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: «Таксофон»

(тема курсовой работы)

Выполнил студент 2 курса группы 18-К-АС1

Кислов А.А.

(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: Кислову А.А. группы 18-К-АС1 курса 2

(Ф.И.О.) (№ группы и курса)

Тема проекта: «Таксофон»

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

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 Кислов А.А. Ф.И.О.

**Реферат**

Курсовая работа: 32 страницы, 14 рисунков, 8 используемых источников.

Ключевые слова: ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ПРОЕКТИРОВАНИЕ, ПОСЛЕДОВАТЕЛЬНОСТИ, МОДЕЛЬ, КЛАСС, ТАКСОФОН, UML, FORM, BPMN, ГАНТ, EPC, FURPS+, IDEF0, DFD, ДИАГРАММЫ.

Объектом исследования является программное обеспечение и симулятор таксофона, который способен передавать сообщения на расстоянии, оказывать платные услуги телефонной связи. Он подключен к линии связи.

Цель работы состоит в разработке проекта программного обеспечения «Таксофон» 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 таксофона. К ним относятся: диаграмма Ганта, UML-диаграмма, IDEF0-диаграмма, DFD-диаграмма, EPC-диаграмма, BPMN-диаграмма, документ FURPS+.

**Содержание**

[Введение 5](#_Toc39780977)

[1 Формулировка задачи 6](#_Toc39780978)

[2 Диаграмма Ганта 8](#_Toc39780979)

[3 Создание модели As-Is в стандарте IDEF0 9](#_Toc39780980)

[4 Диаграмма потоков данных (DFD) 13](#_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 таксофоны служат одним из основных средств общения между людьми на расстоянии. К числу достоинств, послуживших причиной такого масштабного распространения технологии, можно отнести простоту обращения, надёжность, доступность. Применение таксофонов в уличной среде стала неотъемлемой частью человеческого бытия.

Однако, несмотря на повсеместное использование данной технологии, она никогда не смогла бы обеспечить надлежавший уровень передачи информации без тщательного исследования технической области и проведения различных тестов, учитывающих всевозможные взаимодействия с компонентами системы. Для уменьшения денежных затрат стали использовать симуляторы - «имитаторы (обычно механические или компьютерные), задача которых состоит в имитировании управления каким-либо процессом, аппаратом или транспортным средством».

Таким образом, симулятор таксофона – имитатор таксофонной системы, который позволяет управлять основным блоком передачи информации.

**1 Формулировка задачи**

Задачей данного курсового проекта является разработка модели программного обеспечения встроенной системы управления работой таксофона городской телефонной сети.

Таксофон предназначен для оказания платных услуг телефонной связи. Он подключен к линии связи. В нем имеется кнопочная панель, дисплей, трубка со встроенным микрофоном и громкоговорителем, приемник карт – устройство для считывания телефонных карт, используемых для оплаты разговора. В начальном состоянии трубка таксофона повешена, дисплей потушен, таксофон не реагирует на нажатия кнопок и какие-либо сигналы из линии. При снятии трубки таксофон выдает на дисплей сообщение «Вставьте карту» и ожидает, когда пользователь вставит карту в приемник. Дальнейшее функционирование таксофона осуществляется только при вставленной карте. Если карту вынимают, таксофон возвращается к началу и выдает сообщение о необходимости вставить карту. При попадании карты в приемник производится считывание информации с карты. Если кредит исчерпан или карта не пригодна (не удается узнать кредит), то таксофон выдает соответствующее сообщение на дисплей таксофона. Если карта может быть использована для оплаты, то на дисплей выдаётся количество «единиц» на карте, и на телефонную станцию (АТС) подаётся сигнал «Трубка». При получении ответного сигнала «Тон» из линии таксофон воспроизводит звуковой тон «Готов» (длинный непрекращающийся гудок) в трубку. При получении сигнала «Занято», в трубке воспроизводится тон «Занято» (короткие гудки). После получения от АТС сигнала «Тон» от пользователя принимаются семизначный номер вызываемого абонента, остальные нажатия на кнопки игнорируются. Когда пользователь нажимает на кнопку с цифрой соответствующий ей сигнал «Цифра» передается АТС. Во время набора номера введённые цифры отображаются на дисплее. В ответ на набранный номер от АТС приходит либо сигнал «Занято», либо сигнал «Вызов». При получении сигнала «Вызов» таксофон воспроизводит в трубку длинные гудки до того момента, когда АТС осуществит коммутацию и передаст сигнал «Данные». Таксофон воспроизводит данные, передаваемые с сигналом, в трубку. При получении данных из трубки, аппарат преобразует их в сигнал «Данные» и передает их АТС. Во время разговора на дисплее ведётся отсчёт времени и уменьшается кредит на телефонной карте - каждые 15 секунд вычитается четверть «единицы». Обмен данными прерывается, в следующих случаях: − исчерпан кредит; − карта вынута из приемника; − от АТС пришел сигнал «Занято»; − повешена трубка таксофона. Если трубка была повешена, аппарат посылает в линию сигнал «Конец» и выдает на дисплей сообщение «Выньте карту». После извлечения карты из приемника таксофон переходит в начальное состояние.

# 2 Диаграмма Ганта

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

На диаграмме видны не только сами задачи, но и их последовательность. Это позволяет ни о чём не забыть и делать всё своевременно.

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 «Таксофон» 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 «Таксофон».

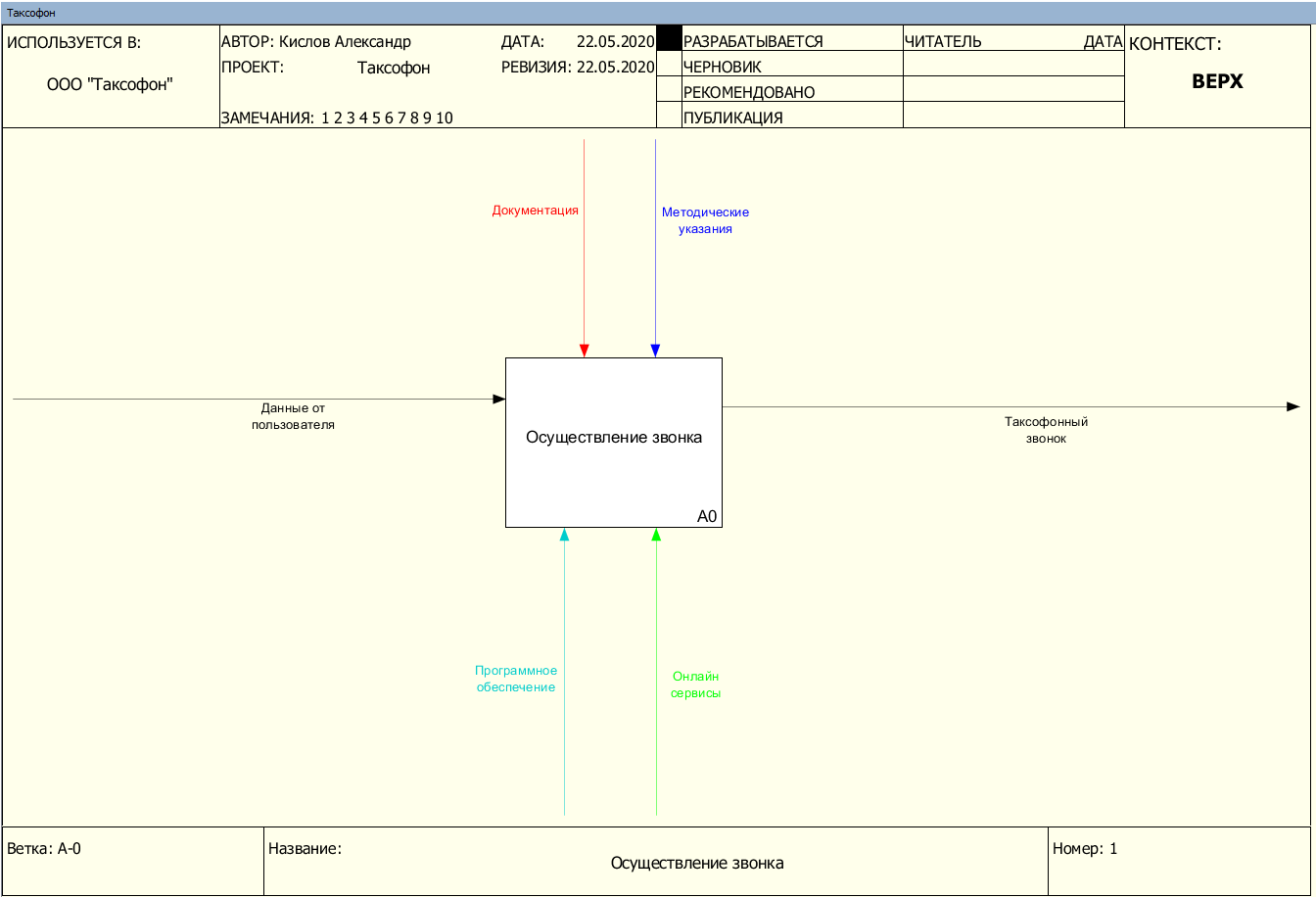


Рисунок 1 – Модель As-Is проекта «Таксофон»

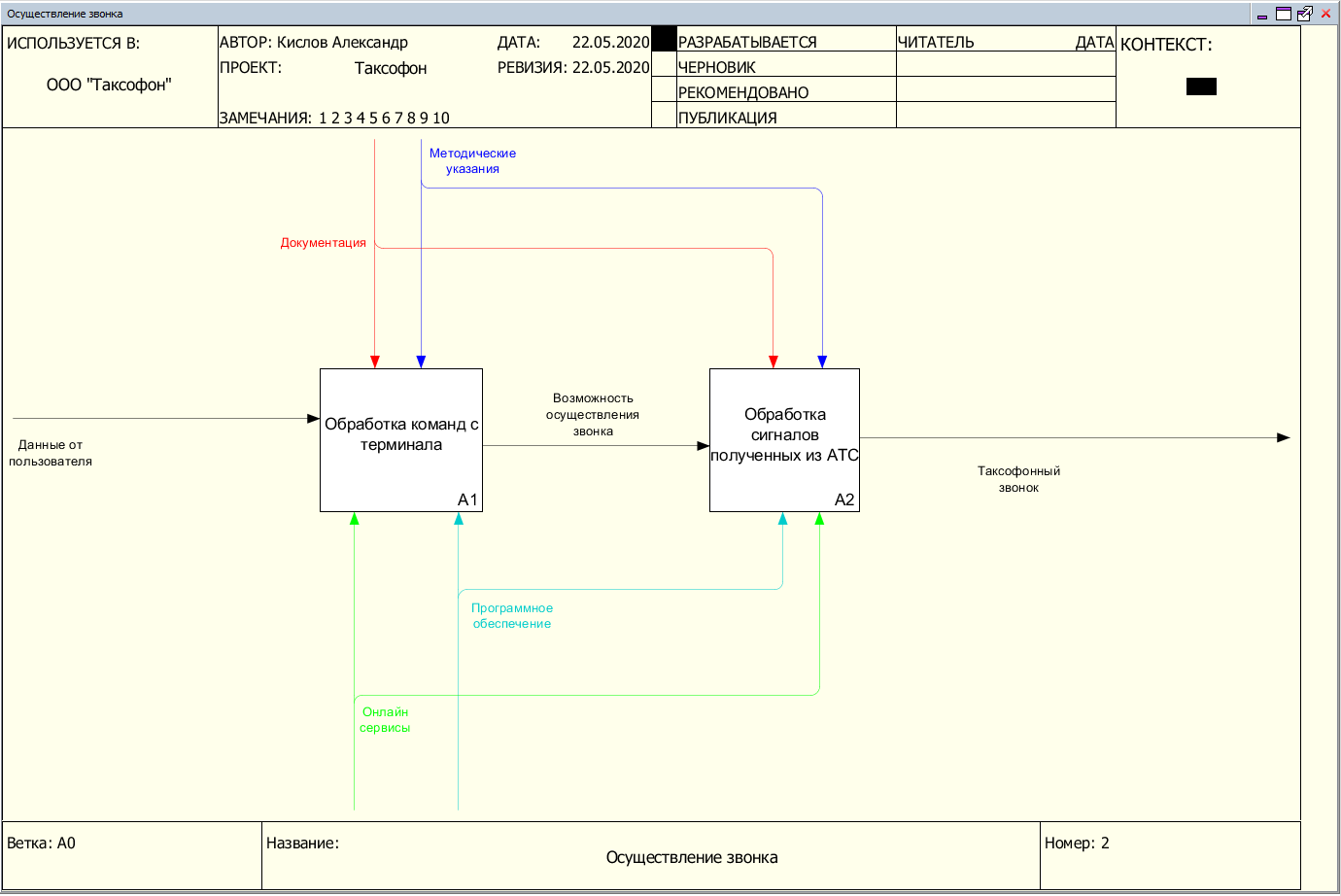


Рисунок 2 – Декомпозиция проекта «Таксофон»

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

На рисунке 3 можно видеть модель обработки команд с терминала «Таксофон» после декомпозиции.

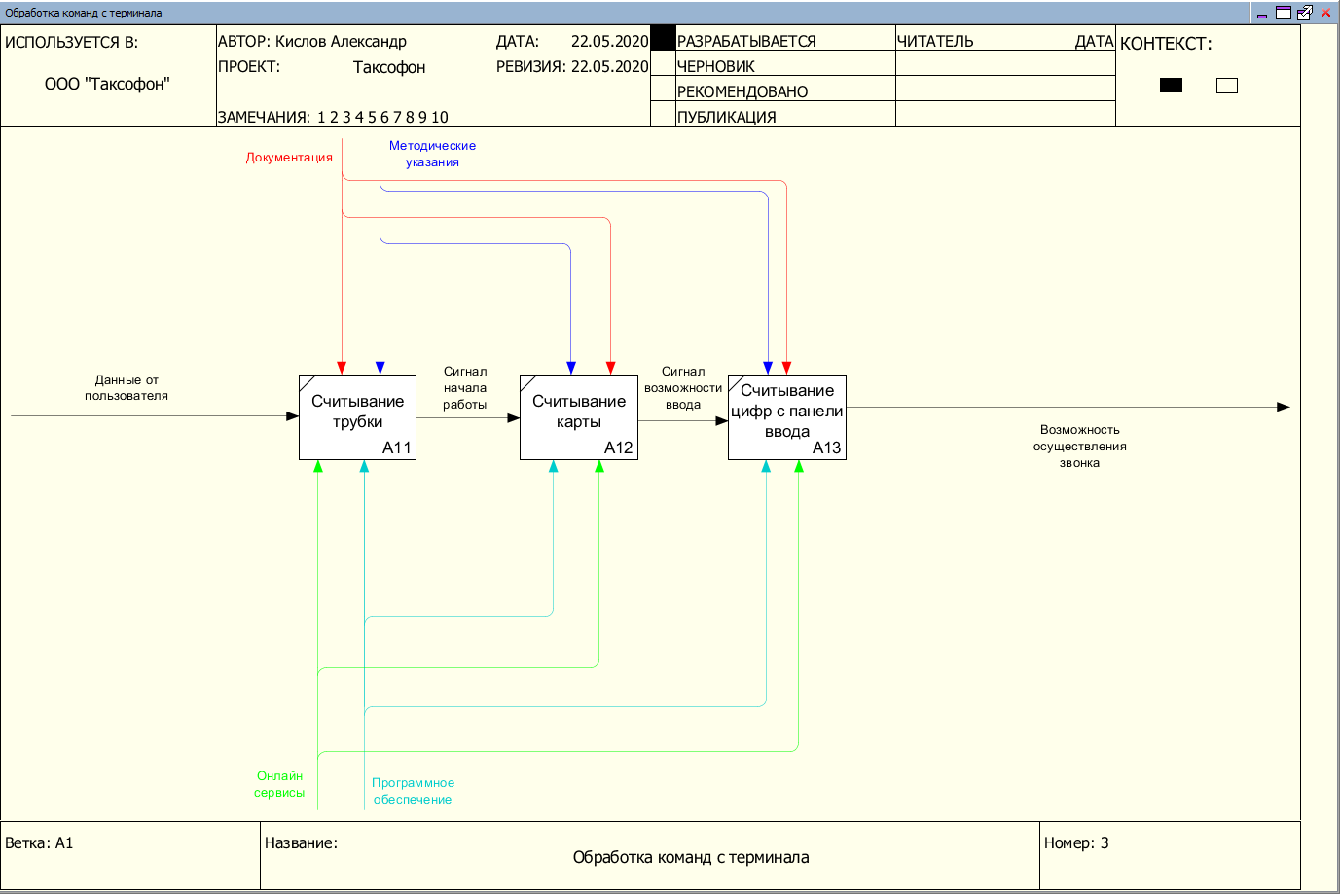


Рисунок 3 – Обработка команд с терминала

Входными данными для данной системы являются действия человека, необходимые для того, чтобы совершить звонок.

Управление происходит благодаря кнопочному интерфейсу, контроллеру устройства таксофона, а также правилам, описанным в информационной безопасности.

Механизмом реализации работы системы являются АТС, сигнал от таксофона и программное обеспечение таксофона.

Результатом деятельности системы является передача информации для осуществление таксофонного звонка.

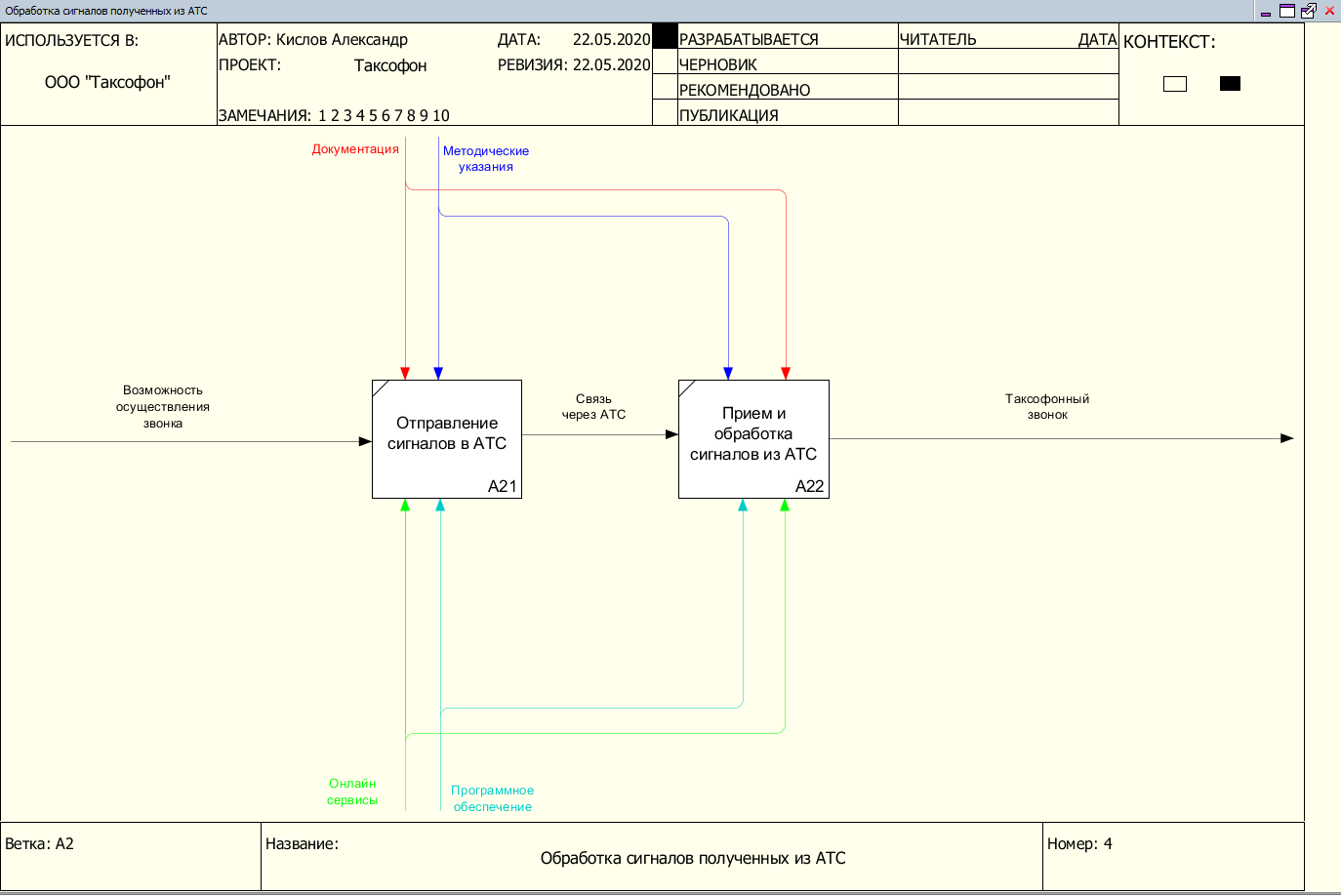


Рисунок 4 – Декомпозиция обработки сигналов из АТС

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

– приём входных данных от пользователя;

– анализ полученных данных;

– передача сообщения мини АТС.

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

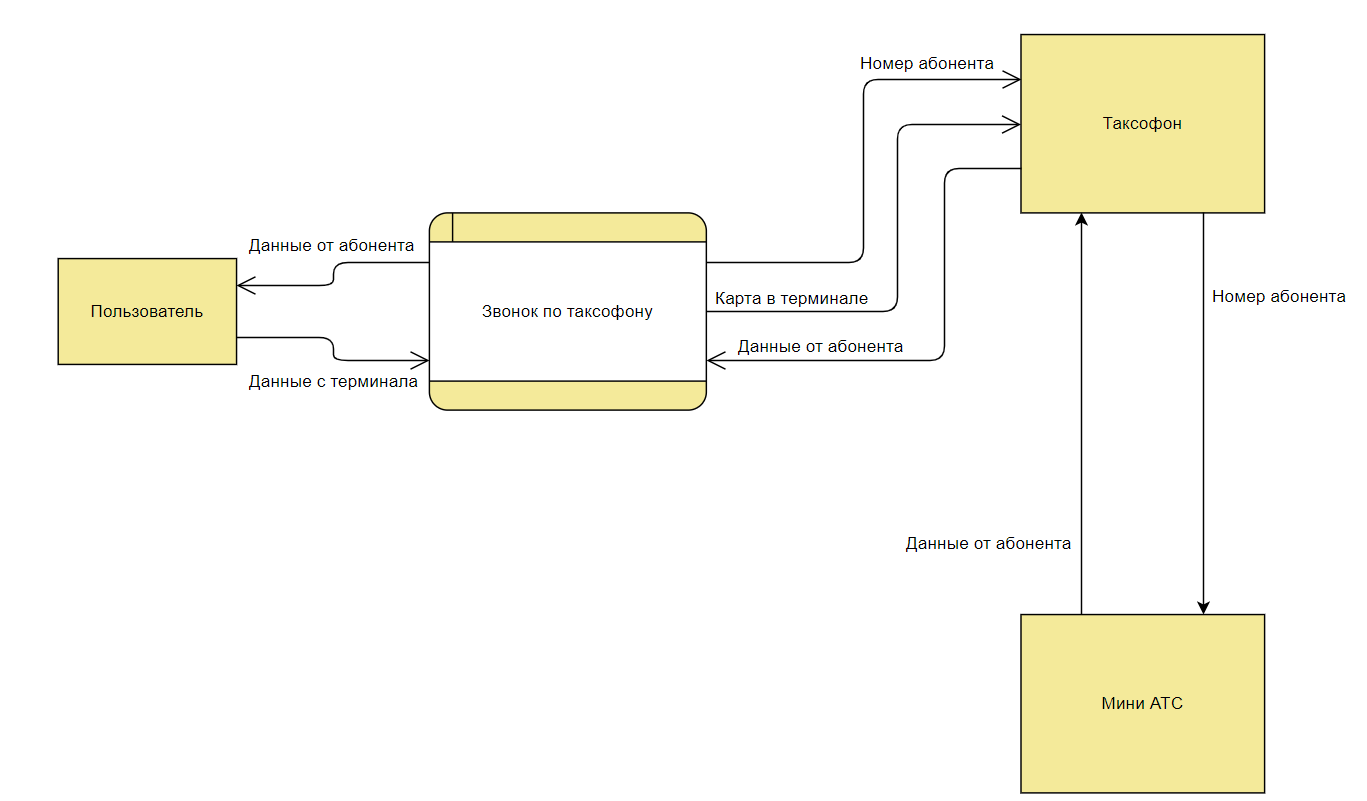


Рисунок 5 – Диаграмма DFD-системы «Таксофон»

Внешними сущностями данной системы являются человек у таксофона, мини АТС, а также сам таксофон. Система не содержит баз данных.

**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.»

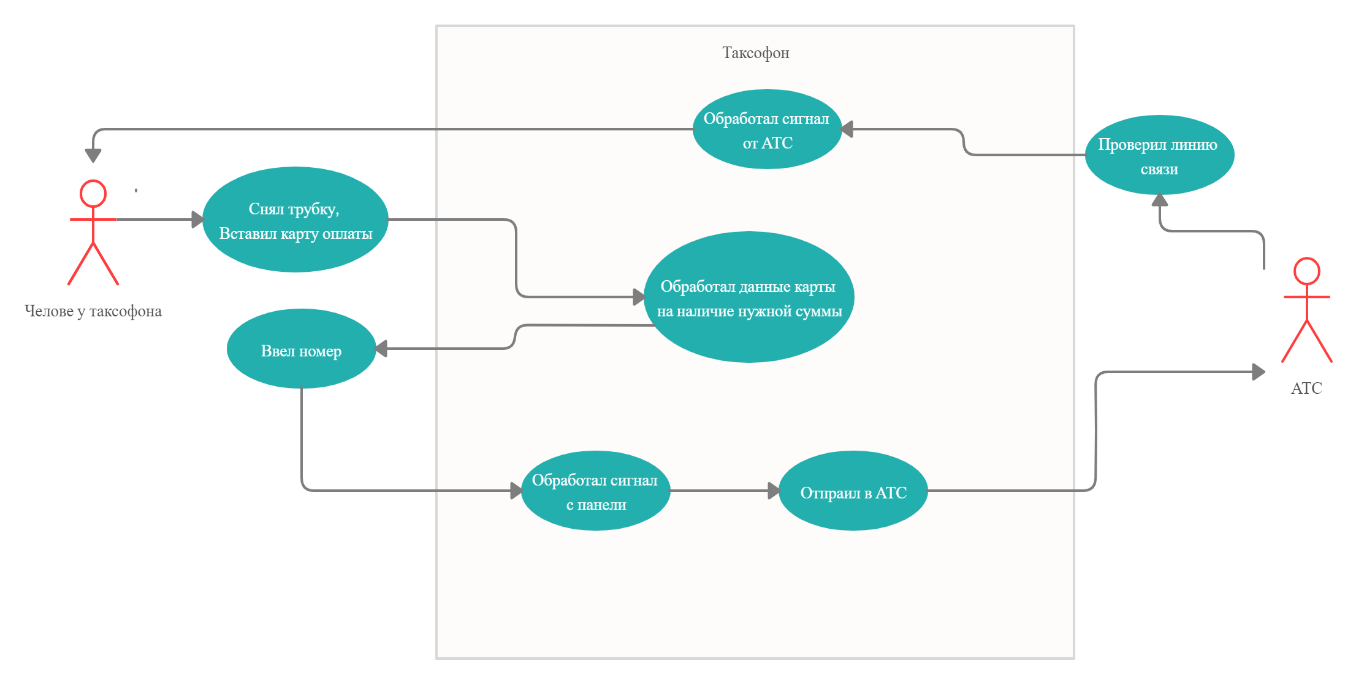


Рисунок 6 – UML-диаграмма системы «Таксофон»

# 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 инструментонезависимый 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.»

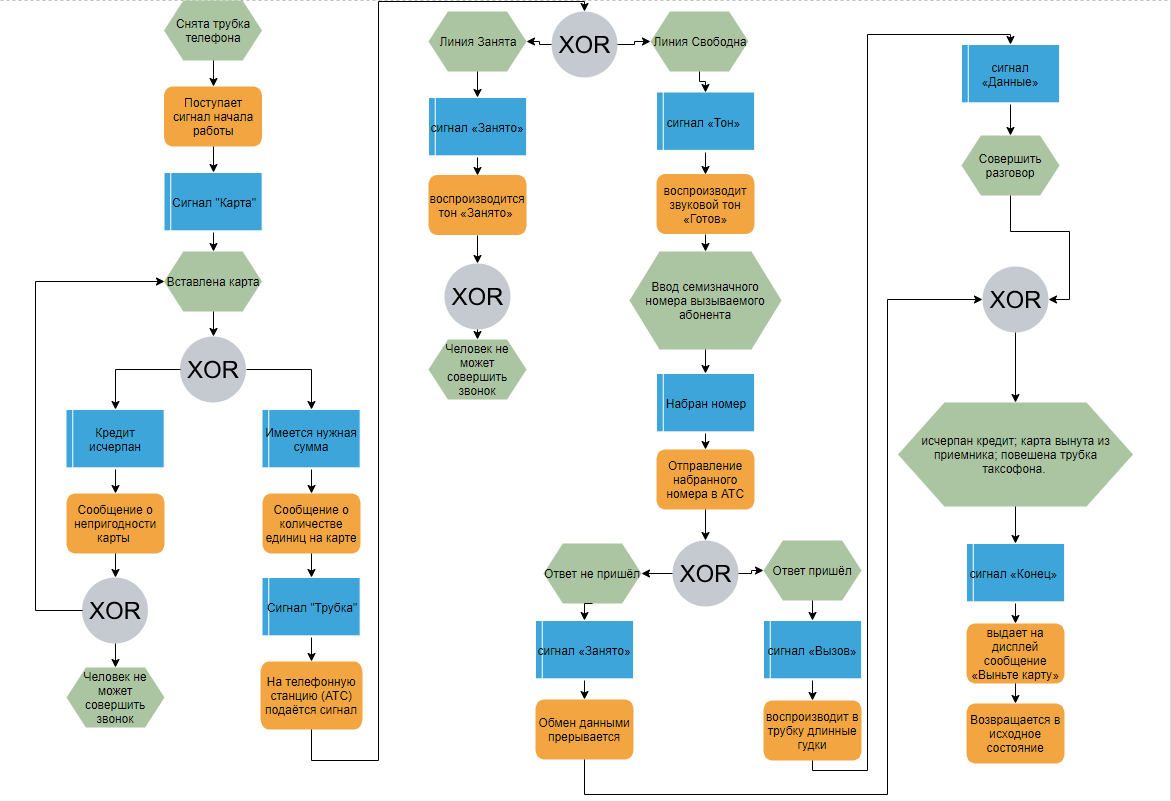


Рисунок 7 – EPC-диаграмма системы «Таксофон»

# 7 BPMN

BPMN (Business Process Management Notation) – «это язык моделирования бизнес-процессов, который является промежуточным звеном между формализацией/визуализацией и воплощением бизнес-процесса.

Говоря проще, такая нотация представляет собой описание графических элементов, используемых для построения схемы протекания бизнес-процесса.

Как минимум, такая схема нужна, чтобы выстроить в соответствии с ней бизнес процесс и понятно регламентировать его для всех участников.

Как максимум, моделирование BPMN позволяет впоследствии провести автоматизацию бизнес-процессов в соответствии с имеющейся схемой.»

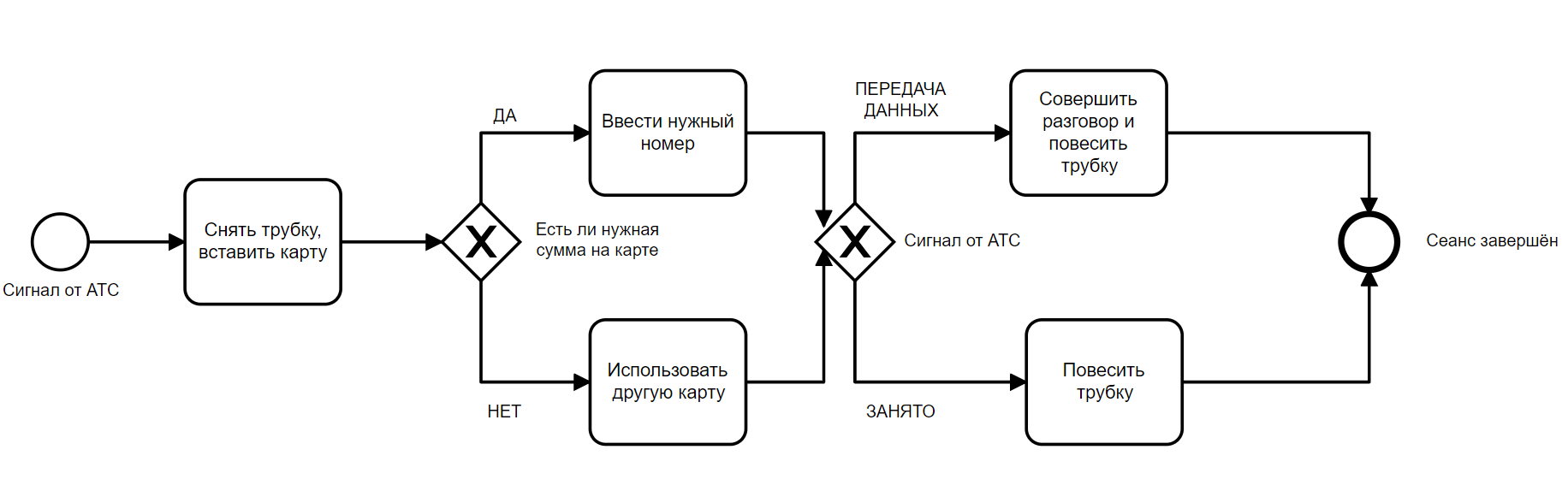


Рисунок 8 – Диаграмма BPMN системы «Таксофон»

# 8 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

Если применить к этой классификации популярное разделение требований на функциональные и нефункциональные, то к последним следует отнести все перечисленные выше группы кроме первой, т.е. URPS+.

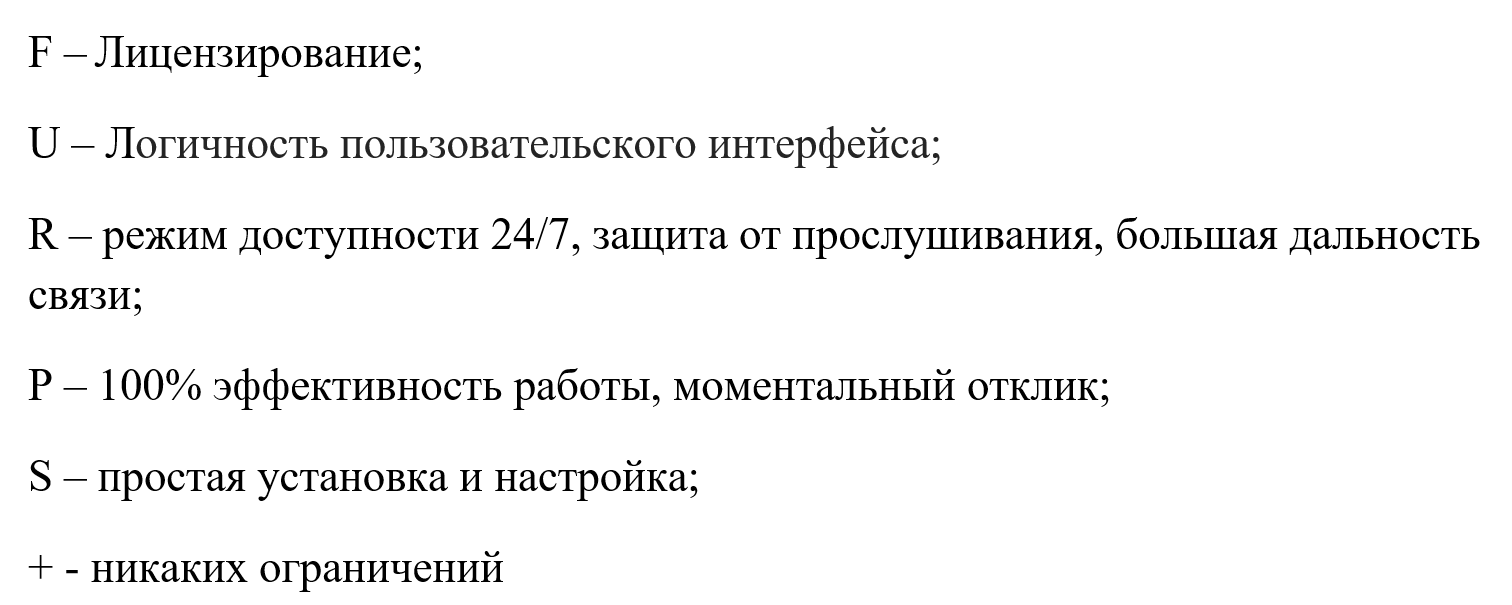


Рисунок 9 – FURPS+ для системы «Таксофон»

# 9 Результаты машинного тестирования программы

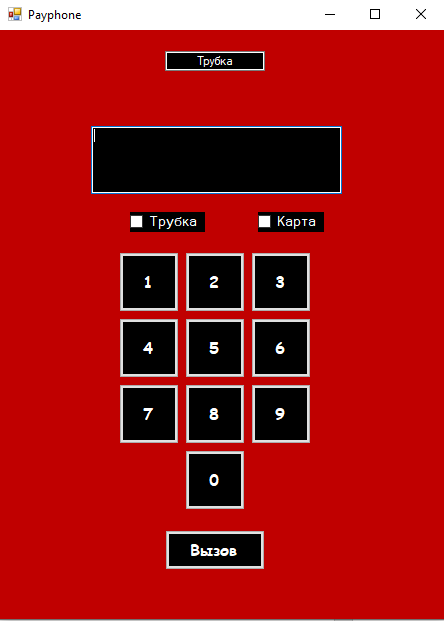


Рисунок 10 – Готовый к работе таксофон

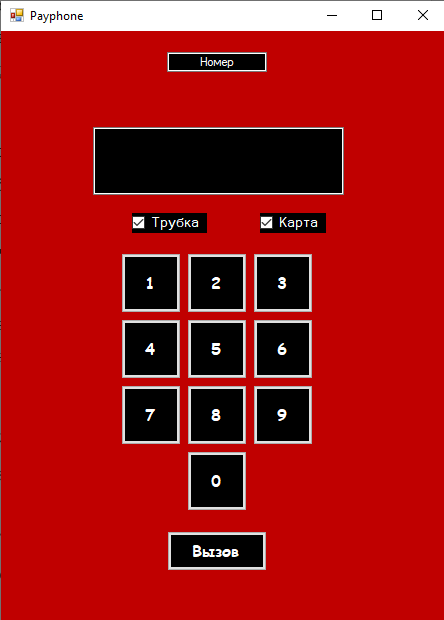


Рисунок 11 – Снята трубка и вставлена карта

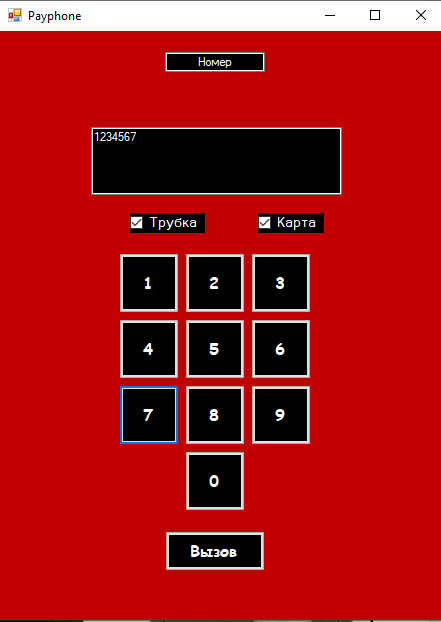


Рисунок 12 – Ввод номера

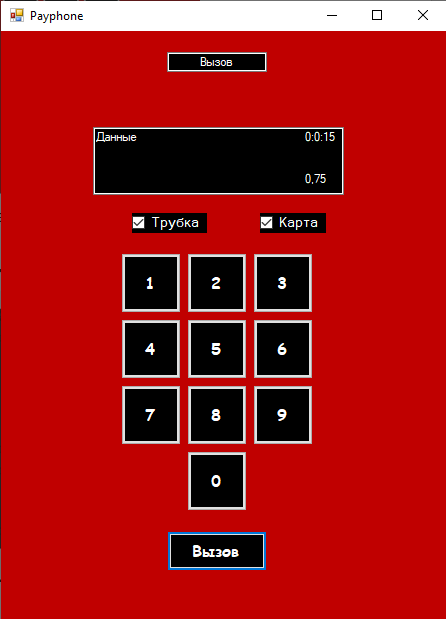


Рисунок 13 – Начало звонка с таймером

**10 Системные требования**

Таблица 1 – Системные требования программы

|  |  |
| --- | --- |
| Процессор | 2.5 ГГц |
| Оперативная память | 150 Мб |
| Монитор | 1920 x 1080 |
| Свободное место на носителе | 5 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: панель ввода номера телефона, трубка, приемник телефонных карт.

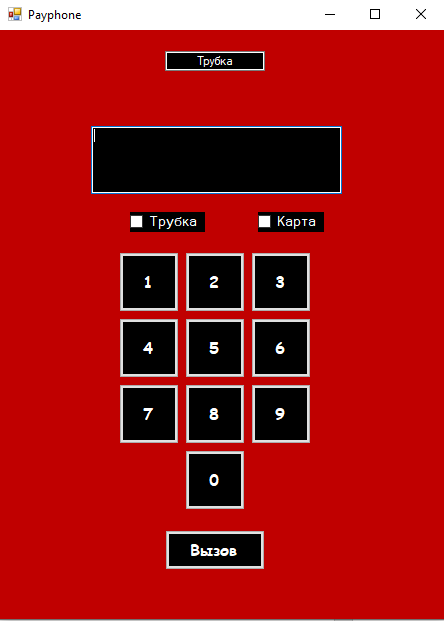


Рисунок 14 – Начальный интерфейс программы

Для начала работы пользователь должен снять трубку (галочка), это требует программа в самом верхнем окне, затем вставить карту (галочка), об этом уведомит программа в том же окне. Если на карте достаточно средств, то таксофон отправит запрос АТС на ячейку линии связи. Если от АТС приходит сообщения о свободной линии, то в верхнем окне появится уведомление о возможности набора номера. Следом вводит 7-ми значный номер и нажимает кнопку вызова. Начнется вызов другого абонента, об этом уведомляет верхнее окно. Если абонент отвечает, то начинается передача данных. Во время передачи данных активируется таймер и начинают сниматься деньги с телефонной карты. Каждые 15 секунд снимается 25 копеек.

Таксофон перестает работать в следующих случаях:

1. Трубка возвращена на исходную позицию.
2. Карта вынута из приемника.
3. На карте закончились деньги.
4. АТС не может предоставить линию связи.

**Заключение**

В результате выполнения данного курсового проекта была спроектирована система «Таксофон» на языке высокого уровня C#, позволяющая наглядно продемонстрировать работу всех её компонентов. Полученные диаграммы обладают простой и понятной, даже для человека незнакомого с данной областью, структурой, описывающей каждый аспект системы с различных сторон.

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 с уверенностью сказать, что полученная система «Таксофон» полностью соответствует современным стандартам безопасности и способна выполнять задуманные для неё задачи.

Были получены важные знания и практические навыки как в области использования объектно-ориентированных языков программирования в целом, так и в области построения диаграмм проектирования, отображающих поведение различных организационных структур.

**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 – Alex-KISLOV/Payphone. [Электронный ресурс]: - Режим доступа: <https://github.com/Alex-KISLOV/Payphone> (Дата обращения 16.05.2020).
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

# Диаграмма ганта

# Приложение В – Листинг программы

Таблица 2 – Характеристика переменных к заданию

|  |  |  |  |
| --- | --- | --- | --- |
| Имя переменной | Смысл переменной | Назначение переменной | Ограничения |
| x | Количество цифр на дисплее | Исходная | Целочисленная |
| m | Часы | Исходная | Целочисленная |
| s | Минуты | Исходная | Целочисленная |
| ms | Секунды | Исходная | Целочисленная |
| karSUMM | Сумма на карте | Исходная | Нет |
| rnd | Вероятность звонка | Вспомогательная | Целочисленная |
| f | Вероятность звонка | Вспомогательная | Целочисленная |
| a3 | Цифра 3 | Исходная | Целочисленная |
| a4 | Цифра 4 | Исходная | Целочисленная |
| a5 | Цифра 5 | Исходная | Целочисленная |
| a6 | Цифра 6 | Исходная | Целочисленная |
| a7 | Цифра 7 | Исходная | Целочисленная |
| a8 | Цифра 8 | Исходная | Целочисленная |

*Продолжение таблицы 2*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a9 | Цифра 9 | Исходная | Целочисленная | |
| a0 | Цифра 0 | Исходная | Целочисленная | |
| textBox1.Text | Дисплей | Исходная | Строковая | |
| textBox2.Text | Требования | Исходная | | Строковая |
| label1.Text | Таймер | Исходная | | Нет |
| label2.Text | Оставшаяся сумма | Исходная | | Нет |
| timer1.Enabled | Запуск таймера | Исходная | | Целочисленная |
| timer2.Enabled | Запуск таймера | Исходная | | Целочисленная |
| textBox1.MaxLength | Максимальное количество символов | Исходная | | Целочисленная |

Код программы:

namespace Payphone

{

public partial class Form1 : Form

{

int x;

public int m, s, ms;

public double karSUMM = 1;

public Form1()

{

InitializeComponent();

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e) // Трубка

{

if (Convert.ToString(textBox2.Text) == "Трубка")

{ textBox2.Text = "Карта"; }

if (checkBox1.Checked == false)

{

System.Windows.Forms.Application.Restart();

System.Environment.Exit(1);

}

}

private void checkBox2\_CheckedChanged(object sender, EventArgs e) // Карта

{

if (Convert.ToString(textBox2.Text) == "Карта")

{

Random rnd = new Random();

int f = rnd.Next(1,2);

if (f == 2)

{

textBox2.Text = "Занято";

}

if (f == 1)

{

textBox2.Text = "Данные";

}

}

if (Convert.ToString(textBox2.Text) == "Данные")

{

textBox2.Text = "Номер";

}

if ((checkBox2.Checked == false) || (Convert.ToString(textBox2.Text) == "Занято"))

{

System.Windows.Forms.Application.Restart();

System.Environment.Exit(1);

}

}

private void button1\_Click(object sender, EventArgs e)

{

string a1 = Convert.ToString(1);

textBox1.Text = textBox1.Text + a1;

x++;

}

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

{

string a2 = Convert.ToString(2);

textBox1.Text = textBox1.Text + a2;

x++;

}

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

{

string a3 = Convert.ToString(3);

textBox1.Text = textBox1.Text + a3;

x++;

}

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

{

string a4 = Convert.ToString(4);

textBox1.Text = textBox1.Text + a4;

x++;

}

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

{

string a5 = Convert.ToString(5);

textBox1.Text = textBox1.Text + a5;

x++;

}

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

{

string a6 = Convert.ToString(6);

textBox1.Text = textBox1.Text + a6;

x++;

}

private void button7\_Click(object sender, EventArgs e)

{

string a7 = Convert.ToString(7);

textBox1.Text = textBox1.Text + a7;

x++;

}

private void button8\_Click(object sender, EventArgs e)

{

string a8 = Convert.ToString(8);

textBox1.Text = textBox1.Text + a8;

x++;

}

private void button9\_Click(object sender, EventArgs e)

{

string a9 = Convert.ToString(9);

textBox1.Text = textBox1.Text + a9;

x++;

}

private void button10\_Click(object sender, EventArgs e)

{

string a0 = Convert.ToString(0);

textBox1.Text = textBox1.Text + a0;

x++;

}

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

{

textBox1.MaxLength = 7;

}

private void timer2\_Tick(object sender, EventArgs e) // Таймер за снятие денег

{

karSUMM -= 0.25;

string a9 = Convert.ToString(karSUMM);

label2.Text = "";

label2.Text = label2.Text + a9;

if (karSUMM == 0)

{

System.Windows.Forms.Application.Restart();

System.Environment.Exit(1);

}

}

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

{

ms += 1;

if (ms == 60)

{

ms = 0;

s += 1;

}

if (s == 60)

{

s = 0;

m += 1;

}

label1.Text = m.ToString() + ":" + s.ToString() + ":" + ms.ToString();

}

private void button11\_Click(object sender, EventArgs e) // Кнопка вызова

{

if (x == 7)

{

textBox2.Text = "Вызов";

}

if (x != 7)

{

textBox2.Text = "Ошибка";

}

if (Convert.ToString(textBox2.Text) == "Вызов")

{

Random rnd = new Random();

int f = rnd.Next(1,2);

if (f == 2)

{

textBox1.Text = "Занято";

}

if (f == 1)

{

textBox1.Text = "Данные";

timer1.Enabled = true;

timer2.Enabled = true;

m = 0; s = 0; ms = 0;

label1.Text = "0:0:0";

}

}

}

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

{

}

}

}