|  |  |
| --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  федеральное государственное АВТОНОМНОЕ образовательное учреждение высшего образования  «Национальный исследовательский ядерный университет «МИФИ» |  |
| **Димитровградский инженерно-технологический институт –**  филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский ядерный университет «МИФИ»  **(ДИТИ НИЯУ МИФИ)** |  |

**Кафедра информационных технологий**

**ОТЧЕТ**

**ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

**(ТЕХНОЛОГИЧЕСКАЯ)**

студента группы АС-51

Иванова Т. И.

(фамилия, инициалы)

Руководитель практики от предприятия: Наскальнюк А. Н.

Заместитель директора ООО «АИС Город»

(Ф. И. О. должность)

Руководитель практики от кафедры: Селиванова С.Е.

старший преподаватель

(Ф. И. О. учёная степень, звание, должность)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_г.

Димитровград, 2023

Содержание

[Введение 3](#_Toc145667632)

[1 Характеристика предприятия ООО «АИС Город» 4](#_Toc145667633)

[1.1 Основные функции, цели ООО «АИС Город» 4](#_Toc145667634)

[1.2 Программные продукты ООО «АИС Город» 5](#_Toc145667635)

[2 Практическая часть разработки услуги «Пополнение школьной карты» 7](#_Toc145667636)

[2.1 Постановка задачи. Основные требования к реализации 7](#_Toc145667637)

[2.2 Backend реализация в проекте личного кабинета 10](#_Toc145667638)

[2.3 Backend реализация в платежной системе 11](#_Toc145667639)

[2.4 Backend реализация в проекте эквайринга 12](#_Toc145667640)

[2.5 Frontend реализация в проекте личного кабинета 14](#_Toc145667641)

[2.6 Иные дополнения и изменения 16](#_Toc145667642)

[Заключение 19](#_Toc145667643)

[Библиографический список 20](#_Toc145667644)

# Введение

Школьная карта – это тариф по операциям физических лиц в «Кредит Урал Банк», предназначенный для оформления банковской карты на подростка от 14 лет либо для привязывания банковской карты ребенка в возрасте от 6 до 13 лет к личному счету одного из родителей. Обладатели карт имеют возможность осуществлять проезд в общественном транспорте по льготным тарифам, а так же имеют повышенный кешбэк на некоторые виды операций.

Производственная практика была пройдена в ООО «АИС Город». АИС Город — это один из лидеров в создании электронных решений для построения Умных городов, это разработчик программного обеспечения для автоматизации и мониторинга жилищно-коммунального хозяйства и смежных сфер. Программные продукты, разработанные компанией АИС Город, успешно используются рядом регионов Российской Федерации, полностью отвечают требованиям законодательства РФ и созданы в соответствии с реальными потребностями всех задействованных сторон.

Комплекс программ, разрабатываемый АИС Город, позволяет создать единую систему для оптимизации ЖКХ региона, повышения управляемости отрасли, достижения прозрачности финансовых потоков и снижения временных и моральных затрат населения.

При прохождении производственной практики были поставлены следующие задачи:

* Изучить общую характеристику ООО «АИС Город»;
* Изучить разрабатываемые программные продукты ООО «АИС Город»;
* Изучить документацию по поставленной задаче;
* Добавить реализацию пополнения школьной карты в проект личного кабинета;
* Добавить реализацию пополнения школьной карты в проект эквайринга;

# Характеристика предприятия ООО «АИС Город»

## Основные функции, цели ООО «АИС Город»

Полное наименование: Общество с ограниченной ответственностью «АИС ГОРОД».

Сокращённое наименование: ООО «АИС Город».

Основным видом деятельности компании является Разработка компьютерного программного обеспечения. Также ООО «АИС Город» работает еще по 13 направлениям. Организация насчитывает 2 учрежденные компании. Действует с 27.11.2003. Клиентами компании являются 23 региона России. Более 16 млн человек пользуются личными кабинетами и электронными сервисами, созданными ООО «АИС Город»[1].

Благодаря сотрудничеству с управляющими и ресурсоснабжающими организациями, органами государственной власти и местного самоуправления программные продукты полностью отвечают требованиям законодательства РФ и созданы в соответствии с реальными потребностями всех задействованных сторон. Программные продукты, разработанные компанией АИС Город, успешно используются и высоко ценятся рядом регионов Российской Федерации. Каждая информационная система предлагает реальное решение существующей проблемы отрасли жилищно-коммунального хозяйства.

Работа каждой из систем обеспечивает всех участников рынка ЖКХ достоверной и актуальной информацией, а в комплексе — системы формируют базовые сведения о населении, необходимые во многих отраслях, дают возможность создать консолидированный массив информации по жилому фонду и составу жильцов как по региону в целом, так и по отдельному муниципальному образованию [2].

Цели организации:

* Обеспечение информацией о домах, их конструктивных особенностях, их техническом состоянии. О количестве проживающих, об объемах и условиях оказания коммунальных услуг, их качестве, о начислениях и оплате, о работах на объектах жилищного фонда и коммунальной инфраструктуры. О планах осмотров, работ и их выполнении, о текущих плановых и капитальных ремонтах, о работах по благоустройству, о жалобах и обращениях граждан;
* Снижение временных трудовых затрат на сбор сведений о ЖКХ;
* Достижение прозрачности финансовых потоков;
* Повышение управляемости отрасли, повышение качества жилищно-коммунальных услуг и снижение социальной напряженности;
* Снижение финансовых затрат при оказании мер социальной поддержки при оплате ЖКХ и достижение адресности при оказании таких мер;
* Создание в единых диспетчерских с пунктами приема жалоб и обращений, контактными центрами для повышения качества услуг в ЖКХ;
* Создание «единого окна ЖКХ», внедрение единых квитанций.
* Снижение временных затрат на общение с поставщиками услуг, бухгалтериями, хождение по кассам;
* Повышение качества выполняемой персоналом работы за счет наличия механизма сквозного контроля;
* Повышение управляемости отраслью за счет наличия детальной достоверной информации;
* Снижение недовольства потребителей за счет наличия механизма контроля исполнительской дисциплины [2].

## Программные продукты ООО «АИС Город»

ООО «АИС Город» имеет множество проектов. Одними из самых популярных являются «Паспортный Стол», «РИАС ЖКХ», «АСКУЭ АИС Город. Приборный учет», «Система Начислений», «Центр Льгот», «Сова. АИС Город», «CRM ЖКХ "Панда"», а так же личные кабинеты поставщиков услуг ООО «РИЦ», АО «Ситиматик», ООО «ВИЦ», «МП ТРЕСТ ТЕПЛОФИКАЦИЯ», ООО «ЕИРЦ РБ», АО «ЕИРЦ ЛО», ООО «ЕРКЦ» [3] и др.

В рамках задачи осуществлялась работа с такими проектами как Личный кабинет оплаты ЖКХ города Магнитогорска, AIS.PAY.Acquiring, Платежная система. Администратор.

Личный кабинет оплаты ЖКХ города Магнитогорска – это сайт для оплаты жилищно-коммунальных услуг (ЖКУ), разработанный для АО «Кредит Урал Банк». Личный кабинет позволяет пользователям оплачивать услуги МП трест «Теплофикация», МП трест «Водоканал», ООО «ГКС МКД», ООО УК «Азимут», ООО «НОВАТЭК-ЧЕЛЯБИНСК», ООО «МЭК», Фонда капитального ремонта, вводить показания приборов учета и отслеживать данные по лицевым счетам.

AIS.PAY.Acquiring – это крупный проект, предназначенный для проведение безналичных платежей личных кабинетов, агентов и сервисов, разрабатываемых и поддерживаемых ООО «АИС Город».

Выводы по разделу:

Основным видом деятельности компании является Разработка компьютерного программного обеспечения. Программные продукты, разработанные компанией АИС Город, успешно используются и высоко ценятся рядом регионов Российской Федерации. ООО «АИС Город» имеет множество проектов. Личный кабинет оплаты ЖКХ города Магнитогорска – это сайт для оплаты жилищно-коммунальных услуг (ЖКУ), разработанный для АО «Кредит Урал Банк». AIS.PAY.Acquiring – это крупный проект, предназначенный для проведение безналичных платежей.

# Практическая часть разработки услуги «Пополнение школьной карты»

## Постановка задачи. Основные требования к реализации

При реализации услуги пополнение школьной карты заказчиками в лице АО Банк «КУБ» было предоставлено техническое задание (ТЗ). В рамках данного рассматривалось добавление данной услуги в личный кабинет https://mgn-city.ru/.

Содержание работ имело следующее содержание:

* Раздел «Школьная карта».
* Услуга «Пополнение школьной карты».
* Настройка платежной страницы, и протокола взаимодействия.

Для каждого раздела были выставлены требования. Для раздела школьная карта необходимо реализовать следующие задачи:

* Новый раздел должен быть доступен только для авторизованных пользователей ЛК.
* Расположить раздел в левом меню, под разделом «История платежей»
* Внутри раздела и для всех услуг внутри установить фон, в виде школьной карты (рис.2.1).
* Создать первую услугу нового раздела «Пополнение школьной карты».

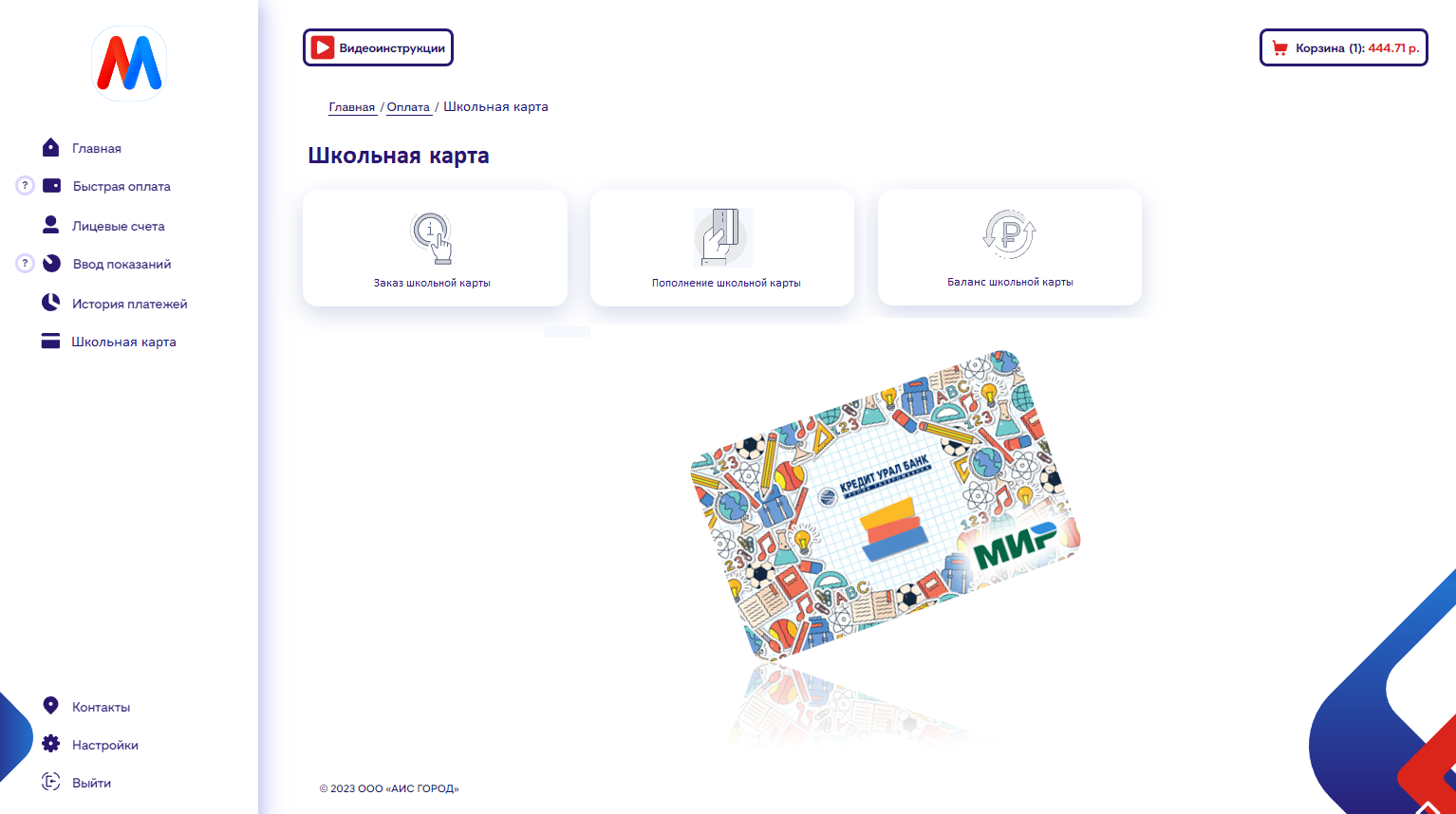


Рисунок 2.1 – Внешний вид раздела «Школьная карта» из ТЗ

Услуга по переводу с карты на карту уже была частично реализована в ЛК. Необходимо дополнить её следующим образом:

* При нажатии на кнопку «Пополнение школьной карты» должна выходить страница, где пользователь будет вводить номер карты, которую хочет пополнить.
* После ввода номера карты, указывается сумма, на которую необходимо пополнить школьную карту. Далее, выбирается с какой карты необходимо оплатить или вводит данные новой карты и нажимает кнопку «Оплатить» (рис.2.2).



Рисунок 2.2 – Внешний вид раздела «Пополнение школьной карты» из ТЗ

* При переходе на платежную страницу Банка, данные номера карты получателя, сумма перевода и данные карты отправителя, должны подтянуться автоматически (рис.2.3).

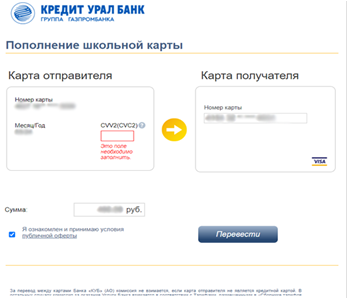


Рисунок 2.3 – Платежная страница Банка

Дополнительно заказчик попросил выполнить следующие требования:

* При нажатии пользователем кнопок «Оплатить» или «Добавить в корзину», необходимо осуществлять проверку на корректность введённого номера карты. Школьные карты начинаются со следующих префиксов:
  + 415432004
  + 4154320042
  + 4154320045
  + 4154320046
  + 4627170040
  + 4627170041
  + 4627170042
  + 4627170043
  + 220056017
* Проверять количество ввода цифр, их должно быть 16 (шестнадцать).
* В случае если номер введен не от школьной карты или введено неверное количество знаков, то:
  + подсветить поле «Номер школьной карты» красным цветом.
  + вывести уведомление об ошибке: «Введен некорректный номер школьной карты» (рис.2.4).



Рисунок 2.4 – Уведомление об ошибке

Помимо технического задания был предоставлен макет верстки в формате .cd, svg и png изображения, необходимые для реализации задачи.

Так же при реализации задачи необходимо соблюдать правила написания качественного кода [4], как общепринятых, так и правил оформления, принятых в ООО «АИС Город».

## Backend реализация в проекте личного кабинета

Для реализации поставленной задачи в проект был добавлен PayTransferCardController, и папка PayTransferCard. Метод Index в PayTransferCardController отвечает за отображение главной страницы раздела Школьная карта.

Для возможности отображения школьной карты в меню был добавлен новый раздел меню в ЛК с помощью специально реализованного интерфейса, доступного для администраторов сайта.

При вводе данных и нажатии кнопки оплатить происходил вызов основного метода оплаты, который производил редирект пользователя на страницу оплаты банка КУБ.

Особенностью данной реализации является необходимость передавать данные карты на страницу оплаты, что позволяет в дальнейшем отображать номер пополненной школьной карты в разделе «История платежей».

Особенностью так же можно назвать то, что при выборе услуги «Пополнение школьной карты» происходит переход сразу на страницу оплаты, а не на страницу с предварительным вводом номера карты и проверки существования данной карты, что привело к необходимости небольшого изменения кода. За отсутствие данной страницы отвечает флаг withoutfields, находящийся в таблицу pay/PaymentServices. При отрицательном его значении страницы с полями ввода не появляется.

Однако это порождает некоторую проблему, так как для отображения страницы оплаты необходимо отправлять на проверку номер лицевого счета. Чтобы обойти данное условие, был переработан метод редиректа на страницу оплаты.

## Backend реализация в платежной системе

В базу данных были добавлены новые записи, позволяющие реализовать оплату по данной услуге. С учетом особенностей реализации оплаты в ЛК и эквайринге, формально данная операция является именно оплатой а не переводом. Однако, при реализации данной оплаты, создание групповых операций не осуществляется.

Сначала было добавлено новое поле в таблицу GetAccountFields.

INSERT INTO lk.GetAccountFields

(

placeholder,

fieldName,

typeAccount,

inputType,

tooltip,

sort

)

VALUES

( ' Номер школьной карты ', -- placeholder - nvarchar(256)

' accountNumber ', -- fieldName - nvarchar(256)

null, -- typeAccount - nvarchar(256)

' text ', -- inputType - varchar(50)

null, -- tooltip - varchar(200)

0 -- sort - int

)

Так же была добавлена запись в таблицу PaymentGroupsService:

INSERT INTO lk.PaymentGroupsService

(

name,

groupId,

serviceCode,

controllerName,

imageService,

isCountersAvailable,

alias,

isVisible,

canEditSumPay,

isUseInTerminal,

sort

)

VALUES

( ' Перевод с карты на карту ', -- name - varchar(350)

13, -- groupId - bigint

119, -- serviceCode - bigint

' PayTransferCard ', -- controllerName - varchar(50)

NULL, -- imageService - varchar(100)

NULL, -- isCountersAvailable - bit

Перевод с карты на карту, -- alias - varchar(50)

1, -- isVisible - bit

NULL, -- canEditSumPay - bit

1, -- isUseInTerminal - bit

NULL -- sort - int

)

Далее была добавлена запись в таблицу Services.

INSERT INTO pay.Services

(

code,

typeId,

name,

paymentTypeId,

barcodePaymentTypeId,

reserveCode,

isClone,

shortName,

localDistrictID,

isChildEPD

)

VALUES

( 119, -- code - int

15, -- typeId - bigint

' Перевод с карты на карту', -- name - varchar(255)

60, -- paymentTypeId - bigint

0, -- barcodePaymentTypeId - bigint

'', -- reserveCode - varchar(100)

NULL, -- isClone - bit

' Перевод с карты на карту', -- shortName - varchar(255)

NULL, -- localDistrictID - int

NULL -- isChildEPD - bit

)

Для отображения услуги в личном кабинете так же были добавлены и изменены некоторые параметры в другой базе данных, использующей PostgreSQL.

## Backend реализация в проекте эквайринга

Для выполнения требований по проверке номера вводимой школьной карты было решено проводить проверку на 2 уровнях: в frontend части в личном кабинете с помощью инструментов javaScript и в backend части на стороне эквайринга. Для этого был разработан метод предоплатой проверки, который проводил проверку данных на валидность. Код данного метода представлен ниже.

protected override void PrePayMethod()

{

if (!string.IsNullOrWhiteSpace(model.accountNumber))

{

string regex = @"^415432004|^462717004[0-3]|^220056017|^220056019";

if (model.accountNumber.Length!=16)

{

throw new CreatePayPackException(null, CreatePayStatusEnum.Ошибка\_выполнений\_предоплатного\_метода\_для\_услуги, "Некорретно задан номер школьной карты. Номер карты состоит из 16 чисел.");

}

if (!Regex.IsMatch(model.accountNumber, regex, RegexOptions.IgnoreCase))

{

throw new CreatePayPackException(null, CreatePayStatusEnum.Ошибка\_выполнений\_предоплатного\_метода\_для\_услуги, "Некорретно задан номер школьной карты");

}

}

}

Для корректной отправки данных в банк был переработан запрос согласно предоставленной документации. В итоговой реализации запрос имел следующий вид:

$@"<?xml version=""1.0"" encoding=""UTF-8""?>

<TKKPG>

<Request>

<Operation>CreateOrder</Operation>

<Language>RU</Language>

<Order>

<OrderType>{orderType}</OrderType>

<Merchant>{merchant}</Merchant>

<Amount>{sum}</Amount>

<Currency>643</Currency>

<Description>{description}</Description>

{myUrl}

{addParams}

</Order>

</Request>

</TKKPG>

Так же в рамках данной задачи был исправлен поиск данных школьной карты.

{

orderType = "P2PTransfer";

sum = !string.IsNullOrWhiteSpace(sum) ? sum : "";

description = "Перевод с карты на карту";

if (Guid.TryParse(cardUID, out var guid))

{

try

{

using (var ais\_pay = new AISPayLKNewDataContext())

{

var operation = ais\_pay.SaveCardOperations.SingleOrDefault(s => s.cardGuid == guid);

if (operation != null)

{

cardUID = operation.cardBankId;

cardHolderName = operation.cardholderName ?? cardHolderName;

}

}

}

catch (Exception ex)

{

ex.WriteExceptionLog("Получение данных карты");

}

}

if (!string.IsNullOrWhiteSpace(recipientsCardNumber))

addParamTegs += $"<RPANBegin>{recipientsCardNumber}</RPANBegin><RPANLen>{recipientsCardNumber.Count()}</RPANLen>";

}

## Frontend реализация в проекте личного кабинета

Frontend разработка заключалась в реализации двух страниц в личном кабинете. Для реализации данной задачи были использованы навыки владения продвинутым уровнем html, css и js. Так же обходимо было иметь познания в области использования razor pages.

Первой страницей для реализации была страницы для выбора услуги пополнения школьной карты (рис. 2.5).

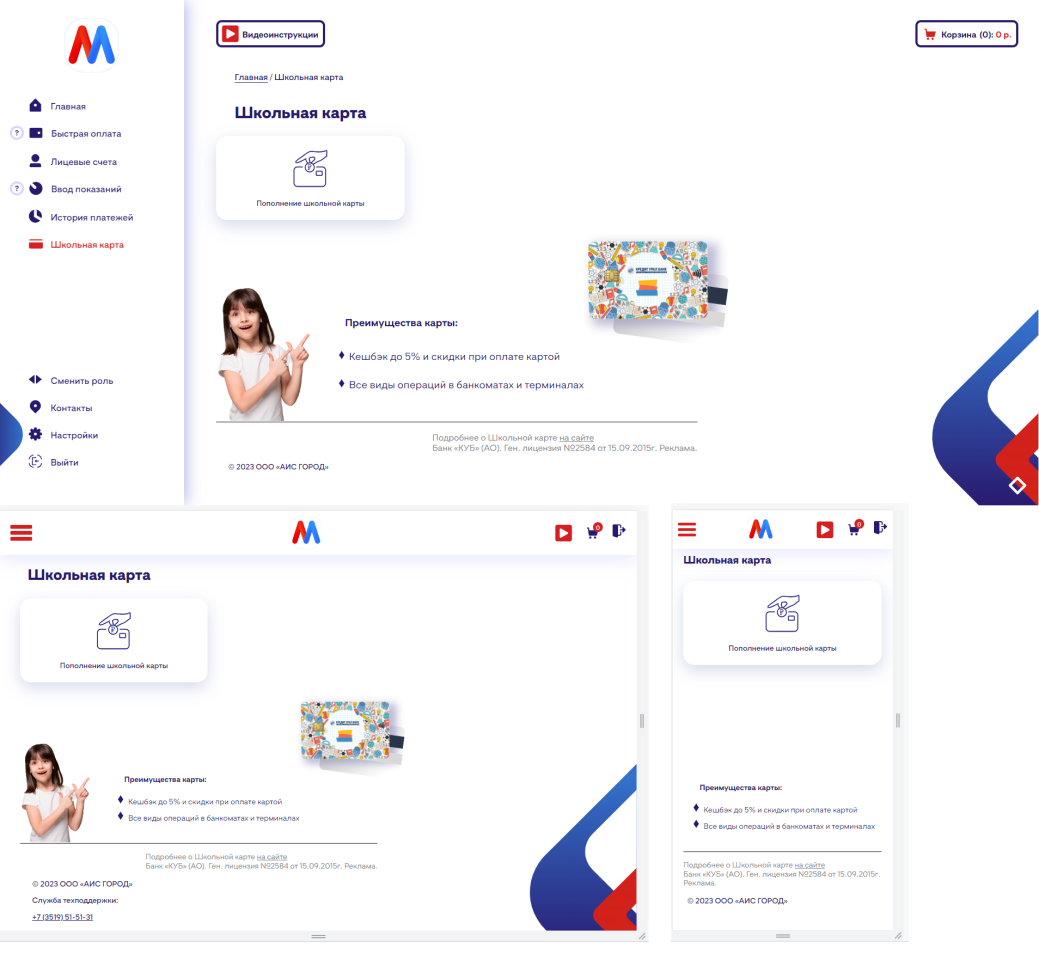


Рисунок 2.5 – Адаптивная верстка страницы «Школьная карта»

Немаловажным требованием к реализации является возможность использования мобильного приложения на любом типе устройств, поэтому особое внимание уделялось адаптивной верстке.

Так же в процессе выполнения задачи заказчиками были изменены требования к внешнему виду страницы.

Аналогично сверстана и страница перевода по школьной карте (рис.2.6)

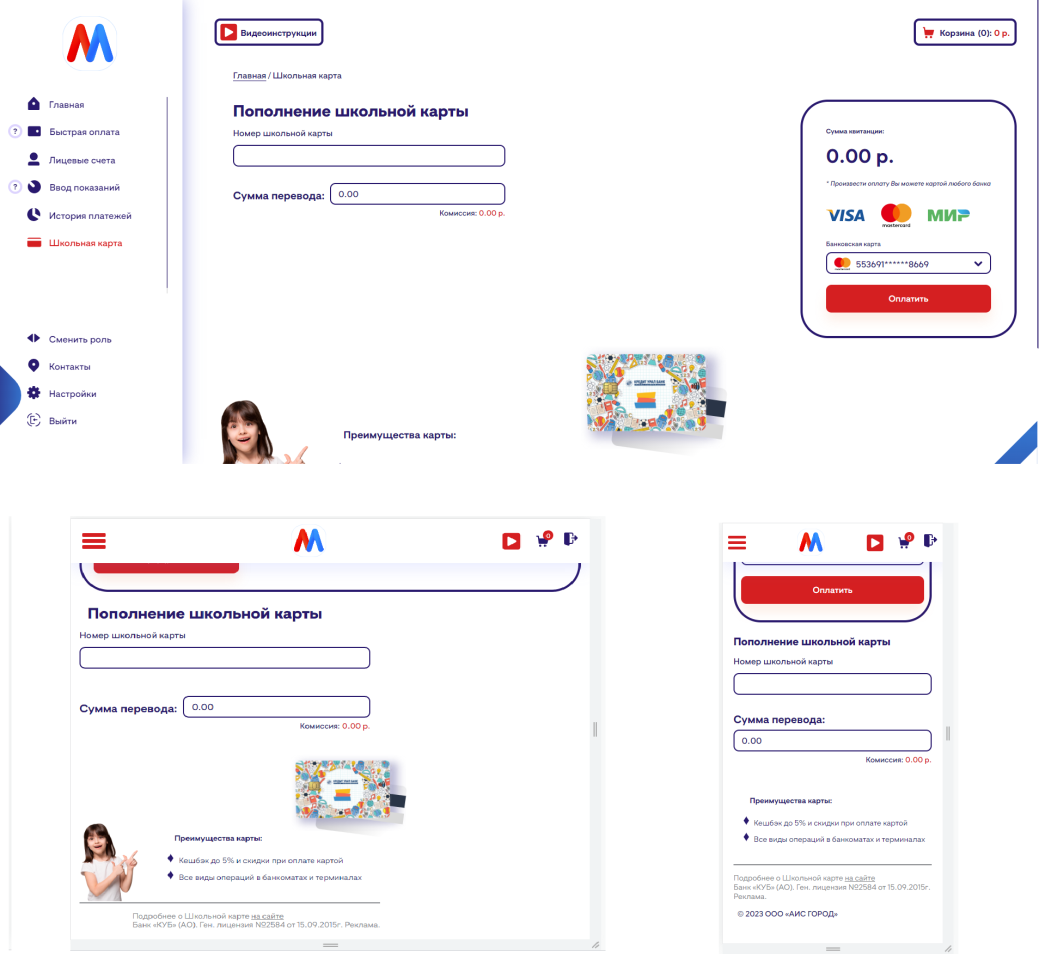


Рисунок 2.6 – Адаптивная верстка страницы «Школьная карта»

Страница, как и все страницы оплаты на данном сайте состоит из 2 частей: блок оплаты, который используется во всех страницах, и частичным представлением с полями для ввода данных по конкретной услуге.

Необходимым условием так же является валидация на стороне пользователя, которая была выполнена с использованием библиотеки jquery validation версии 1.19.5.

Код скрипта представлен ниже:

<script>

$(document).ready(function() {

$.validator.addMethod(

"regex",

function(value, element, regexp) {

var re = new RegExp(regexp);

return this.optional(element) || re.test(value);

},

"\* Указаны некорректные данные"

);

$("[name='PaymentForm']").validate();

$('#accountNumber').rules('add', {

required: true,

minlength: 16,

maxlength: 16,

regex: /^415432004|^462717004[0-3]|^220056017|^220056019/,

messages: {

required: "\* Это обязательное поле",

regex: "\* Указаны некорректные данные",

minlength: "\* Номер карты состоит из 16 чисел",

maxlength: "\* Номер карты состоит из 16 чисел"

}

});

});

</script>

## Иные дополнения и изменения

В процессе разработки было выявлено, что при тестировании задачи на начальных этапах не было учтено, что при оплате через личный кабинет все платежи добавлялись в таблицу BeforePayUserServices. Это в дальнейшем вызывало их добавление в регулярные платежи. Для переводов на школьную карту нет необходимости добавлять их в список регулярных платежей, поэтому было добавлено условие, не допускающее добавление платежей данного типа в таблице BeforePayUserServices.

Однако, ранее оплаченные платежи уже были добавлены в таблицу регулярных платежей UserServices. Их удаление не давало результата, так как со временем они снова добавлялись в данную таблицу. Было исследовано данное явление и выяснено, что обновление данных в данной таблице происходит каждые 10 минут.

recurringJobManager.AddOrUpdate<PaymentServicesSender>(

"Обновление статуса платежей",

x => x.UpdatePaymentStatus(),

"\*/10 \* \* \* \*"

); //каждые 10 минут

Данный код вызывал каждые 10 минут функцию GetPayPacksWithStatusForPayment (), имеющую следующий код:

var query = @"SELECT ""packPaymentId""

FROM pay.""PackElements""

WHERE status = 1 OR (status = 2 AND (""groupOperId"" IS NULL OR ""groupOperId"" = 0))

;";

using (var connection = \_context.CreateConnection())

{

connection.Open();

return connection.Query<string>(query).ToList();

}

Так как успешно проведенные переводы имели статус 2 а так же не имели идентификатора групповой операции, то они каждый раз подходили под условие.

Обновление данных в таблице PackElements вызывало триггер BindRegPay, имеющий следующий код.

BEGIN

IF (NEW."status" = 2)

THEN

INSERT INTO pay."UserServices"("UserId", "Created", "Code", "Provider", "AccountNumber", "Info", "TypeAccount", "PaymentGroupId")

SELECT pa."userId", NOW(), ps."code", ps."provider", pe.accountnumber, bfus."Info", bfus."TypeAccount", null FROM pay."PackElements" pe

INNER JOIN pay."Packs" pa on pe."packPaymentId" = pa."paymentId"

INNER JOIN pay."PaymentServices" ps on pe."serviceCode" = ps."code"

INNER JOIN pay."BeforePayUserServices" bfus on new."Id" = bfus."PackElementId"

WHERE pe."Id" = new."Id" AND (pa."userId" != null OR pa."userId" != '') AND

NOT EXISTS(select 1 from pay."UserServices"

WHERE "UserId" = pa."userId" AND "Code" = ps."code" AND "AccountNumber" = pe.accountnumber

) RETURNING "Id" INTO "userServicesId";

IF ("userServicesId" IS NOT NULL)

THEN

INSERT INTO pay."UserServiceAdditionalFields"(

"UserServiceId", "Key", "Value")

SELECT "userServicesId", bpaf."Key", bpaf."Value" FROM pay."BeforePayAdditionalFields" bpaf

INNER JOIN pay."BeforePayUserServices" bfus on new."Id" = bfus."PackElementId"

WHERE bfus."PackElementId" = new."Id" AND bpaf."BeforePayUserServicesId" = bfus."Id";

END IF;

END IF;

RETURN NEW;

END

Так как проведённые ранее платежи подходили по всем условиям, то они добавлялись в таблицу UserServices.

Для решения данной проблемы было решено удалить 2 тестовых перевода из таблицы BeforePayUserServices, в которую все последующие платежи даже не добавлялись.

Так как данное условие INNER JOIN pay."BeforePayUserServices" bfus on new."Id" = bfus."PackElementId" перестало выполнятся, то платежи перестали добавляться в раздел регулярных платежей.

Выводы по главе

В рамках данной работы были добавлены новые записи в таблицах БД на языке Transact-SQL, PostgreSQL. Был выполнен анализ функции обновления статуса платежей, изучена работа триггеров на таблицах БД.

Были разработаны методы для передачи введенной пользователем в личном кабинете информации на страницу оплаты банка. Реализована предоплатная валидация на стороне АПИ.

Был разработан пользовательский интерфейс страниц услуги и оплаты по услуге «Школьная карта».

Была реализована валидация на стороне пользователя.

По результатам данной работы все условия, поставленные в техническом задании, были успешно выполнены.

.

# Заключение

В процессе прохождения практики в отделе биллинговых и платежных систем компании ООО «АИС Город» были изучены:

* организационная структура ООО «АИС Город»;
* общая характеристика ООО «АИС Город»;
* программный продукт «AIS.PAY.Acquiring»;
* работа с базами данных PostgreSQL.

В ходе практики приобретены необходимые для дальнейшей профессиональной деятельности навыки. А именно работа в коллективе, умение быстро реагировать на поставленную задачу, прогнозировать результаты действий. Поставленные задачи имели разнообразный характер: верстка web-страниц, работа с функционалом (фронтенд и бэкенд), работа с базами данных.

Были закреплены навыки работы с базами данных, проектами ASP Net MVC, языками программирования C#, JavaScript, PostgreSQL, Transact-SQL.

В рамках данной работы в личный кабинет оплаты ЖКУ города Магнитогорска была добавлена услуга «Перевод по школьной карте» и раздел «Школьная карта», которые полностью соответствуют заявленным в техническом задании требованиям.

# Библиографический список

1. АИС Город, ООО [Электронный ресурс]: Тензор, CБИС. Режим доступа: https://sbis.ru/contragents/7302026625/730201001 (дата обращения: 05.07.2023).
2. О Компании «АИС Город» [Электронный ресурс]: АИС ГОРОД. Режим доступа: https://www.aisgorod.ru/about/ (дата обращения: 04.07.2023).
3. Программное обеспечение для управления ЖКХ [Электронный ресурс]: АИС ГОРОД. Режим доступа: https://www.aisgorod.ru/products/ (дата обращения: 04.07.2023).
4. 15 правил написания качественного кода [Электронный ресурс]: Tproger. Режим доступа: https://tproger.ru/translations/15-rules-for-writing-quality-code/ (дата обращения: 04.07.2023).