**Описание основных процедур,**

**включающих взаимодействие с системой Ариадна (oracle)**

**1. Авторизация.**

**1.1 По логину и паролю.**

1) пользователь вводит логин (phone\_number) и пароль;

2) запрос в oracle на существование и активность логина:

наименование хранимой процедуры oracle XXX\_001

передаваемые в oracle данные (входные параметры) – {“phone\_number”: string};

получаемые из oracle данные (выходные параметры)

при существовании логина:

{

“error”: null

“result”: {

“oracle\_user\_id”: integer,

“is\_active”: bool

}

}

при отсутствии логина:

{

“error”: “Login does not exist”

“result”: null

}

3) если error == null и is\_active == True

валидация пароля в локальной базе;

4) если валидация пароля успешна, то выдача secret\_token;

**1.2 По телефону (Flash Call)**

1) пользователь вводит номер телефона;

2) запрос в oracle на существование и активность логина:

наименование хранимой процедуры oracle XXX\_001

передаваемые в oracle данные (входные параметры) – {“phone\_number”: string};

получаемые из oracle данные (выходные параметры)

при существовании логина:

{

“error”: null

“result”: {

“oracle\_user\_id”: integer,

“oracle\_card\_id”: integer,

“oracle\_personal\_area\_id”: integer,

“phone\_number”: string,

“first\_name”: string,

“last\_name”: string,

“patronymic”: string,

“birth\_date”: string,

“gender”: string,

“email”: string,

“additional\_phone\_number”: string,

“is\_active”: True

}

}

при отсутствии логина:

{

“error”: “Login does not exist”

“result”: null

}

3) если error == null и is\_active == True

- генерация кода;

- вызов API Flash Call;

4) получение от пользователя кода, полученного через Flash Call;

5) сравнение сгенерированного на шаге 3) кода с кодом,полученным от пользователя,

6) если сравнение True, то выдача secret\_token;

7) если сравнеине True, а пользователя в базе нет, то запись пользователя в базу.

**2. Регистрация**

**2.1 По логину и паролю**

1) пользователь заполняет требуемые поля, включая логин (phone\_number) и дважды пароль;

2) валидация пароля по заданным критериям;

3) при успешной валидации пароля запрос в oracle на регистрацию нового пользователя:

наименование хранимой процедуры oracle XXX\_002

передаваемые в oracle данные (входные параметры)

{

“phone\_number”: string,

“first\_name”: string,

“last\_name”: string,

“patronymic”: string,

“birth\_date”: string,

“gender”: string,

“email”: string,

“additional\_phone\_number”: string

}

получаемые из oracle данные (выходные параметры):

при существовании логина в системе oracle:

{

“error”: “Login already exists”

“result”: null

}

при отсутствии логина в системе oracle:

{

“error”: null

“result”: {

“oracle\_user\_id”: integer,

“oracle\_card\_id”: integer,

“oracle\_personal\_area\_id”: integer,

“phone\_number”: string,

“first\_name”: string,

“last\_name”: string,

“patronymic”: string,

“birth\_date”: string,

“gender”: string,

“email”: string,

“additional\_phone\_number”: string,

“is\_active”: True

}

}

3) если “error”: null, то уведомление пользователя об успешной регистрации:

4) новый пользователь записывается в локальнуб базу.

**2.2 По телефону (Flash Call)**

1) пользователь заполняет требуемые поля, включая логин (phone\_number) и дважды пароль (password);

2) валидация пароля по заданным критериям;

3) при успешной валидации пароля переходим передаче пакета данных для регистрации пользователя на бэкэнд;

4) на бэкэнде - генерация кода, вызов API Flash Call;

5) получение от фронтэнда кода, полученного потльзователем от Flash Call;

6) сравнение сгенерированного на шаге 4) кода с кодом,полученным от пользователя;

7) если сравнение True, то проверка наличия пользователя в базе;

8) если пользователь c таким email или phone\_number уже существует в базе, то уведомление пользователя о невозможности повторной регистрации (в респонзе query, передающей код Flash Call);

9) если пользователь в базе отсутствует, то он записывается в базу.

**3. Смена пароля с использоанием Flash Call (стандартная)**

Производится без взаимодействия с системой oracle.

**4. Смена логина (основного телефона)**

**4.1 Оба телефоных номера доступны пользователю**

1) потльзователь авторизован по старому номеру или по логину и паролю

2) пользователь вводит новый номер телефона;

3) выполняется Flash Call на новый номер

3) если результат Flash Call == True, то в систему oracle выполняется запрос на изменение логина (основного номера)

наименование хранимой процедуры oracle XXX\_003

передаваемые в oracle данные (входные параметры)

{

“oracle\_user\_id”: integer,

“phone\_number”: string

}

получаемые из oracle данные (выходные параметры):

при существовании нового логина в системе oracle:

{

“error”: “New phone number is already registered, input other number”

“result”: null

}

при отсутствии нового логина в системе oracle:

{

“error”: null

“result”: {

“oracle\_user\_id”: integer,

“phone\_number”: string

}

}

в поле result[“phone\_number”] возвращается новый логин.

4) новый логин поьлзователя записывается в локальную базу.

**4.2 Старый телефоный номер не доступен пользователю (например, утерян)**

4.2.1 Пользователь авторизуется по логину и паролю и далее выполняются действия п.4.1

4.2.2 Пользователь забыл пароль и не может авторизоваться по логину и паролю.

1) выполнить процедуру восстановления пароля через email, указанный при регистрации.

2) выполняются действия п.4.1

4.2.3 Пользователь меняет логин при личном визите в медцентр

1) производится очный процесс идентификации личности путем предъявление пользователем документа, удостоверяющего его личность;

2) операция выполняется оператором непосредственно в системе oracle.

3) изменения в локальной базе отражаются при помощи процедуры синхронизации (описана ниже).

**5. Смена пароля через email, указанный при регистрации (дополнительная)**

Производится без взаимодействия с системой oracle.

**6. Запись на прием (получение номерков, бронирование, освобождение, закрепление)**

**6.1 Получение раписания доступных услуг (номерков)**

1) пользователь выбирает:

- услуга – обязательный параметр;

и/или;

- медицинский центр/центры;

и/или

- ФИО врача;

и/или

- дату;

и/или

- время;

2) в систему oracle выполняется запрос на получение доступных услуг (номерков)

наименование хранимой процедуры ХХХ004;

входные параметры:

{

“oracle\_center\_id”: integer,

“oracle\_service\_id”: integer, ***- обязательный параметр***

“oracle\_doctor\_id”: integer,

“needed\_date”: string,

“needed\_time”: string

}

выходные данные:

при успешном ответе

{

“error”: null,

“result”:[

{

“oracle\_rnumb\_id”: integer,

“oracle\_center\_id” integer,

“oracle\_department\_id”: integer,

“oracle\_doctor\_id”: integer,

“oracle\_service\_id”: integer,

“rnumb\_datetime”: string,

“rnumb\_duration”: integer ***– длительность в минутах***

“description”: text

},

{

…...

}

]

}

при неуспешном ответе

{

“error”: string, *- содержит текст ошибки*

“result”: null

}

**6.2 Временное бронирование номерка**

1) пользователь выбирает номерок;

2) в систему oracle направляется запрос на временное (до момента оплаты или до истечения таймаута, выделенного для оплаты) бронирование номерка;

наименование хранимой процедуры ХХХ005;

входные параметры

{

“oracle\_rnumb\_id”: integer,

“oracle\_user\_id”: integer

}

выходные параметры

при положительном ответе

{

“error”: null,

“result”: “ок”

}

при отрицательном ответе

{

“error”: string, ***– текст ошибки***

“result”: null

}

3) интервал времени бронирования задается в конфиге. По истечении данного интервала номерок освобождается от брони в системе oracle автоматически. В локальной базе запись о бронировании номерка перестает существовать также по истечению интервала бронирования.

**6.3 Закрепление номерка за пользователем после оплаты или при наличии у пользователя разрешения оплачивать позже (например, при посещении):**

1) пользователь оплатил услугу и от банка получено подтверждение оплаты услуги,

или

аванс пользователя больше или равен стоимости услуги и активна кнопка “Оплатить в счет аванса”,

или

у пользователя есть абонемент. Остаток по которому больше или равен стоимости услуги и активна кнопка “Оплатить в счет абонемента”,

или

активна кнопка “Оплатить наличными при посещении”.

2) при получении подтверждения банком или нажатии любой из кнопок, перечисленных в п.п.1) в систему oracle направляется запрос на закрепление номерка:

наименование хранимой процедуры ХХХ006

входные параметры

при оплате через банк

{

“oracle\_rnumb\_id”: integer,

“oracle\_user\_id”: integer,

“bank\_transactions\_id”: string,

“pay\_by\_balance”: False,

“pay\_by\_subscription”: False,

“pay\_on\_visit”: False

},

при оплате за счет аванса

{

“oracle\_rnumb\_id”: integer,

“oracle\_user\_id”: integer,

“bank\_transactions\_id”: null,

“pay\_by\_balance”: True,

“pay\_by\_subscription”: False,

“pay\_on\_visit”: False

}

при оплате за счет абонемента

{

“oracle\_rnumb\_id”: integer,

“oracle\_user\_id”: integer,

“bank\_transactions\_id”: null,

“pay\_by\_balance”: False,

“pay\_by\_subscription”: True,

“pay\_on\_visit”: False

}

при оплате при посещении

{

“oracle\_rnumb\_id”: integer,

“oracle\_user\_id”: integer,

“bank\_transactions\_id”: null,

“pay\_by\_balance”: False,

“pay\_by\_subscription”: True,

“pay\_on\_visit”: True

}

выходные данные

при положительном ответе

{

“error”: null,

“result”: “ok”

}

при отрицательном ответе

{

“error”: string, - текст сообщения об ошибке

“result”: null

}

**6.4 Освобождение номерка (отмена записи)**

1) пользователь нажимает кнопку “Отменить запись”;

2) в систему oracle направляется запрос на отмену номерка

наименование хранимой процедуры ХХХ007;

входные параметры:

{

“oracle\_rnumb\_id”: integer,

“oracle\_user\_id”: integer

}

выходные параметры

при положительном ответе

{

“error”: null,

“result”: “ok”

}

при отрицательном ответе

{

“error”: string, ***- текст сообщения об ошибке***

“result”: null

}

**6.5 Перенос записи (номерка)**

1) пользователь нажимает кнопку “Перенести запись”:

2) выполняются действия по отмене записи по п.п. 6.4;

3) выполняются действия по получению доступных номерков по выбранной пользователем услуге по п.6.1

4) выполняются действия по закреплению номерка за пользователем по п.6.3.

**7. Запрос текущего баланса**

1) пользователь переходит на страницу, где отражается его баланс или имеется кнопка “Оплатить за счет баланса”;

2) в систему oracle направляется запрос на получение информации о текущем балансе пользователя:

наименование хранимой процедуры ХХХ008

входные параметры

{

“oracle\_user\_id”: integer

}

выходные параметры:

при успешном ответе

{

“error”: null,

“result”: {

“balance”: float

}

}

при неуспешном ответе

{

“error”: string, ***- текст сообщения об ошибке***

“result”: null

}

**8. Запрос текущего состояния счета абонемента**

1) пользователь переходит на страницу, где отражается состояние счета его абонемента или имеется кнопка “Оплатить за счет абонемента”;

2) в систему oracle направляется запрос на получение информации о текущем состоянии счета абонемента пользователя:

наименование хранимой процедуры ХХХ009

входные параметры

{

“oracle\_subscription\_id”: integer

}

выходные параметры:

при успешном ответе

{

“error”: null,

“result”: {

“current\_subscription\_sum”: float

}

}

при неуспешном ответе

{

“error”: string, ***- текст сообщения об ошибке***

“result”: null

}

**9. Запрос доступных персональных скидок**

1) пользователь переходит на страницу, где отражается доступные скидки или отражается конечная стоимость услуги;

2) в систему oracle направляется запрос на получение информации о доступных для пользователя персональных скидках:

наименование хранимой процедуры ХХХ010

входные параметры

{

“oracle\_subscription\_id”: integer

}

выходные параметры:

при успешном ответе

{

“error”: null,

“result”: [

{

“discount”: float,

“description”: string

},

{

…

}

]

}

при неуспешном ответе

{

“error”: string, ***- текст сообщения об ошибке***

“result”: null

}