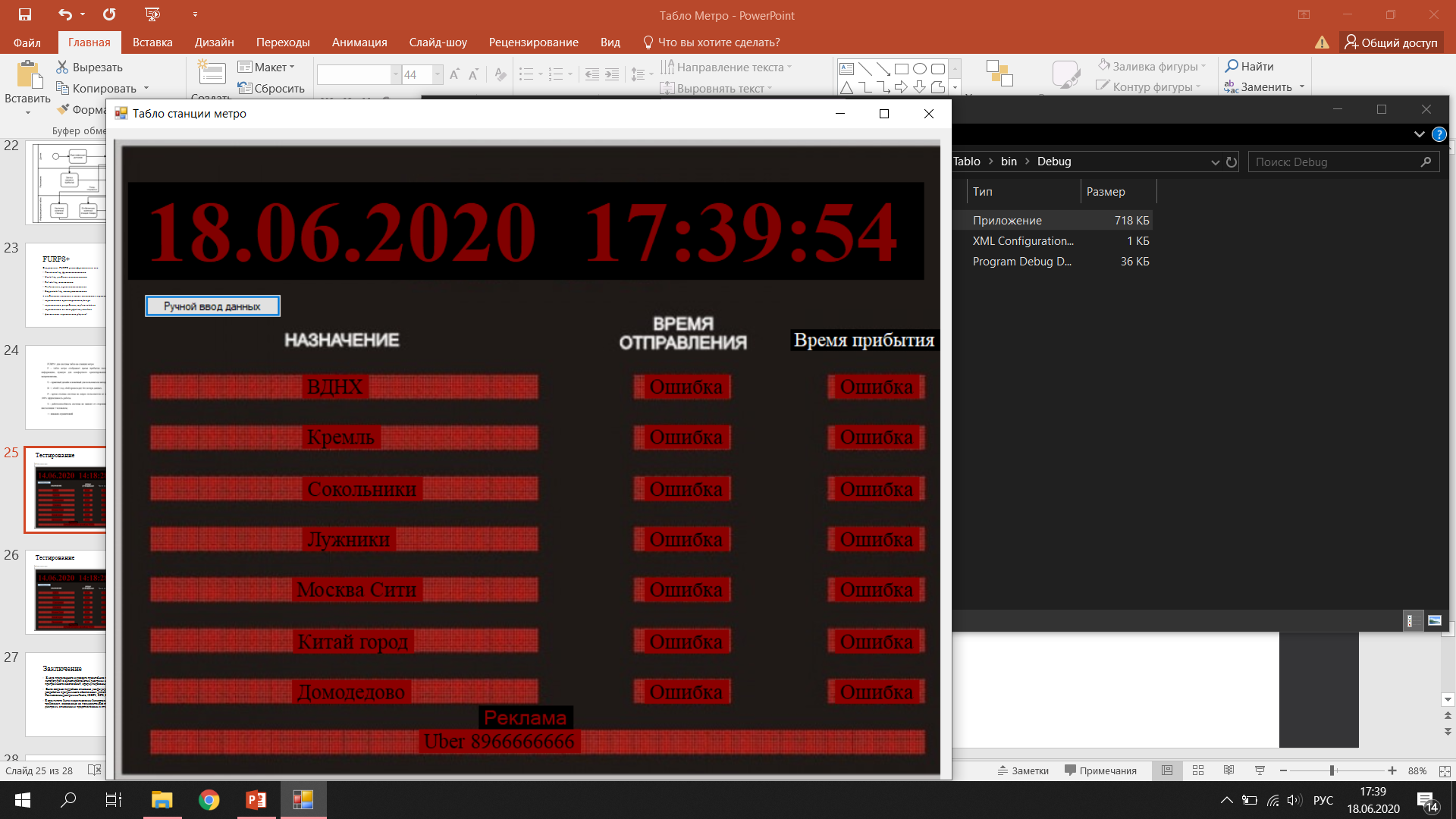


Рисунок 14 – Стартовое окно программы

По нажатию на клавишу «Ручной ввод данных» появляются соответствующие поля для заполнения времени движения поездов.

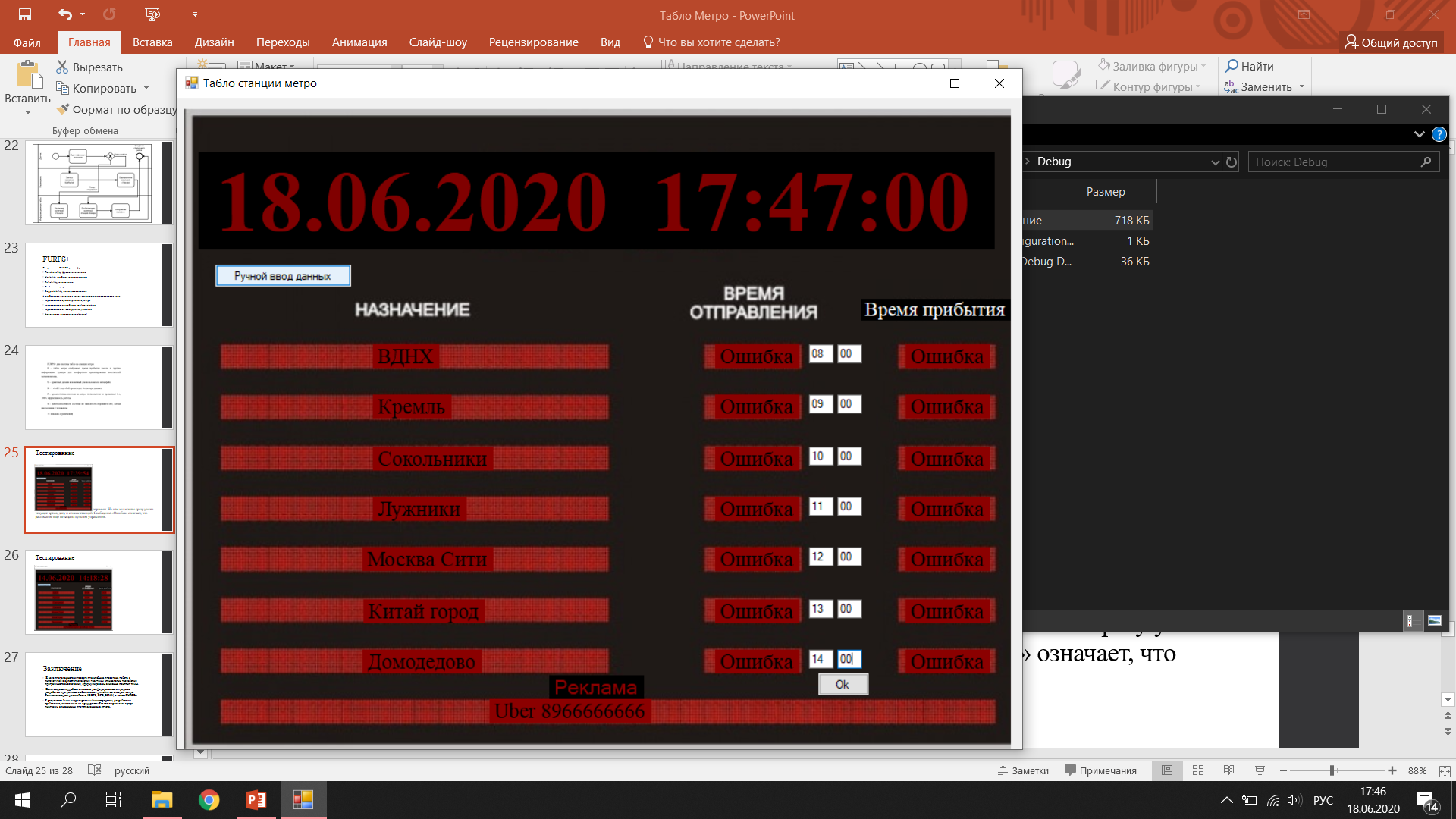


Рисунок 15 – Установка времени движения

По нажатию на клавишу «Ok» формируется расписание движения поездов. На нижней части экрана пользователь может увидеть рекламу,  
программа создана таким образом, что реклама периодически меняется. Данный факт можно лицезреть если сравнить скриншот ниже со скриншотами, представленными ранее.

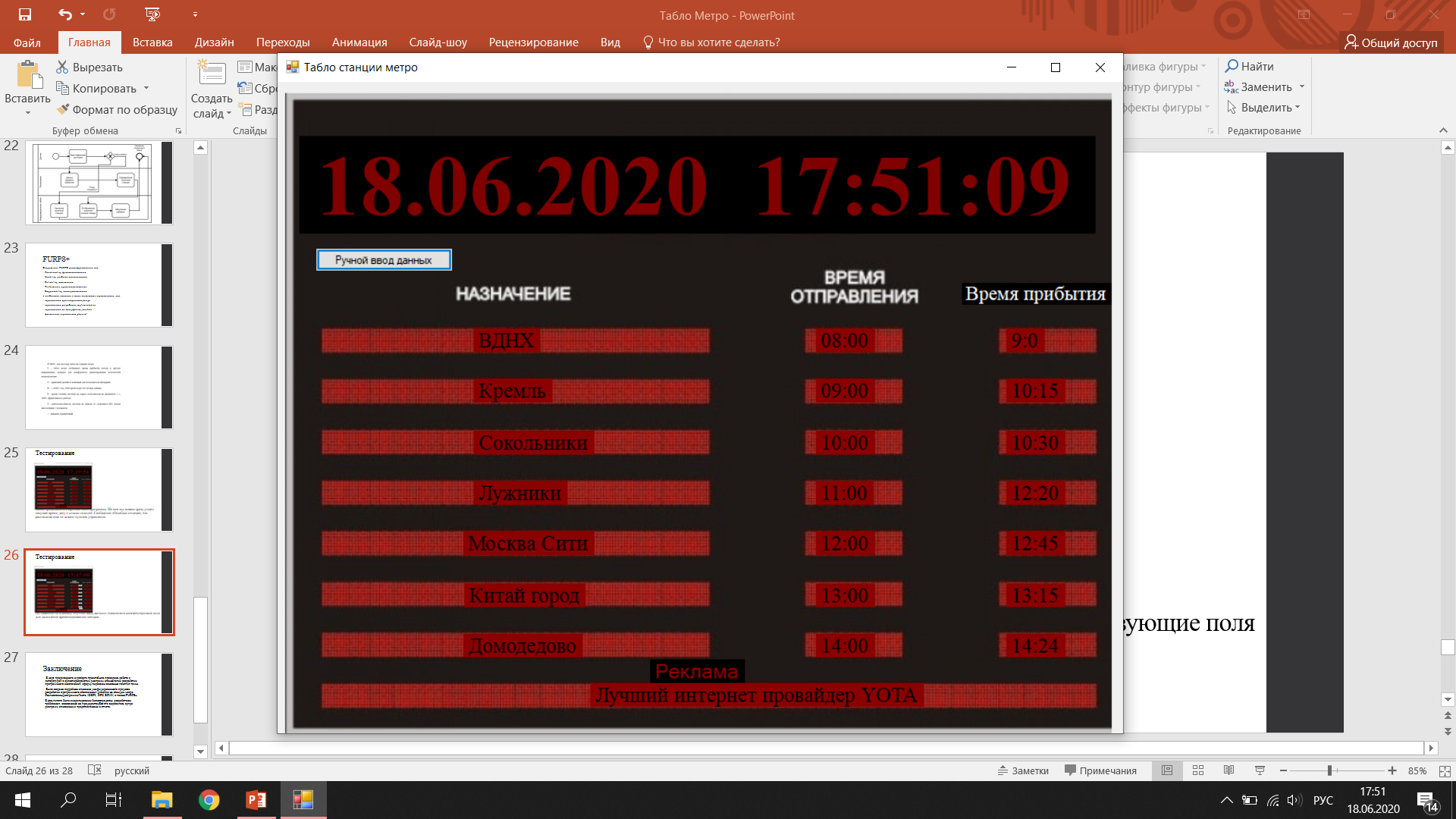


Рисунок 16 – Полное функционирование табло

**Заключение**

В результате выполнения данного курсового проекта было спроектировано «Табло на станции метро» на языке высокого уровня C#, позволяющее наглядно продемонстрировать работу всех его компонентов. В ходе выполнения были получены различные диаграммы для детального описания всех процессов протекающих в системе. Не смотря на размер диаграмм, они обладают достаточно простой структурой, понятной даже для обывателя.

При построении диаграмм использовались основные правила и принципы моделирования, включающие графическое представление объектов и связей между ними, иерархическое построение, а также названия, отражающие назначение той или иной сущности, или взаимодействия.

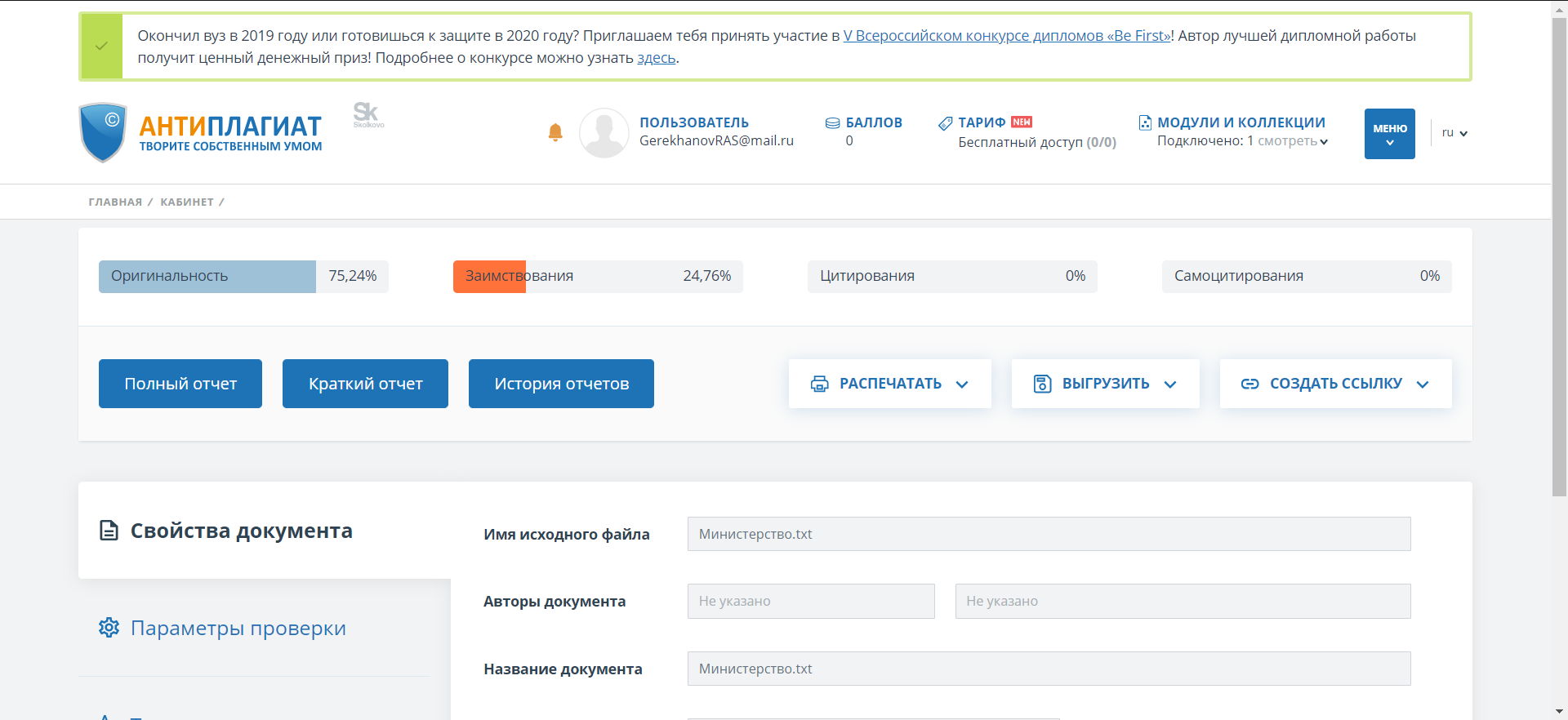
Благодаря детальному разбору проекта при помощи диаграмм проектирования, полученных в процессе разработки, можно заверить, что полученное «Табло на станции метро» полностью соответствует современным стандартам качества и способно выполнять различные функции требующиеся во всех крупных и мелких метрополитенах.

Были получены важные знания и практические навыки в области использования объектно-ориентированного языка программирования C#, а также обретен опыт построения различных диаграмм проектирования, детально описывающих процессы происходящие в системе.

**Список использованных источников**

1. Ларман, Крэг. Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку / КрэгЛарман. - Москва: Гостехиздат, 2017. - 736 c.
2. Роберт А. Максимчук. UML для простых смертных / Роберт А. Максимчук, Эрик Дж. Нейбург. - Москва: СИНТЕГ, 2014. - 272 c.
3. Йордон, Эдвард. Объектно-ориентированный анализ и проектирование систем / Эдвард Йордон , Карл Аргила. - М.: ЛОРИ, 2014. - 264 c.
4. SoloLearn – C# Tutorial. [Электронный ресурс]: - Режим доступа: <https://www.sololearn.com/Course/CSharp/> .
5. Википедия. [Электронный ресурс]: - Режим доступа: [https://ru.wikipedia.org](https://ru.wikipedia.org/wiki/Тестовые_функции_для_оптимизации)
6. Comindware – Нотация BPMN 2.0 [Электронный ресурс]: - Режим доступа: <https://comindware.com/ru/blog-нотация-bpmn-2-0-элементы-и-описание/>.
7. SysAna– Требования к системе: классификация FURPS+ [Электронный ресурс]: - Режим доступа: <https://sysana.wordpress.com/2010/09/16/furps/>.
8. Business Studio – Нотация EPC [Электронный ресурс]: - Режим доступа:<https://www.businessstudio.ru/wiki/docs/v4/doku.php/ru/csdesign/bpmodeling/epc_notation>.
9. Habr – Что такое DFD? [Электронный ресурс]: - Режим доступа: <https://habr.com/ru/company/trinion/blog/340064/>.

# Приложение А – Проверка на антиплагиат



# Приложение Б – Диаграмма Ганта

