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: «Турникет»

(тема курсовой работы)

Выполнил студент 2 курса группы 18-К-АС1

Мешков Г.Е.

(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: Мешков Г.Е. группы 18-К-АС1 курса 2

(Ф.И.О.) (№ группы и курса)

Тема проекта: «Турникет»

План работы:

1. Изучение предметной области

2. Проектирование

3. Описание реализованных диаграмм

Объем работы:

а) пояснительная записка 32 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

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

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 «27» ноября 2020 г.

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

(должность, подпись,)

Задание принял студент Мешков Г.Е, Ф.И.О.

**Реферат**

Курсовая работа: 32 страниц, 13 рисунков, 8 используемых источников.

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

Объектом исследования является программное обеспечение турникета в метро, который способен открывать и закрывать блокировочные элементы.

Цель работы состоит в разработке проекта программного обеспечения «Турникет» 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 Турникета. 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](#_Toc39780977)

[1 Формулировка задачи 6](#_Toc39780978)

[2 Диаграмма Ганта 7](#_Toc39780979)

[3 Создание модели As-Is в стандарте IDEF0 8](#_Toc39780980)

[4 Диаграмма потоков данных (DFD) 12](#_Toc39780981)

[5 UML 14](#_Toc39780982)

[6 EPC 15](#_Toc39780983)

[7 BPMN 16](#_Toc39780984)

[8 FURPS+ 17](#_Toc39780985)

[9 Результаты машинного тестирования программы 18](#_Toc39780986)

[10 Системные требования 20](#_Toc39780987)

[11 Руководство пользователя 21](#_Toc39780988)

[Заключение 23](#_Toc39780989)

[Список использованных источников 23](#_Toc39780989)4

[Приложение А – Проверка на антиплагиат 25](#_Toc39780990)

[Приложение Б – Диаграмма Ганта 26](#_Toc39780991)

[Приложение В – Листинг программы 27](#_Toc39780992)

**Введение**

В настоящее время турникеты получили большую популярность и используются во многих предприятиях, ТЦ, метро, учебных заведениях и т.п.

В первых турникетах приходилось использовать одноразовые жетоны для оплаты прохода. В современных используется технология бесконтактной оплаты (NFC), что упрощает взаимодействия людей с турникетом.

Формируются базы данных вручную или автоматизировано, то есть, новому пользователю выдается специальная карта, оснащенная NFC, которой присваивается свой ID и ведется учет информации о количестве поездок и сроки действия самой карты.

Базы данных могут находится как на личных серверах, установленных в здании частного предприятия и подключенных по локальной сети с турникетом, для введения учета о передвижениях сотрудников(во сколько приходят на работу, во сколько уходят), так и через интернет сеть, если необходим доступ к БД из разных точек, раскинутых по району, городу и т.д.

Данный проект – это макет турникета, в котором реализованы функции оплаты проезда, создания БД, добавление новых ID пассажиров в БД или скачивание уже готовой БД.

**1 Формулировка задачи**

Программное обеспечение встроенного процессора турникета для метрополитена.

При помощи турникета контролируется проход пассажиров в метро и взимается входная плата. Турникет имеет приемник карт, устройство для перекрывания доступа, таймер, три оптических датчика для определения прохода пассажира, устройство подачи звуковых сигналов, ин­дикаторы «Проход» и «Стоп».

В начальном состоянии турникета зажжен индикатор «Стоп», индикатор «Проход» потушен. Если один из датчиков посылает сигнал, то проход через турникет сразу же перекрывается, и подается предупредительный звуковой сигнал. Для прохода пассажир должен поместить карту в приемник карт. Турникет считывает с нее данные: срок годности карты и количество «единиц» на ней. Если данные не удается считать, или карта просрочена, или заблокирована, то карта возвращается пассажиру, и турникет остается в исходном состоянии. В другом случае с карты списывается одна «единица», карта возвращается из приемника, индикатор «Стоп» гаснет, зажигается индикатор «Проход», и пассажир может пройти через турникет. Получив от одного из датчиков сигнал, турникет ожидает время, отведенное на проход пассажира (5 секунд), после чего он возвращается в начальное состояние.

Наличие трех датчиков в турникете гарантирует, что при проходе пассажира хотя бы один из них подаст сигнал (датчики невозможно перешагнуть, перепрыгнуть и т. д.). Во время прохода пассажира возможна ситуация, когда все три датчика посылают сигналы. В этом случае принимается только первый сигнал и от момента его приема отсчитывается положенное время. Остальные сигналы игнорируются. Турникет заносит в свою память время всех оплаченных проходов. В конце рабочего дня он передает всю информацию, накопленную за день, в АСУ метрополитена.

# 2 Диаграмма Ганта

Упростить планирование проекта со множеством задач поможет составление диаграммы Ганта. На ней будет виден не только список задач, но и кто, что и когда должен сделать.

Диаграмма Ганта — «это популярный тип столбчатых диаграмм, который используется для иллюстрации плана, графика работ по какому-либо проекту. Является одним из методов планирования проектов. Придумал американский инженер Генри Гант. Выглядит это как горизонтальные полосы, расположенные между двумя осями: списком задач по вертикали и датами по горизонтали.

Ключевым понятием диаграммы Ганта является «веха» — метка значимого момента в ходе выполнения работ, общая граница двух или более задач. Вехи позволяют наглядно отобразить необходимость синхронизации, последовательности в выполнении различных работ. Вехи, как и другие границы на диаграмме, не являются календарными датами. Сдвиг вехи приводит к сдвигу всего проекта. Поэтому диаграмма Ганта не является, строго говоря, графиком работ. Кроме того, диаграмма Ганта не отображает значимости или ресурсоемкости работ, не отображает сущности работ (области действия). Для крупных проектов диаграмма Ганта становится чрезмерно тяжеловесной и теряет всякую наглядность.»

Диаграмма Ганта для проекта «Турникет» находится в «Приложении Б».

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

Прежде чем пытаться выбрать существующую или создать собственную информационную систему, а затем внедрить ее, необходимо проанализировать, как работает система в настоящее время. Для этого строится функциональная модель AS-IS. Анализ этой функциональной модели позволяет понять, где находятся наиболее слабые места, в чем будут состоять преимущества новых бизнес-процессов. Детализация бизнес-процессов позволяет выявить недостатки. Найденные в модели AS-IS недостатки можно исправить при создании модели ТО-ВЕ - модели новой организации бизнес-процессов. Модель ТО-ВЕ нужна для оценки последствий внедрения информационной системы и анализа альтернативных путей выполнения работы и документирования того, как система будет функционировать в будущем.

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 ресурсы;

– стрелкой сверху обозначаются контролирующие нормы;

– стрелкой справа обозначает результат работы;

– стрелкой снизу обозначает механизм, с помощью которого осуществлялась работа.

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

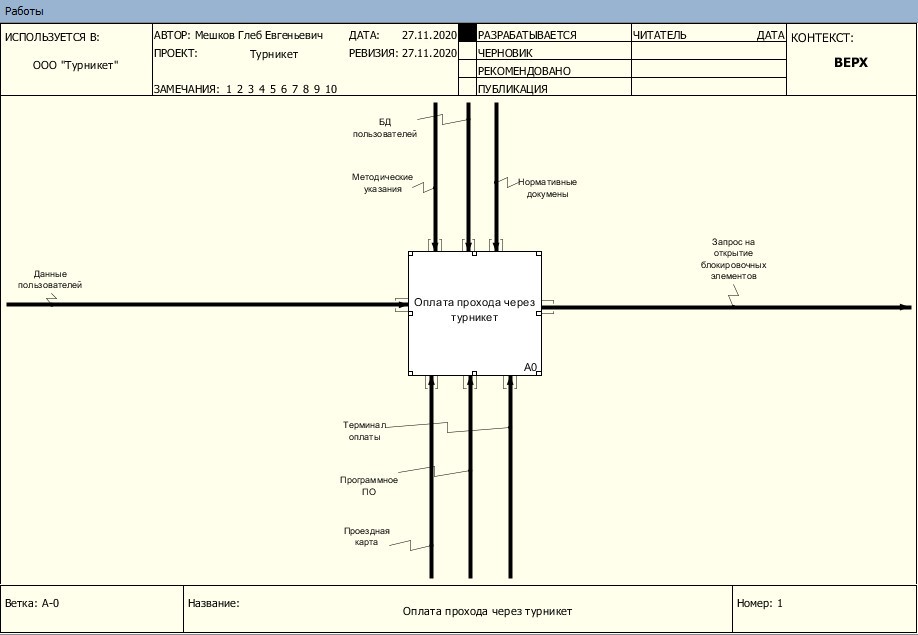


Рисунок 1 – Модель As-Is проекта «Турникет»

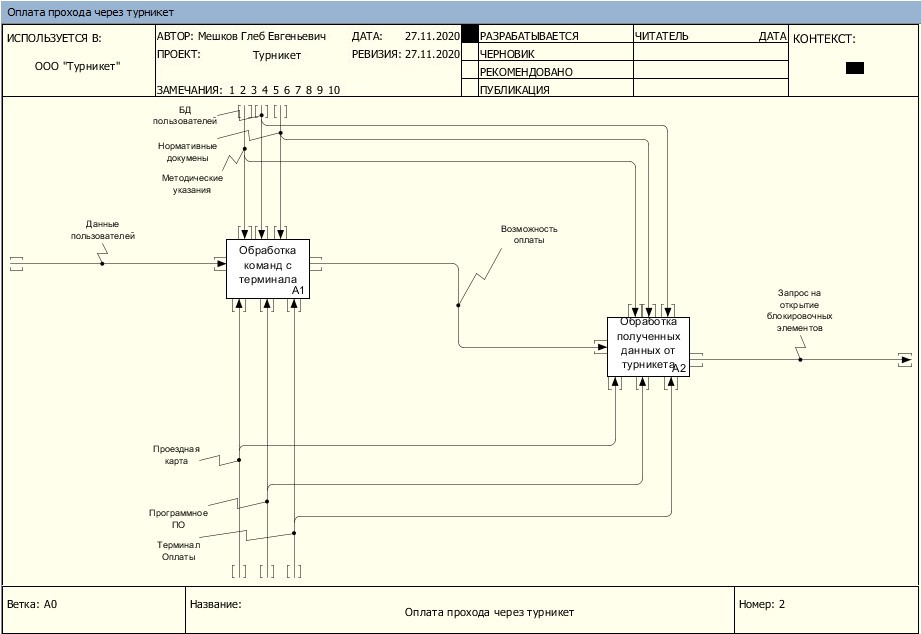


Рисунок 2 – Декомпозиция проекта «Турникет»

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

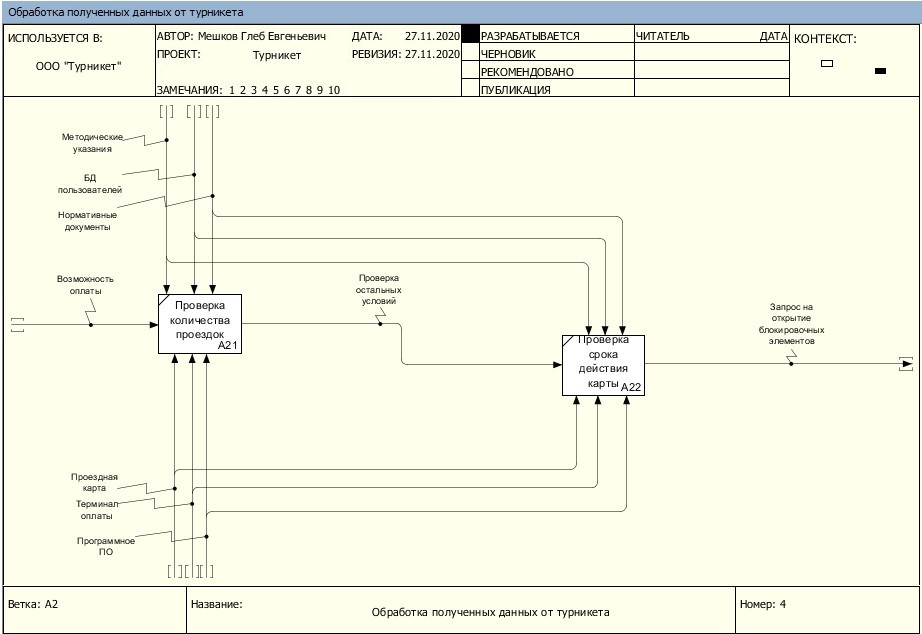


Рисунок 3 – Декомпозиция кода обработки полученной информации от проездной карты.

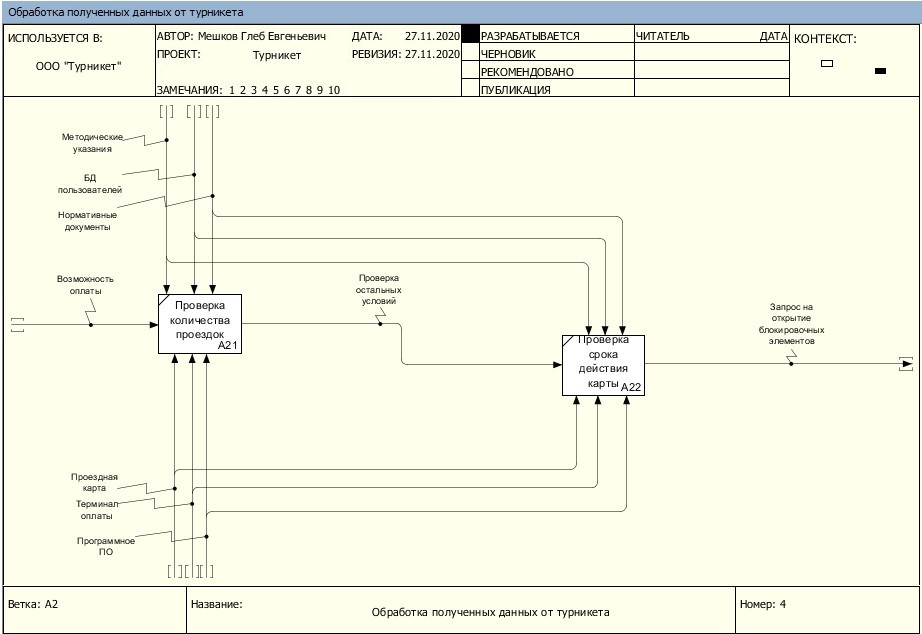


Рисунок 4 – Декомпозиция кода

На вход поступают данные с проездной карты, которые вводятся в терминал и обрабатываются турникетом. Эти данные обрабатываются и проходят условия проверки.

В конце выводиться результат проверки, который сопровождается визуальными и звуковыми эффектами.

Таким образом работа, рассматриваемой системы разбивается на две основанных функции:

– обработка полученной информации;

– вывод результата обработки;

**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 «Турникет» была получена следующая диаграмма DFD (рис. 5).

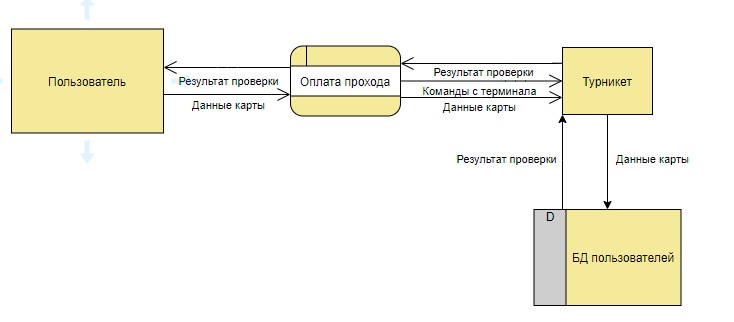


Рисунок 5 – Диаграмма DFD-системы «Турникет»

Внешними сущностями данной системы являются: пользователь, Турникет, база данных.

**5 UML**

UML — 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.»

Преимущества UML:

UML объектно-ориентирован, в результате чего методы описания результатов анализа и проектирования семантически близки к методам программирования на современных объектно-ориентированных языках;

UML позволяет описать систему практически со всех возможных точек зрения и разные аспекты поведения системы;

Диаграммы UML сравнительно просты для чтения после достаточно быстрого ознакомления с его синтаксисом;

UML расширяет и позволяет вводить собственные текстовые и графические стереотипы, что способствует его применению не только в сфере программной инженерии;

UML получил широкое распространение и динамично развивается.

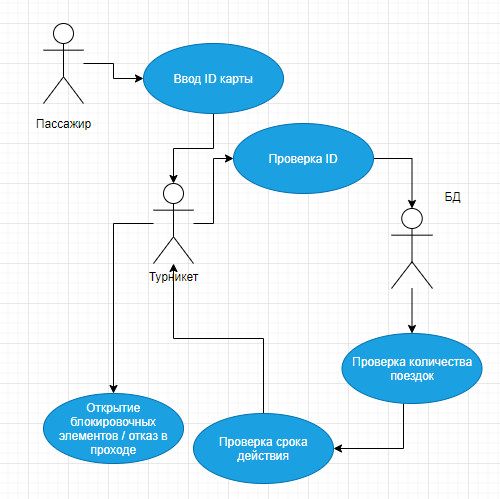


Рисунок 6 – UML-диаграмма системы «Турникет»

# 6 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 инструмент 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.»

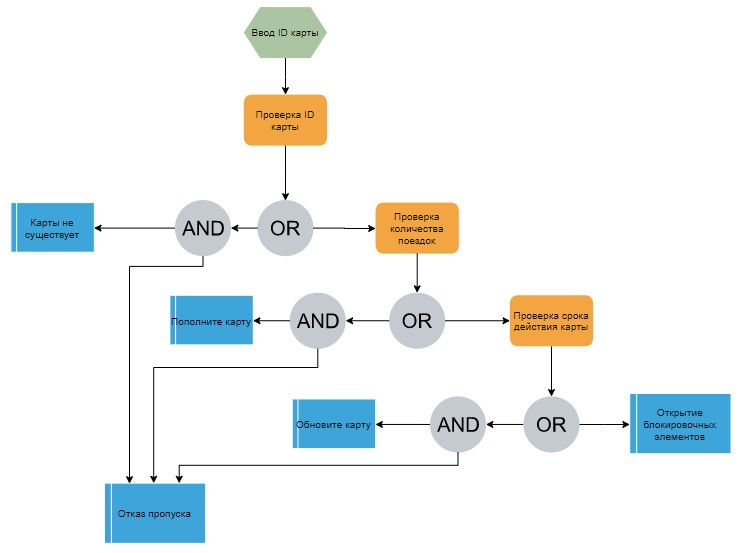


Рисунок 7 – EPC-диаграмма системы «Турникет»

# 7 BPMN

Бизнес-процесс — это многократно повторяющаяся, логически связанная последовательность действий, направленная на создание ценности и формирование результата. И для таких процессов существует нотация, которая представляет собой описание графических элементов, используемых для построения схемы протекания рассматриваемого процесса.

BPMN – это язык моделирования бизнес-процессов, который является промежуточным звеном между формализацией/визуализацией и воплощением бизнес-процесса.

Такая схема нужна, чтобы выстроить в соответствии с ней бизнес-процесс и понятно регламентировать его для всех участников. Моделирование BPMN позволяет впоследствии провести автоматизацию бизнес-процессов в соответствии с имеющейся схемой.»

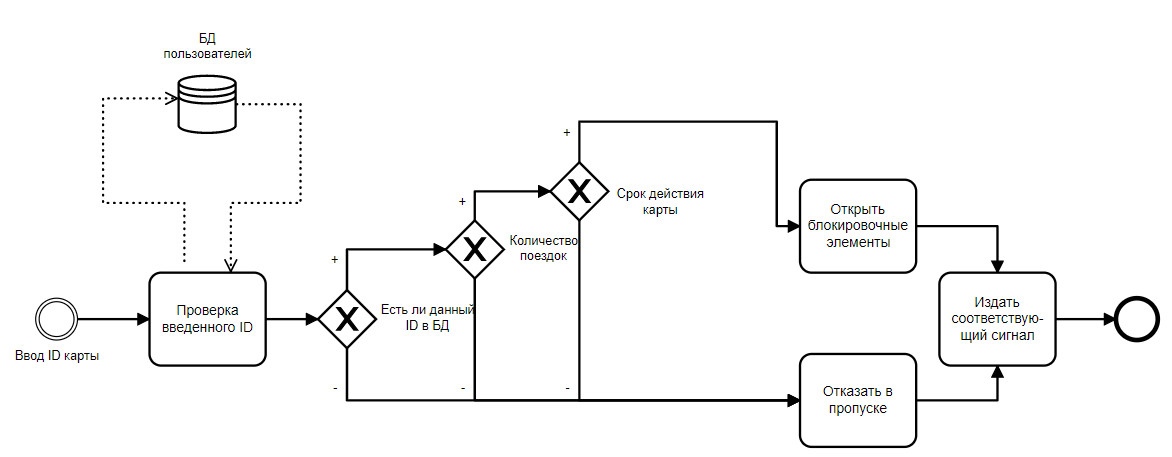


Рисунок 8 – Диаграмма BPMN

# 8 FURPS+

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

F – Functionality, функциональность. Включает в себя функциональные возможности системы, которыми могут выступать возможность автоматизированного выполнения функций основных бизнес-процессов, автоматизация отдельных функций по информационной поддержке работы с партнерами, автоматизированная обработка и анализ управленческой информации.

U – Usability, удобство использования. Включает в себя: интуитивная понятность, эстетическая привлекательность, доступность и удобство в получении справочной информации.

R – Reliability, надежность. Включает в себя: режим доступности системы, точность проводимых расчетов, отказоустойчивость, пригодность ИС к восстановлению данных и информации.

P – Performance, производительность. Включает в себя: пропускная способность, время отклика системы, масштабируемость.

S – Supportability, поддерживаемость. Включает в себя: пригодность к проведению ремонтных работ, простота установки, адаптируемость к условиям конкретной среды, совместимость с программно-аппаратным обеспечением и другими ИС.

+ необходимо помнить о таких возможных ограничениях, как:

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

Полученная для данного проекта FURPS+ выглядит так:

F – отсутствует;

U – интуитивная понятность;

R – режим доступности 24/7, защита от прослушивания, большая дальность связи;

P – долговечные аккумуляторы, моментальный отклик;

S – простая установка и настройка, не требующая профессиональных знаний;

+ - никаких ограничений

# 9 Результаты машинного тестирования программы

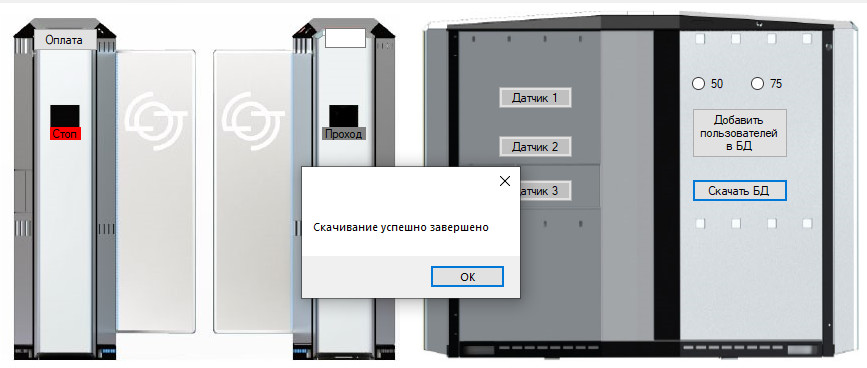


Рисунок 10 – Скачана база данных



Рисунок 11 – Успешная оплата



Рисунок 12 – Рабочие датчики

**10 Системные требования**

Таблица 1 – Системные требования программы

|  |  |
| --- | --- |
| Процессор | 2.5 ГГц |
| Оперативная память | 150 Мб |
| Монитор | 1920 x 1080 |
| Свободное место на носителе | 15 Мб |
| Устройства взаимодействия с пользователем | Клавиатура и мышь |
| Программное обеспечение | Visual Studio 2019 года последней версии |

**11 Руководство пользователя**

Запуск программы можно осуществить следующим способом: открыть исполняемый файл.

Запуск программы при помощи исполняемого файла:

* 1. Находим исполняемый файл.
  2. Запускаем исполняемый файл с форматом .exe.

Теперь перед нами интерфейс программы: кнопка «Оплата», «Скачать готовую БД», «Добавить пользователей в БД» и три кнопки датчиков движения.

Рисунок 13 – Общий вид программы

* 1. При наличии интернет-соединения нажимаем кнопку «Скачать БД», если нет – создаем базу данных с помощью выбора количества поездок и нажатию кнопки «Добавить пользователей в БД»
  2. В ячейку над подписью «Проход» вводим ID пассажира начиная с «1»
  3. Нажимая на кнопки датчиков можно убедиться, что пропуск одобрен или отказан.
  4. При нажатии кнопки «Esc» на клавиатуре, программа будет завершена.

**Заключение**

В результате выполнения данного курсового проекта была спроектирована система «Турникета» на языке высокого уровня C#.

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, а также тестированию, были выявлены и устранены ошибки в проекте. Конечный результат, полученный в данном проекте не идеален, но может быть доработан во время его эксплуатации и при дальнейшей поддержке проекта.

Во время выполнения работы, разработка «Турникет» несла в себе новые знания о работе с языком высокого уровня С#. Новая программа объединяет в себе две разные программы, которые взаимодействуют между собой и не конфликтуют.

**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/> (Дата обращения 13.03.2020).
5. Википедия. [Электронный ресурс]: - Режим доступа: [https://ru.wikipedia.org](https://ru.wikipedia.org/wiki/Тестовые_функции_для_оптимизации) (Дата обращения 17.09.2019).
6. GitHub – yarajtf/intercom. [Электронный ресурс]: - Режим доступа: <https://github.com/yarajtf/intercom> (Дата обращения 06.05.2020).
7. Comindware – Нотация BPMN 2.0 [Электронный ресурс]: - Режим доступа: <https://comindware.com/ru/blog-нотация-bpmn-2-0-элементы-и-описание/> (Дата обращения 28.02.2020)
8. SysAna– Требования к системе: классификация FURPS+ [Электронный ресурс]: - Режим доступа: <https://sysana.wordpress.com/2010/09/16/furps/> (Дата обращения 03.03.2020)

# Приложение А – Проверка на антиплагиат



# Приложение Б – Диаграмма Ганта

# 

# Приложение В – Листинг программы

Код программы:

using System;

using System.Drawing;

using System.IO;

using System.Media;

using System.Net;

using System.Windows.Forms;

namespace Турникет

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

timer1.Interval = 5000;

timer2.Interval = 5000;

timer3.Interval = 5000;

timer4.Interval = 2000;

timer5.Interval = 2000;

}

SoundPlayer spg = new SoundPlayer("C:\\Проект Турникет\\Прочее\\spg.wav");

SoundPlayer spb = new SoundPlayer("C:\\Проект Турникет\\Прочее\\spb.wav");

WebClient wc = new WebClient();

static string cardid = "C:\\Проект Турникет\\Прочее\\БД пользователей.txt";

bool butt = false;

bool but2 = false;

bool but4 = false;

bool but5 = false;

bool but6 = false;

int i = 1;

bool lb1 = false;

bool lb2 = false;

private void CloseAway()

{

timer5.Start();

if (!but2)

label2.BackColor = Color.Red;

}

private void Button1\_Click(object sender, EventArgs e)

{

butt = true;

Directory.CreateDirectory("C:\\Проект Турникет\\Прочее\\");

wc.DownloadFile("https://raw.githubusercontent.com/Bagmesh/Toorniket/master/Турникет/Турникет/Прочее/IDпользователя.txt", cardid);

MessageBox.Show("Скачивание успешно завершено");

}

public string FindID(string s)

{

string str = null;

string[] array = File.ReadAllLines(cardid);

int index = Array.IndexOf<string>(array, s);

if (butt)

{

for (int i = 0; i < array.Length; ++i)

if (index < array.Length)

str = array[index];

}

else

MessageBox.Show("ID не найден!");

return str;

}

private void button2\_Click(object sender, EventArgs e)

{

try

{

string[] array = File.ReadAllLines(cardid);

int num2 = Array.IndexOf<string>(array, textBox1.Text);

int t = Convert.ToInt32(array[num2 + 1]);

if (t != 0)

{

if (textBox1.Text != FindID(textBox1.Text))

{

CloseAway();

spb.Play();

MessageBox.Show("Введенный id не существуюет!");

}

else

{

array[num2 + 1] = (t - 1).ToString();

File.Delete(cardid);

for (int i = 0; i < array.Length; ++i)

{

using (StreamWriter streamWriter = new StreamWriter(cardid, true))

streamWriter.WriteLine(array[i]);

}

spg.Play();

MessageBox.Show("Успешно =D");

lb1 = true;

but2 = true;

timer4.Start();

if (but2)

label1.BackColor = Color.Green;

}

}

else

{

CloseAway();

spb.Play();

MessageBox.Show("Пополните карту!");

}

}

catch (FormatException)

{

spb.Play();

MessageBox.Show("Введите id карты!");

}

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

Directory.CreateDirectory("C:\\Проект Турникет\\Прочее\\БД пользователей.txt");

++i;

bool T = Convert.ToBoolean(radioButton1.Checked);

bool G = Convert.ToBoolean(radioButton2.Checked);

if (T == true)

{

if (button1.Visible)

{

if (i < 10)

i = 11;

using (StreamWriter streamWriter = new StreamWriter(cardid, true))

{

streamWriter.WriteLine(i);

streamWriter.WriteLine(50);

}

}

else

{

using (StreamWriter streamWriter = new StreamWriter(cardid, true))

{

streamWriter.WriteLine(i);

streamWriter.WriteLine(50);

}

}

}

if (G == true)

{

if (button1.Visible)

{

if (i < 10)

i = 11;

using (StreamWriter streamWriter = new StreamWriter(cardid, true))

{

streamWriter.WriteLine(i);

streamWriter.WriteLine(75);

}

}

else

{

using (StreamWriter streamWriter = new StreamWriter(cardid, true))

{

streamWriter.WriteLine(i);

streamWriter.WriteLine(75);

}

}

}

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

}

private void radioButton2\_CheckedChanged\_1(object sender, EventArgs e)

{

}

private void button4\_Click(object sender, EventArgs e)

{

but4 = true;

timer1.Start();

if (but2)

button4.BackColor = Color.Green;

else

{

button4.BackColor = Color.Red;

spb.Play();

}

}

private void button5\_Click(object sender, EventArgs e)

{

but5 = true;

timer2.Start();

if (but2)

button5.BackColor = Color.Green;

else

{

button5.BackColor = Color.Red;

spb.Play();

}

}

private void button6\_Click(object sender, EventArgs e)

{

but6 = true;

timer3.Start();

if (but2)

button6.BackColor = Color.Green;

else

{

button6.BackColor = Color.Red;

spb.Play();

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

if (but4)

button4.BackColor = Color.Silver;

but4 = !but4;

but2 = !but2;

}

private void timer2\_Tick(object sender, EventArgs e)

{

if (but5)

button5.BackColor = Color.Silver;

but5 = !but5;

but2 = !but2;

}

private void timer3\_Tick(object sender, EventArgs e)

{

if (but6)

button6.BackColor = Color.Silver;

but6 = !but6;

but2 = !but2;

}

private void Form1\_Load(object sender, EventArgs e)

{

}

protected override bool ProcessDialogKey(Keys keyData)

{

if (keyData == Keys.Escape)

{

this.Close();

return true;

}

else

return base.ProcessDialogKey(keyData);

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void timer4\_Tick(object sender, EventArgs e)

{

if (but2)

label1.BackColor = Color.Gray;

lb1 = !lb1;

but2 = !but2;

}

private void timer5\_Tick(object sender, EventArgs e)

{

if (!but2)

label2.BackColor = Color.Red;

else

label2.BackColor = Color.Gray;

lb2 = !lb2;

but2 = !but2;

}

}

}