[**Доступ к тестовой базе** 2](#_Toc434828231)

[**1. Сущность «Пользователь»** 2](#_Toc434828232)

[**Сущность «Договор»** 3](#_Toc434828233)

[**3. Сущность «Информация по пакетам»** 3](#_Toc434828234)

[**4. Сущность «Расходы за период»** 4](#_Toc434828235)

[**5. Сущность «Подключенные услуги»** 4](#_Toc434828236)

[**6. Сущность «Платежи»** 5](#_Toc434828237)

[**7. Сущность «Обещанный платеж»** 6](#_Toc434828238)

[**8. Сущность «Детализация услуг»** 6](#_Toc434828239)

Необходимо разработать личный кабинет абоннета.

Дизайн сайта уже разработан.

**Доступ к тестовой базе****:**

СУБД: Oracle 11

Host: hub1.tarifer.ru

Port: 22521

SID: DEVK7

User: K7\_LK

Password: K7\_LK\_USER6

**Описание сущностей:**

**1. Сущность «Пользователь»**

Поля:

* 1. Имя
  2. Е-mail
  3. День рождение
  4. Ключевое слово
  5. Паспортные данные
  6. Пароль

Методы:

1. Сохранить личные данные
2. Изменение пароля
3. Регистрация пользователя
4. авторизация

Страницы сайта:

1. **new-password.html:**
2. Работа с капчей: gem galetahub-simple\_captcha
3. Запрос/сброс пароля : функция reset\_password
4. **login.html:**
5. Кнопка войти: берет данные из представления LK\_USERS
6. USSD команда: K7\_LK.USSD\_RESET\_PASSWORD, Так же необходимо забить в таблицу ussd\_leafs текст запрос: «select K7\_LK.USSD\_RESET\_PASSWORD (p\_phone ) from dual» (**не реализуется со стороны личного кабинета!**)
7. **settings.html** :
8. Сохранение/ обновление информации о пользователе: SET\_USER\_INFO
9. Получение информации о пользователе:

select NAME ,

EMAIL,

BirthDay,

PASSPORT,

KeyWORD

from lk\_users

**2. Сущность «Договор»**

Поля:

1. Номер телефона
2. Баланс
3. Статус
4. Тариф
5. Система оплаты

Методы:

1. Заблокировать номер
2. Обновить баланс
3. Подробнее о тарифе

Страницы сайта:

**index.html:**

1. Баланс: поле balance из lk\_users (см. Пункт Ваш тариф и Система оплаты)
2. Кнопка обновить(рядом с балансом): GET\_ABONENT\_BALANCE (pPhoneNumber)
3. Статус: get\_phone\_status (pPhoneNumber)
4. Заблокировать номер: LOCK\_PHONE (pPhoneNumber)
5. Абон.плата по аналогии с detailing.html
6. Ваш тариф и Система оплаты

select TARIFF\_NAME, --название тариыфа

contract\_type\_name, -- Система оплаты

balance --баланс

from lk\_users

where login = :pLogin;

**tariff.html**

Описание тарифного плана берется из API клиента.

Документация по API: <http://git.ditrade.ru/Serafim/diadmin-api/blob/master/docs/native.md>

Ссылка для получениея описания требуемого тарифа: <http://tariff.ditrade.ru/api/tariffs?crm_uuid=param_values> ,

где param\_values - значение crm\_uuid из lk\_users.CRM\_UUID

**3. Сущность «Информация по пакетам»**

Поля:

1. Название пакета
2. Объем пакета
3. Текущее значение
4. Остаток

Методы:

1. Получить остаток по пакетам

Страницы сайта:

**index.html:**

1. Информация по пакетам услуг:

select CURR\_VALUE,--текущее значение

INITIAL\_SIZE,--начальный размер услуги

UNIT\_TYPE, --тип слуги

REST\_NAME --название услуги

from table(TARIFF\_RESTS\_TABLE(:pPhone))

order by DECODE(UNIT\_TYPE, 'VOICE', 1, 2), ROWNUM

**4. Сущность «Расходы за период»**

Поля:

1. Абон. плата по тарифу
2. Тип вызова (смс, ммс, интернет, звонки, прочие услуги)
3. Направление вызова (исходящий, входящий)
4. Стоимость вызова
5. Длительность вызова

Методы

* 1. Получение расходов за текущий период

Страницы сайта:

**index.html:**

1. Расходы текущего периода cм detailing.html.

**detailing.html**

1. Расходы за период

**5. Сущность «Подключенные услуги»**

Поля:

1. Название
2. Статус (подключена/ отключена)
3. Абоненсткая плата в сутки
4. Дата подключения
5. Возможность отключения

Методы:

1. Подключённые услуги:
2. Список опций:

select TARIFF\_OPTION\_ID, --ИД опции

NAME, -- название

TURN\_ON\_DATE, -- дата подключения

TURN\_OFF\_DATE, --дата отключения

MONTHLY\_COST, -- стоимость в месяц

CAN\_BE\_TURNED\_BY\_ABONENT--признак того что опция может быть подключина

from table(TURNED\_OPTIONS(:pUSER\_id)) order by TURN\_ON\_DATE, NAME

1. Услуги доступные для отключения/подключения абоненту

Select

TARIFF\_OPTION\_ID, --ид поции

CATEGORY\_NAME, --название категории

NAME, --наименование опции

TURN\_ON\_COST, --стоимость подключения

MONTHLY\_COST,--стоимость в месяц

DESCRIPTION-- комментарий

from table(AVAILABLE\_OPTIONS\_FOR\_ABONENT(:pUSER\_id))

order by CATEGORY\_NAME, MONTHLY\_COST, NAME

1. Отображение услуг пользователю:  
   def get\_available\_options

@my\_services = @user.get\_services # Список опций

@available\_options = @user.available\_options# Услуги доступные для отключения/подключения абоненту

# Исключим из списка доступных уже подключенные (и не отключенные)

my\_ids = @my\_services.select{|s| s[:end\_date].nil? }.map{|s| s[:id] }

@available\_options.delete\_if{|option| my\_ids.index(option[:id]) }

еnd

Страница сайта **index.html.**

**6. Сущность «Платежи»**

Поля:

1. Дата
2. Сумма
3. Способ платежа

Методы:

* 1. Получение списка платежей:  
      select

ID, --ид

USER\_ID, --ид пользователя

phone\_number,--номер телефона

payment\_sum,-- сумма платежа

payment\_date,--дата платежа

payment\_remark,--примечание к платежу

payment\_type\_name--тип платежа

from PAYMENTS

where USER\_ID = :pUser\_ID

Страница сайта **payments.html**

**7. Сущность «Обещанный платеж»**

Поле:

1. Сумма платежа

Методы:

1. Добавить платеж:

Result := add\_promised\_payment(pPhone, :promised\_payment\_sum);

Страницы сайта:

**promised-payment.html**

**8. Сущность «Детализация услуг»**

Поля:

1. Дата вызова
2. Тип услуги
3. Направление
4. Собеседник
5. Объем услуг
6. Стоимость

Методы:

1. Получение детализации:  
   case plsql.get\_hot\_billing\_month\_source(year\_month)

when 'DISK'

# todo: Сделать контроль ошибок (возникает при сбоях запросов к роботу)

data = plsql.get\_phone\_detail(self.id, year\_month)

data = data.split("\n") if data.present?

when 'TABLE'

sql = "SELECT \* FROM TABLE(GET\_PHONE\_DETAIL\_FROM\_CALL(#{connection.quote(login)}, #{connection.quote(year\_month)}))"

data = connection.select\_rows(sql).map(&:first)

else

data = nil

end

result = DetailArray.new

if data.present?

data.each do |row|

fields = row.split("\t")

fields.shift

fields[2] = decode\_service\_type(fields[2])

# Стоимость

fields[6].sub!(',', '.') if fields[6].present?

fields[7] = fields[7]=='1' ? 'Роуминг' : ''

fields[11] = decode\_bs(fields[10].to\_i)

result << Detail.new(fields)

end

end

result.user = self

result.year\_month = year\_month

result

1. Получение расходов за период:  
   def get\_statistics(year\_month)

phone = login

sql = "SELECT ROW\_IDENT, NAME, VALUE FROM TABLE(GET\_PHONE\_CURRENT\_STATS(#{connection.quote(phone)}, #{connection.quote(year\_month)}))"

connection.select\_rows(sql).map do |row|

{row\_ident: row[0], name: row[1], value: row[2]}

end

end

1. Получить абон.плату по тарифу:  
   1. Получение расшифровки баланса:

def balance\_rows

sql = "select ROW\_DATE, ROW\_COST, DESCRIPTION from table(CALC\_BALANCE\_ROWS\_TABLE(#{connection.quote(login)})) order by row\_date desc"

connection.select\_rows(sql).map do |row|

{date: row[0], cost: row[1], description: row[2]}

end

end

1. Получение абон платы:  
   def get\_cur\_month\_payments

current\_month = Date.today.strftime('%Y%m')

subscriber\_payment = 0

other\_charges = 0

balance\_rows.each do |row|

# Только начисления текущего периода

if row[:date].strftime('%Y%m') == current\_month

case row[:description]

when /^Оплата/

# Оплату игнорируем

when /Абонплата \(/

# Оплата по тарифу

subscriber\_payment -= row[:cost]

else

other\_charges -= row[:cost]

end

end

end

[subscriber\_payment, other\_charges]

end

Страница сайта **detailing.html**