

ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА МОСКВЫ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ГОРОДА МОСКВЫ
ШКОЛА № 7

Проектная работа
По разработке программного обеспечения
для терминалов
«АльТер (Альтернативный Терапевт)»

Выполнили:

Ученики 11 «Б» класса

ГБОУ Школы № 7

Журавлев Иван Юрьевич

Левкин Артем Игоревич

Руководитель проекта:

Молотков Никита

Александрович

г. Москва
2023 г.

Оглавление

Цель	3
Актуальность.....	3
Методика выполнения:	
Постановка задач.....	3
Изучение литературы для реализации проекта	4
Выполнение задач	4
Этапы разработки.....	5
Используемые программы, языки программирования, сервисы.....	18
Результат.....	19
Перспективы	19
Ссылки	19
Список литературы	20

Аннотация

На сегодняшний день для решения проблемы здравоохранения населения создано множество инструментов: мобильные приложения для дистанционной записи на прием к врачу, средство диагностики по симптомам на основе уже существующих диагнозов и множество других. Однако они никаким образом не связаны и не ссылаются друг на друга, несмотря на то, что это позволило бы существенно облегчить задачу работникам сферы здравоохранения. Проект объединяет существующие сервисы и собственный функционал в единую систему, которая позволяет пациенту быстро и удобно получать оценку своего здоровья, создавать и управлять записями к врачу, просматривать информацию медкарты и расшифровывать анализы.

Цель

Целью проекта является разработка удобного и эффективного программного обеспечения для терминалов, которое в перспективе упростит процесс медицинского обслуживания, увеличит пропускную способность медучреждений, освободит врачей от рутинной работы, а также снизит процент постановки неправильных диагнозов.

Актуальность

Актуальность проекта обусловлена отсутствием аналогичных систем. В данный момент доступны лишь отдельные, никак не связанные сервисы диагностики, которые, к тому же не работают на терминалах в поликлиниках и больницах.

Постановка задач

При работе над проектом были поставлены следующие задачи:

- Анализ существующих сервисов в области здравоохранения.
- Выбор языка программирования, фреймворка, библиотек, а также изучение литературы по ним.
- Формулировка представления о конечном виде продукта.
- Разработка программного обеспечения.
- Дизайн и полировка конечного продукта.

Изучение литературы для реализации проекта

- Для библиотек, требующих знание парсинга и структуры сайта Selenium и Requests были изучены официальные документации, примеры использования.
- При разработке был использован фреймворк Kivy навыки работы с которым уже имелись, однако, также в проекте было решено использовать дополнение KivyMD для изучения которого была использована официальная документация и видеоуроки разных авторов.
- Для обработки полученной информации используется библиотека Pandas изучение которой проводилось по официальной документации.

Выполнение задач

1. Анализ существующих сервисов в области здравоохранения.

Для реализации АльТер необходимы лабораторные анализы и медкарта пользователя, а также возможность записи к врачу, из-за чего Единая Московская Информационно-Аналитическая Система (ЕМИАС) является одной из самых важных составляющих конечного программного обеспечения. Для определения диагноза по симптомам было решено использовать отечественный сервис Helzy, который вдобавок умеет расшифровывать анализы.

2. Формулировка представления о конечном виде продукта.

Конечный продукт должен объединять в себе Единую Московскую Информационную Аналитическую Систему (ЕМИАС), сервис Helzy, собственный функционал для измерения и анализа физических показателей, а также понятный и удобный дизайн.

3. Выбор языка программирования, фреймворка, библиотек.

При разработке было решено использовать известный и освоенный язык программирования Python, фреймворк Kivy, навыки работы с которым уже имелись, его дополнение KivyMD, Pandas, Selenium, Requests и другие вспомогательные библиотеки.

4. Разработка программного обеспечения.

Разработка АльТер велась последовательно, с разделением задач. Код по мере разработки был структурирован и оптимизирован (Рис.3). Интерфейс проекта улучшен для комфортного использования. В ходе тестирования были устранены обнаруженные ошибки и недочеты.

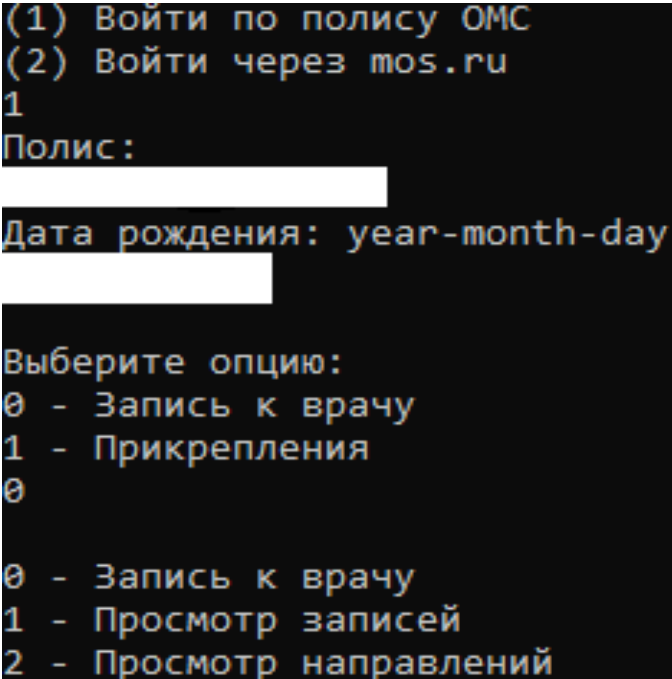
5. Дизайн.

Дизайн несколько раз менялся и пересматривался. В итоге было решено перейти от ярких и раздражающих цветов к темно-зеленым тонам. Внешний вид не вызывает диссонанса вследствие контраста с внешним уровнем освещения. Интерфейс интуитивно понятен и прост. Работа с дизайном проводилась в графическом онлайн-редакторе Figma.

Этапы разработки

I. Начало разработки.

1. Исследование сайта emias.info, изучение способов авторизации.
Создание терминальной версии парсера для ЕМИАС (Рис. 1).



```
(1) Войти по полису ОМС
(2) Войти через mos.ru
1
Полис:
Дата рождения: year-month-day
Выберите опцию:
0 - Запись к врачу
1 - Прикрепления
0
0 - Запись к врачу
1 - Просмотр записей
2 - Просмотр направлений
```

Рис. 1 - Пример работы терминала

2. Исследование сайта lk.emias.mos.ru для доступа к медицинской карте.
Создание способов входа с использованием библиотеки Selenium и Requests.
3. Начало разработки графического интерфейса с использованием фреймворка Kivy и KivyMD, а также дизайна графического онлайн-редактора Figma (Рис. 2, Рис. 4, Рис. 5).

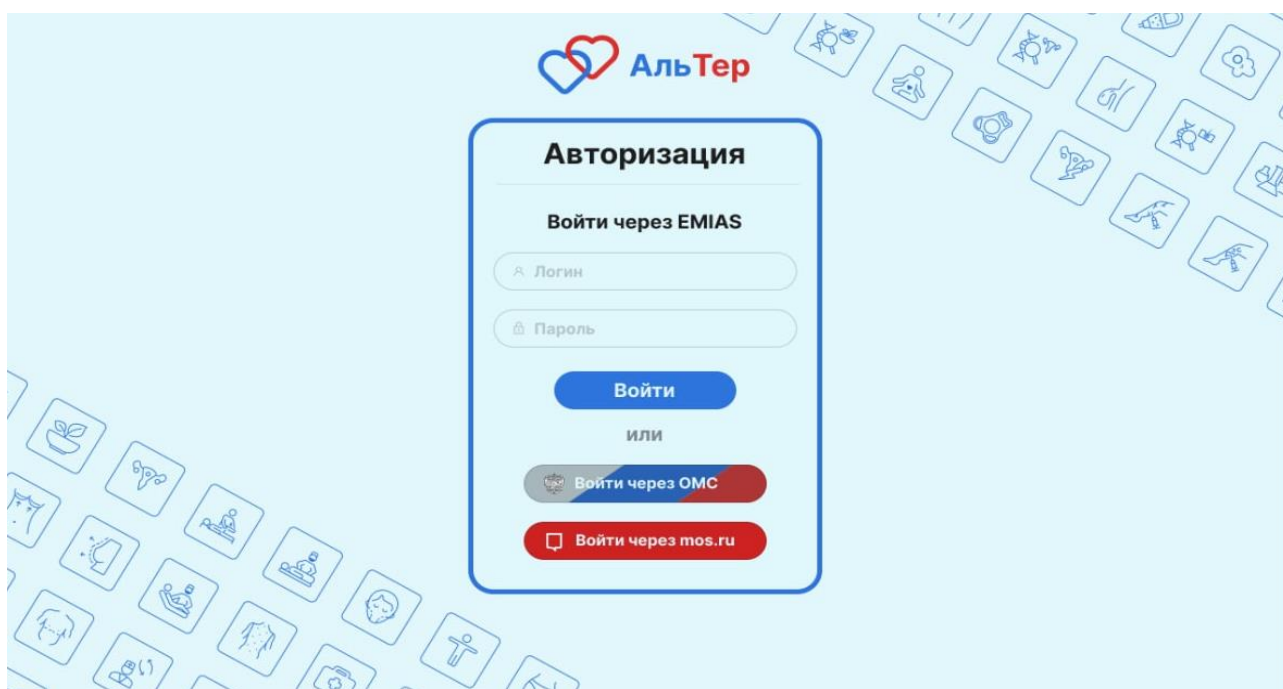


Рис. 2 - Форма авторизации

4. Реализация интерфейса систем входа через mos.ru и полис ОМС (Рис.6).

II. Внедрение ЕМИАС.

1. Начало внедрения ЕМИАС, создание графического интерфейса главного экрана и записи к врачу (Рис. 7, Рис. 8).
2. Добавление возможности просмотра направлений.
3. Добавление возможности просмотра прикреплений.
4. Добавление возможности просмотра, отмены и переноса записи.

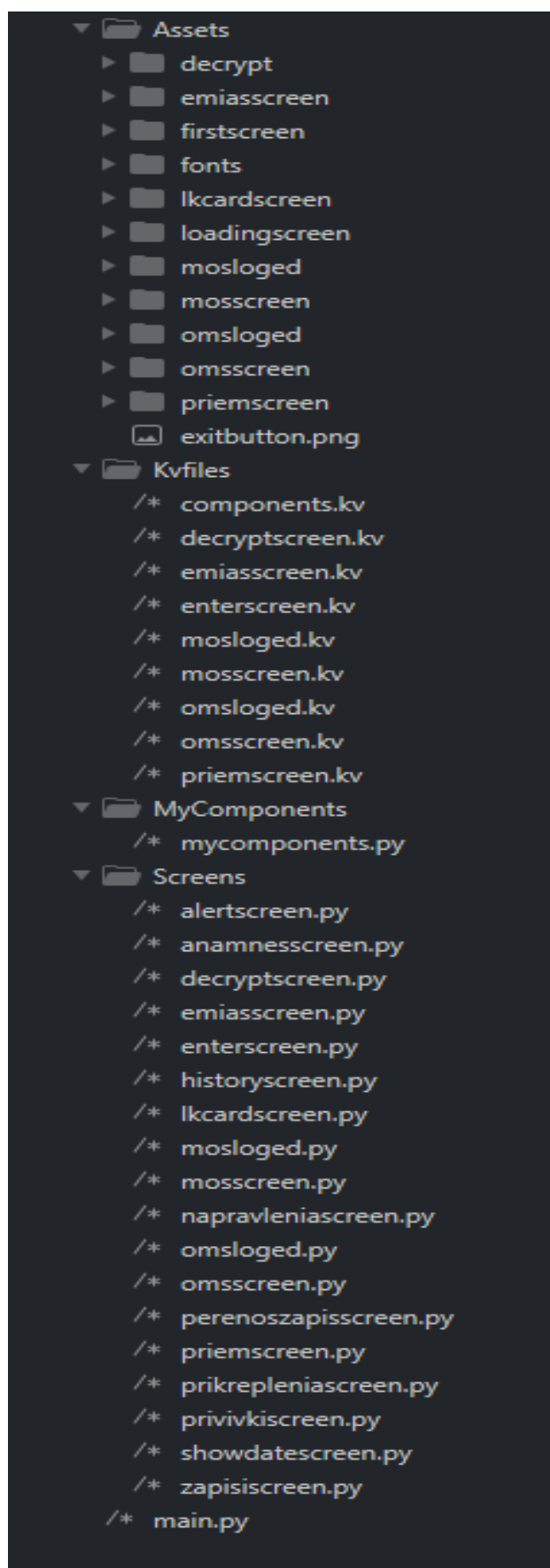


Рис. 3 - Структура проекта.

```

<OMSLogged>:
  name: 'logged'
  Image:
    source: 'Assets/omsloded/bg/mainmenubgoms.png'
    allow_stretch: True
    keep_ratio: False
  MDTextButton:
    size_hint: .2, .15
    halign: 'center'
    pos_hint: {'center_x': .89, 'center_y': .08}
    font_size: dp(25)
    on_release: root.exits()
    Image:
      source: 'Assets/moslogged/exitbut.png'
      center_x: self.parent.center_x
      center_y: self.parent.center_y
      allow_stretch: True
      size: self.parent.size
  MDLabel:
    id: authname
    text: 'Если вы не видите здесь свою фамилию вам очень повезло'
    text_color: get_color_from_hex('#72C3AC')
    theme_text_color: 'Custom'
    halign: 'center'
    size_hint: .27, .06
    pos_hint: {'center_x': .515, 'center_y': .09}
    font_size: dp(40)

  MDTextButton:
    orientation: "vertical"
    pos_hint: {'center_x': .358, 'center_y': .575}
    size_hint: .106, .069
    on_release:
      root.manager.current = "load"
      root.manager.get_screen('logged').zapisi()
    Image:
      source: 'Assets/moslogged/myzapis.png'
      center_x: self.parent.center_x
      keep_ratio: False
      allow_stretch: True
      center_y: self.parent.center_y
      size: self.parent.size
  MDTextButton:
    orientation: "vertical"
    pos_hint: {'center_x': .488, 'center_y': .575}
    size_hint: .1226, .0694
    on_release:
      root.manager.current = "load"
      root.manager.get_screen('logged').prosmotrnapr()
    Image:
      source: 'Assets/moslogged/mynapr.png'
      center_x: self.parent.center_x
      center_y: self.parent.center_y
      keep_ratio: False
      allow_stretch: True
      size: self.parent.size
  MDTextButton:
    orientation: "vertical"
    pos_hint: {'center_x': .63, 'center_y': .575}
    size_hint: .13125, .0694
    on_release:
      root.manager.current = "load"
      root.manager.get_screen('logged').prikreplenia()
    Image:
      source: 'Assets/moslogged/myprik.png'
      center_x: self.parent.center_x
      center_y: self.parent.center_y
      size: self.parent.size
      allow_stretch: True
      keep_ratio: False
  MDTextButton:
    orientation: "vertical"
    pos_hint: {'center_x': .5, 'center_y': .7059}
    size_hint: .3906, .1159
    on_release:
      root.manager.current = "load"
      root.manager.get_screen('logged').newzapis()
    Image:
      source: 'Assets/moslogged/emiasbutton.png'
      center_x: self.parent.center_x
      center_y: self.parent.center_y
      size: self.parent.size
      allow_stretch: True
      keep_ratio: False
  MDTextButton:
    orientation: "vertical"

```

Рис. 4 - Часть кода разметки UI написанная с помощью KVLing.


```

window.size = (1920, 1080)

for kv in listdir('kvfiles'):
    Builder.load_file(f"kvfiles/{kv}")

class Loading(Screen):
    pass

class AFK(Screen):
    pass

class MyToggleButton(MDFlatButton, MDToggleButton):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.background_down = get_color_from_hex('#51857A')
        self.font_color_normal = get_color_from_hex('#D4F5EC')
        self.font_color_down = get_color_from_hex('#72C3AC')

class AlterApp(MDApp):
    sm = None

    def build(self):
        self.sm = ScreenManager()
        sm = self.sm
        sm.add_widget(AFK(name='afk'))
        sm.add_widget(SuccPerenos(name='perenossucc'))
        sm.add_widget(SuccOtmena(name='otmenasucc'))
        sm.add_widget(Succ(name='succnew'))
        sm.add_widget(OMSErrorScreen(name='omserror'))
        sm.add_widget(OMSErrorUnkscreen(name='omserrorunk'))
        sm.add_widget(OMSAlerScreen(name='omsalert'))
        sm.add_widget(EmiassScreen(name='emiasscreen'))
        sm.add_widget(Alert(name='alert'))
        sm.add_widget(HelpScreen(name='help'))
        sm.add_widget(ENTERScreen(name='enter'))
        sm.add_widget(OMSScreen(name='oms'))
        sm.add_widget(MOSScreen(name='mos'))
        sm.add_widget(OMSLogged(name='logged'))
        sm.add_widget(MOSLogged(name='moslogged'))
        sm.add_widget>Loading(name='load'))
        sm.add_widget>Zapisi(name='zapisi'))
        sm.add_widget>Perenos(name='perenos'))
        sm.add_widget>Showdate(name='timetable'))
        sm.add_widget>Prikreplenia(name='prik'))
        sm.add_widget>Napravlenia(name='napr'))
        sm.add_widget>LKCard(name='lkcard'))
        sm.add_widget>History(name='history'))
        sm.add_widget>AnamnesView(name='anamn'))
        sm.add_widget>Privivki(name='priv'))
        sm.add_widget>(PrivivkiView(name='privview'))
        sm.add_widget>Priem(name='priem'))
        sm.add_widget>Decrypt(name='decrypt'))
        Clock.schedule_interval(sm.get_screen('enter').update, 2)
        for i in range(1, 32):
            twl = MyToggleButton(halign="center", text=f'{i}', text_color=get_color_from_hex('#D4F5EC'),
                                theme_text_color='Custom', font_style='H4', on_press=sm.get_screen('oms').days,
                                md_bg_color=get_color_from_hex('#32494B'), size_hint=(1, .2), group="x")
            twl.date = i
            sm.get_screen('oms').ids.container1.add_widget(twl)
        months = [
            'Январь',
            'Февраль',
            'Март',
            'Апрель',
            'Май',
            'Июнь',
            'Июль',
            'Август',
            'Сентябрь',
            'Октябрь',
            'Ноябрь',
            'Декабрь'
        ]
        for i in months:
            twl = MyToggleButton(halign="center", text=f'{i}', text_color=get_color_from_hex('#D4F5EC'),
                                theme_text_color='Custom', font_style='H4', on_press=sm.get_screen('oms').months,
                                md_bg_color=get_color_from_hex('#32494B'), size_hint=(1, .2), group="y")
            twl.month = i
            sm.get_screen('oms').ids.container2.add_widget(twl)
        for i in reversed(range(1900, datetime.datetime.now().year + 1)):
            twl = MyToggleButton(halign="center", text=f'{i}', text_color=get_color_from_hex('#D4F5EC'),
                                theme_text_color='Custom', font_style='H4', on_press=sm.get_screen('oms').years,
                                md_bg_color=get_color_from_hex('#32494B'), size_hint=(1, .2), group="z")
            twl.year = i
            sm.get_screen('oms').ids.container3.add_widget(twl)
        return sm

if __name__ == '__main__':
    AlterApp().run()

```

Рис. 5 - Часть кода запуска программы.

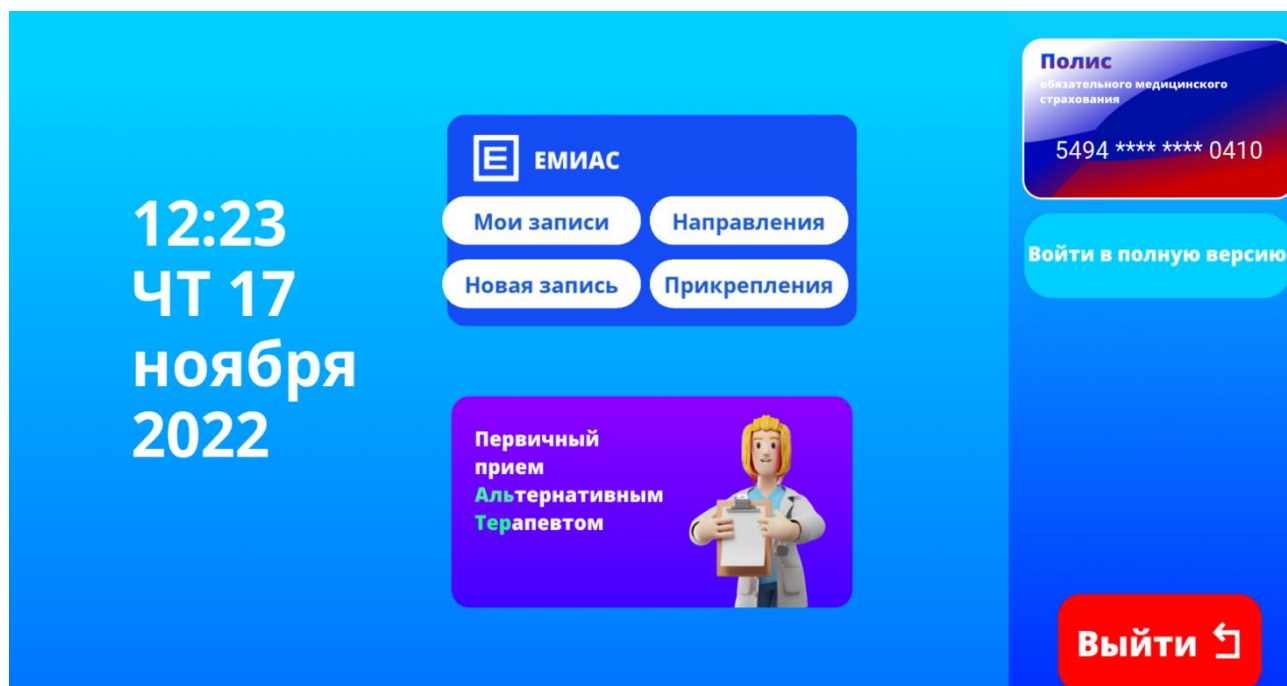


Рис. 7 - Главное меню.

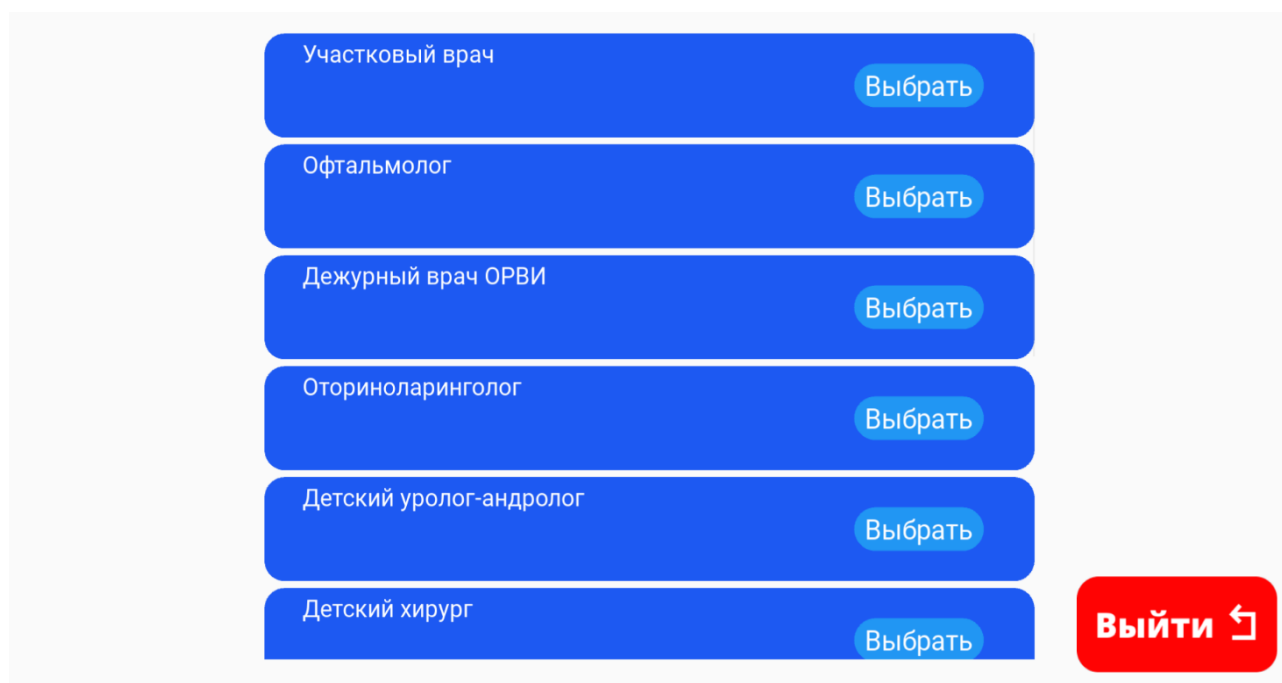


Рис. 8 - Панель записи на прием.

III. Внедрение медкарты.

1. Создание экрана выбора для просмотра необходимого документа.
2. Создание парсера документов.
3. Форматирование полученного html кода в изображение для отображения пользователю. (Рис. 9)

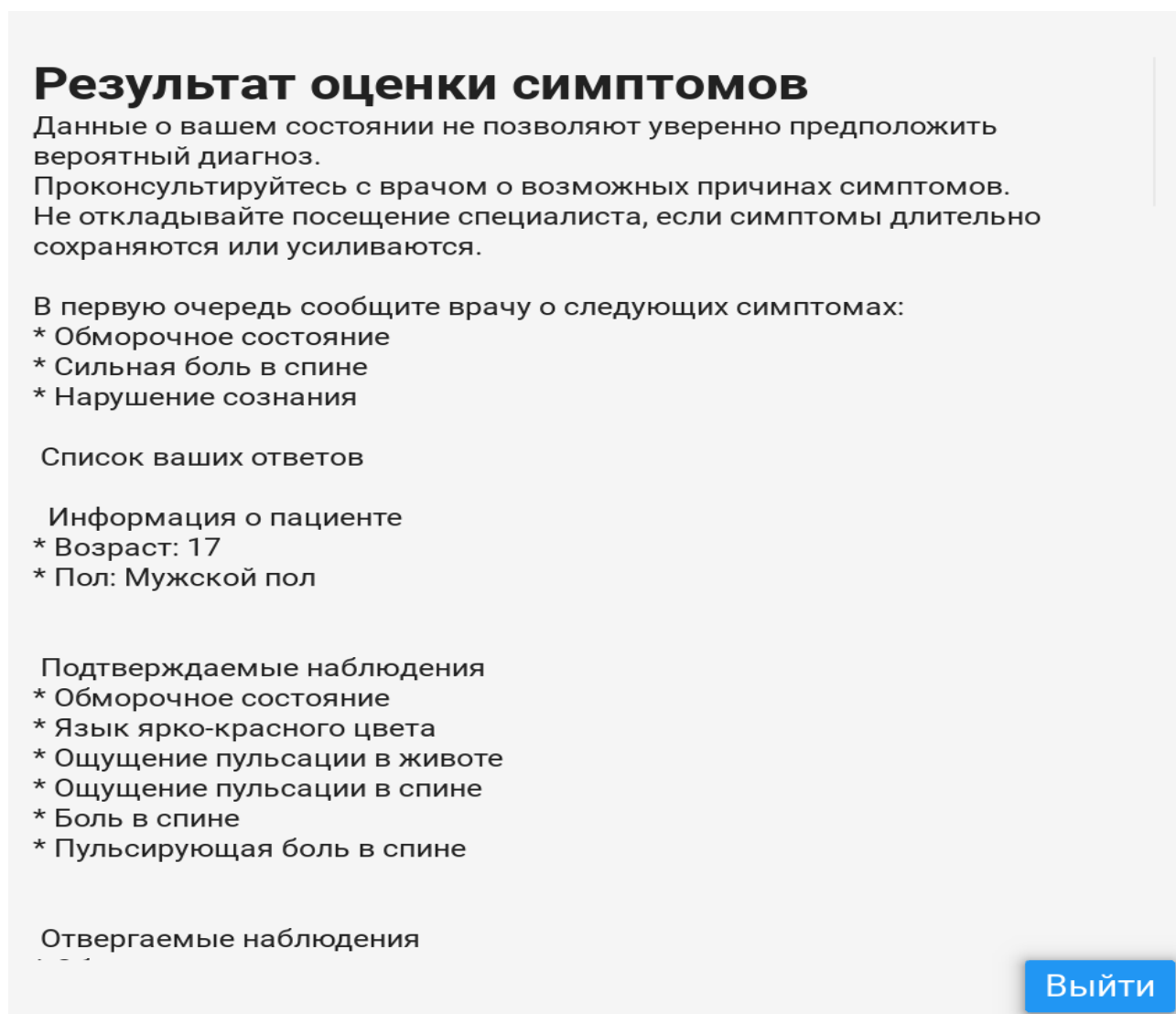
ГБУЗ «ДКЦ № 1 ДЗМ» КЛИНИКО-ДИАГНОСТИЧЕСКАЯ ЛАБОРАТОРИЯ					
РЕЗУЛЬТАТЫ ЛАБОРАТОРНЫХ ИССЛЕДОВАНИЙ					
ОАК + СОЭ + микроскопия мазка при выявлении патологии (капиллярная кровь), дети					
Биоматериал: кровь цельная. Локус: капилляр.					
Тест	Результат	Норма	Отклонение	Критичность отклонения	Ед. изм.
Количество лейкоцитов	5,72	4 - 9,23	НОРМА	Норма (не числовой результат)	10 ⁹ /л
Скорость оседания эритроцитов (по Вестергрену)	4	1 - 15	НОРМА	Норма (не числовой результат)	мм/ч
Гемоглобин общий	151	130 - 160	НОРМА	Норма (не числовой результат)	г/л
Гематокрит	44,3	35 - 47	НОРМА	Норма (не числовой результат)	%
Средний объем эритроцита	85	76,7 - 89,2	НОРМА	Норма (не числовой результат)	fL
Среднее содержание гемоглобина в эритроците	29	25,2 - 30,2	НОРМА	Норма (не числовой результат)	пг
Средняя концентрация гемоглобина в эритроците	341	320 - 360	НОРМА	Норма (не числовой результат)	г/л
Ширина распределения эритроцитов по объему	12	11,5 - 14,5	НОРМА	Норма (не числовой результат)	%
Количество тромбоцитов	219	160 - 380	НОРМА	Норма (не числовой результат)	10 ⁹ /л
Средний объем тромбоцитов в крови	11,1	7,8 - 11,8	НОРМА	Норма (не числовой результат)	fL
Общий объем тромбоцитов в крови (тромбокрит, РСТ)	0,24	0,15 - 0,45	НОРМА	Норма (не числовой результат)	%
Относительное количество нейтрофилов	47	44 - 72	НОРМА	Норма (не числовой результат)	%
Абсолютное количество нейтрофилов	2,69	1,82 - 6,6	НОРМА	Норма (не числовой результат)	10 ⁹ /л
Относительное количество лимфоцитов	39,7	19 - 45	НОРМА	Норма (не числовой результат)	%

Выйти

Рис. 9 - Пример результатов анализов.

IV. Внедрение Helzy (Первичный прием).

1. Парсинг Helzy с помощью Selenium и Requests.
2. Обработка и вывод результата полученного по окончании тестирования Helzy. (Рис. 10)



Результат оценки симптомов

Данные о вашем состоянии не позволяют уверенно предположить вероятный диагноз.
Проконсультируйтесь с врачом о возможных причинах симптомов.
Не откладывайте посещение специалиста, если симптомы длительно сохраняются или усиливаются.

В первую очередь сообщите врачу о следующих симптомах:

- * Обморочное состояние
- * Сильная боль в спине
- * Нарушение сознания

Список ваших ответов

Информация о пациенте

- * Возраст: 17
- * Пол: Мужской пол

Подтверждаемые наблюдения

- * Обморочное состояние
- * Язык ярко-красного цвета
- * Ощущение пульсации в животе
- * Ощущение пульсации в спине
- * Боль в спине
- * Пульсирующая боль в спине

Отвергаемые наблюдения

Выйти

Рис. 10 - Пример результата исследования.

3. Удобное отображение тестирования Helzy в виде web приложения (Рис.11).

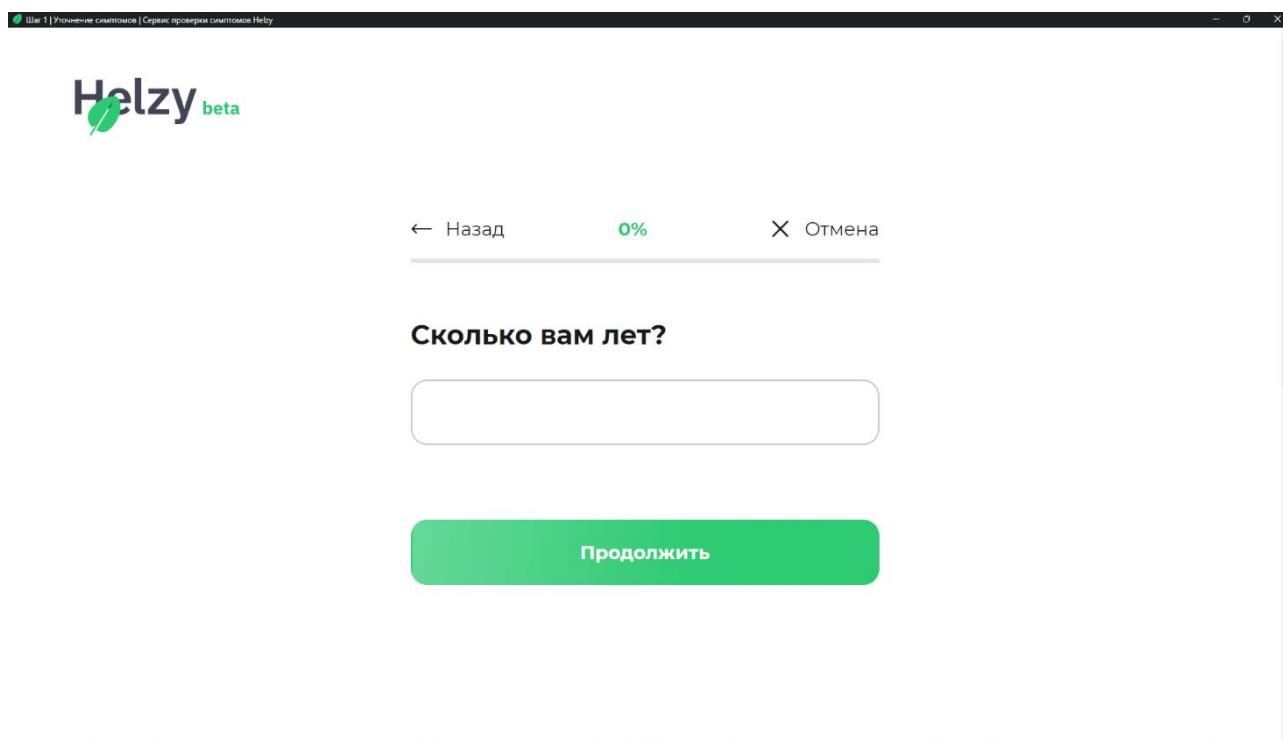


Рис. 11 - Пример формы сервиса Helzy.

4. Создание системы расшифровки анализов.

5. Вывод результатов расшифровки (Рис. 12, Рис. 13).

Выявление лабораторных признаков анемии и уточнение состояния эритропоэза в целом

У вас не выявлено признаков нарушения эритропоэза (процесса выработки эритроцитов - клеток, обеспечивающих органы и ткани необходимым количеством кислорода), в том числе, не выявлено наличия лабораторных признаков анемии. Выявленное у вас изменение средней концентрации гемоглобина в эритроците без наличия признаков анемии и изменения количества эритроцитов является клинически незначимым.

Состояние тромбоцитарного гемостаза

У вас не выявлено признаков нарушения тромбоцитарного гемостаза, уровень тромбоцитов в норме.

Уровень лейкоцитов крови

Лейкоциты – клетки крови, участвующие в защите организма от инфекций и чужеродных агентов. Изменение их количества в крови может происходить при инфекционных и других заболеваниях. У вас не обнаружено признаков нарушения процесса выработки лейкоцитов, количество лейкоцитов в крови находится в пределах нормы.

Рис. 12 - Пример результатов расшифровки анализов.

```

def myanaliz(self):
    try:
        analiz = self.s.get(
            f'https://lk.emias.mos.ru/api/1/documents/analyzes?ehrId={self.idus}&shortDateFilter=all_time',
            headers={'X-Access-JWT': self.authtoken})
        jsanaliz = analiz.json()
        for i in range(len(jsanaliz['documents'])):
            card = MDCard(size_hint=(1, None), height=330, md_bg_color=(0, 0, 0, 0))
            layout = RelativeLayout()
            layout.add_widget(Image(source='Assets/omsloged/zapisperenos.png', keep_ratio=False))
            title = MDLabel(
                text=f'{jsanaliz["documents"][i]["title"]}',
                theme_text_color='Custom',
                text_color=get_color_from_hex('#D4F5EC'),
            )
            if len(jsanaliz['documents'][i]['title']) < 88:
                title.font_size = 40
            else:
                title.font_size = 30
            title.font_name = 'Assets/fonts/roboto.ttf'
            title.pos_hint = {'center_x': .55, 'center_y': .7}
            layout.add_widget(title)
            time = datetime.datetime.fromisoformat(jsanaliz['documents'][i]['date'])
            timelab = MDLabel(
                text=f'{time.strftime("%a, %d %b %Y")}',
                theme_text_color='Custom',
                text_color=get_color_from_hex('#D4F5EC'),
            )
            timelab.font_size = 40
            timelab.font_name = 'Assets/fonts/roboto.ttf'
            timelab.pos_hint = {'center_x': .55, 'center_y': .45}
            if 'ОАК' in jsanaliz['documents'][i]['title'] or 'Общий клинический анализ крови' in \
                jsanaliz['documents'][i]['title'] or (
                    'крос' in jsanaliz['documents'][i]['title'] and 'общ' in jsanaliz['documents'][i]['title']) or 'Клинический анализ крови' in jsanaliz['documents'][i]['title']:
                dec = MDFlatButton(
                    text='Расшифровать',
                    theme_text_color='Custom',
                    text_color=get_color_from_hex('#D4F5EC'),
                    size_hint=(.4846, .17),
                    md_bg_color=(0, 0, 0, 0))
                dec.docid = jsanaliz['documents'][i]['documentId']
                dec.date = jsanaliz['documents'][i]['date']
                dec.bind(on_release=self.OKAKLK, )
                dec.pos_hint = {'center_x': .748, 'center_y': .23}
                dec.font_size = 30
                dec.font_name = 'Assets/fonts/roboto.ttf'
                look = MDFlatButton(
                    text='Просмотр',
                    theme_text_color='Custom',
                    text_color=get_color_from_hex('#D4F5EC'),
                    size_hint=(.49, .17),
                    md_bg_color=(0, 0, 0, 0))
                look.pos_hint = {'center_x': .253, 'center_y': .23}
                look.font_size = 30
                look.font_name = 'Assets/fonts/roboto.ttf'
                look.docid = jsanaliz['documents'][i]['documentId']
                look.bind(on_release=self.manager.get_screen('lkcard').documentview)
                layout.add_widget(look)
                layout.add_widget(dec)
                layout.add_widget(timelab)
                card.add_widget(layout)
                self.ids.scrollid.add_widget(card)
            elif 'ОАМ' in jsanaliz['documents'][i]['title'] or 'Общий клинический анализ мочи' in \
                jsanaliz['documents'][i]['title'] or 'Клинический анализ мочи' in jsanaliz['documents'][i]['title'] or (
                    'моч' in jsanaliz['documents'][i]['title'] and 'анализ' in jsanaliz['documents'][i]['title']):
                dec = MDFlatButton(
                    text='Расшифровать',
                    theme_text_color='Custom',
                    text_color=get_color_from_hex('#D4F5EC'),
                    size_hint=(.4846, .17),
                    md_bg_color=(0, 0, 0, 0))
                dec.docid = jsanaliz['documents'][i]['documentId']
                dec.date = jsanaliz['documents'][i]['date']
                dec.bind(on_release=self.OKAKLK, )
                dec.pos_hint = {'center_x': .748, 'center_y': .23}
                dec.font_size = 30
                dec.font_name = 'Assets/fonts/roboto.ttf'
                look = MDFlatButton(
                    text='Просмотр',
                    theme_text_color='Custom',
                    text_color=get_color_from_hex('#D4F5EC'),
                    size_hint=(.49, .17),
                    md_bg_color=(0, 0, 0, 0))
                look.pos_hint = {'center_x': .253, 'center_y': .23}
                look.font_size = 30
                look.font_name = 'Assets/fonts/roboto.ttf'
                look.docid = jsanaliz['documents'][i]['documentId']
                look.bind(on_release=self.manager.get_screen('lkcard').documentview)
                layout.add_widget(look)
                layout.add_widget(dec)
                layout.add_widget(timelab)
                card.add_widget(layout)
                self.ids.scrollid.add_widget(card)
        self.manager.current = 'decrypt'

```

Рис. 13 - Часть кода расшифровки анализов.

6. Объединение системы измерения и анализа физических показателей с Helzy.

V. Полировка и дизайн.

1. Полная переработка дизайна (Рис. 14 - Рис. 17).

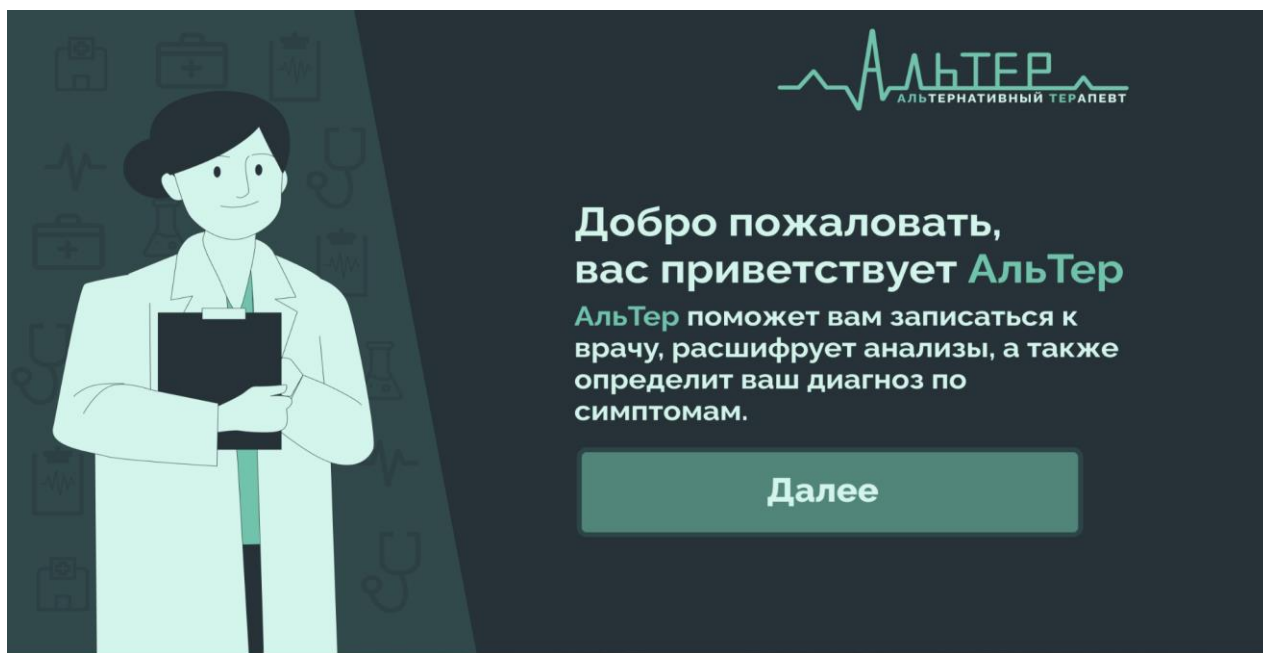


Рис. 14 - Экран приветствия

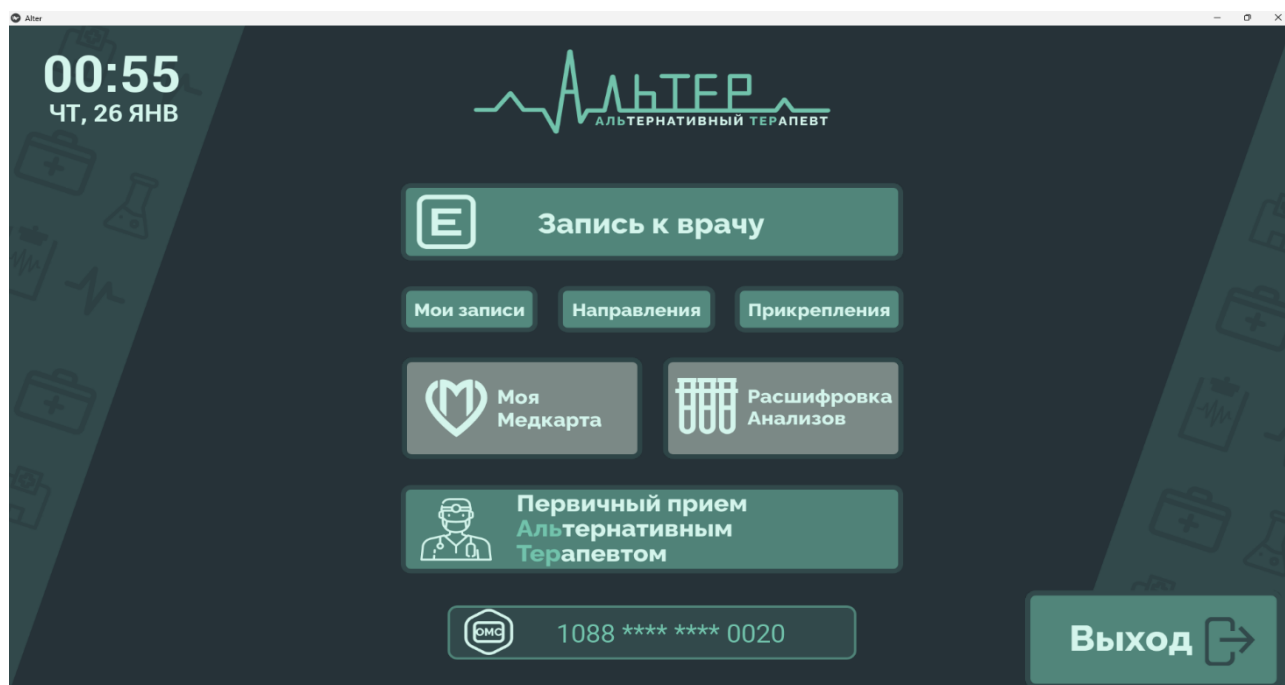


Рис. 15 - Главное меню

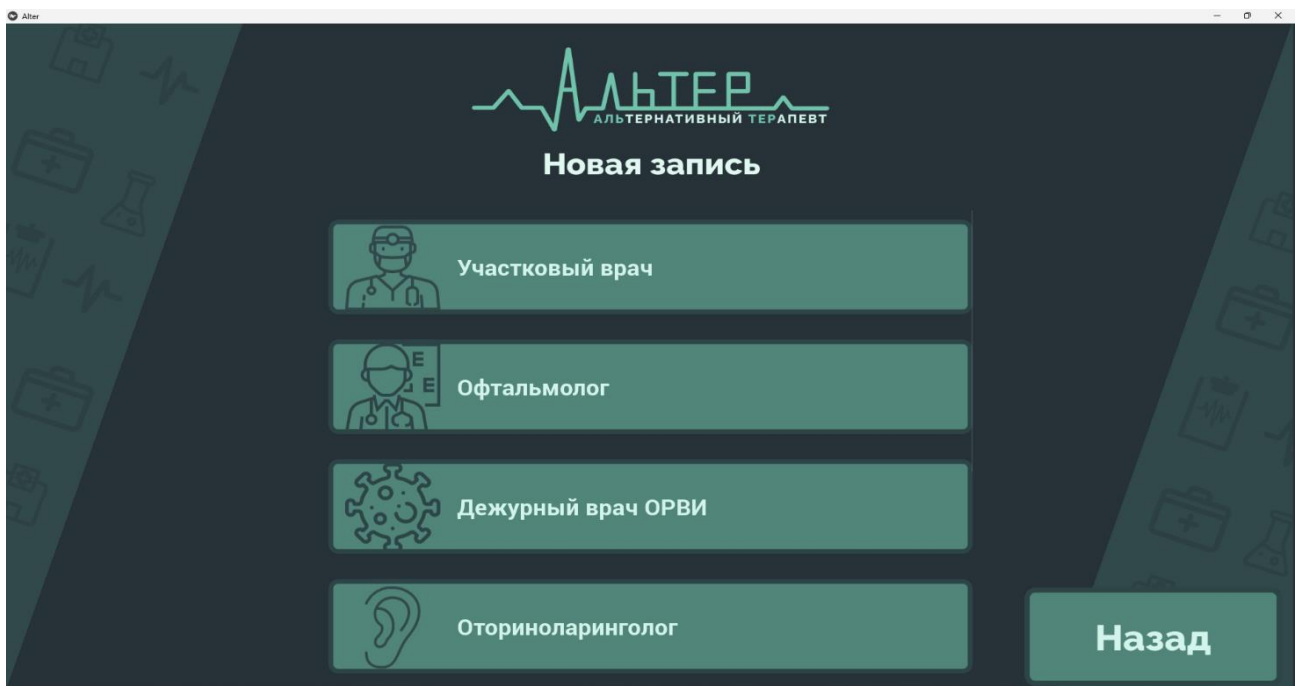


Рис. 16 - Меню записи на прием

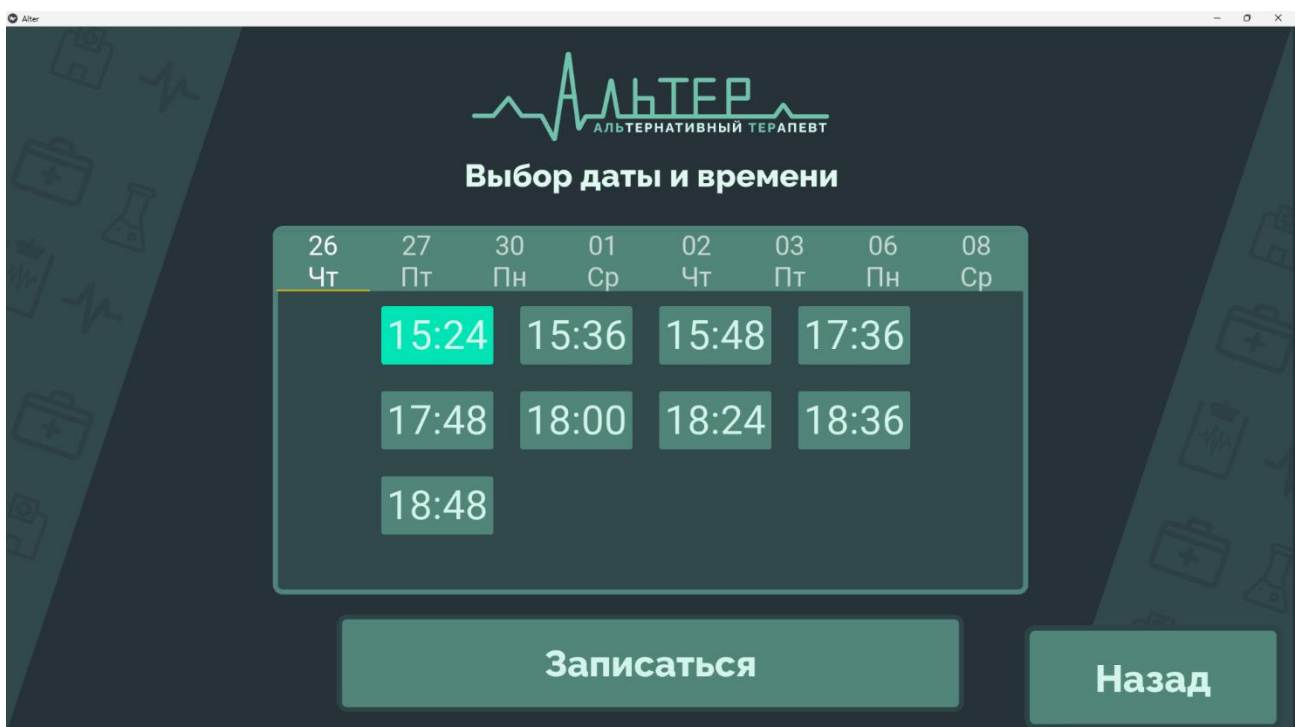


Рис. 17 - Меню выбора даты и времени

2. Исправление отображения виджетов.
3. Добавление исключений (Пропущенные значения, символы, аргументы).
4. Оптимизация кода (Избавление от глобальных переменных, работа в потоке).
5. Исправление критических багов (Отказ работы программы из-за устаревшего токена авторизации).

Используемые программы, языки программирования, сервисы

Используемые программы:

Sublime Text (Редактор кода).

Figma (Сервис для разработки интерфейсов и прототипирования).

Firefox (Браузер, исследование сайта и запросов ЕМИАС, а также конвертация html в pdf).

Chrome (Авторизация в программе через mos.ru).

Языки программирования:

Python.

KVLang.

Библиотеки:

Kivy.

KivyMD.

Requests.

Selenium.

Sele-Tools.

CairoSVG.

Pandas.

Сервисы:

ЕМИАС.

ЛК.ЕМИАС.

Helzy.

Результат

Результатом является программное обеспечение, которое объединяет различные медицинские сервисы в единую систему и делает их использование удобным и эффективным, позволяя пациенту получать всю необходимую информацию о его здоровье, расшифровывать анализы, записываться к врачу, исходя из симптомов, жалоб и физических показателей получать вероятный диагноз, а также направление к врачу специалисту для его подтверждения.

Таким образом, АльТер позволяет миновать консультацию врача терапевта, экономит время, а также значительно упрощает диагностику и сбор анамнеза.

Перспективы

В будущем возможно добавление новых сервисов диагностики (SBER MED AI), а также расширение способов измерения физ. показателей (Анализ фотографий кожи, горла и т.д.)

Ссылки

Репозиторий проекта GitHub - <https://github.com/QuasyStellar/Alter>

Дизайн проекта -

<https://www.figma.com/file/0HvtdPQZix1DgVWHQ4BBVv/screen?node-id=0%3A1&t=s11tRpVLqh0DIkgb-1>

Презентация -

<https://www.figma.com/file/Rv98GK3NpQKUzNSIKzBsRN/present?node-id=0%3A1&t=8mFbj4xtbMiaFMXU-1>

Список литературы

Kivy (бесплатный GUI фреймворк для Python) - <https://kivy.org/>

KivyMD (дополнение к фреймворку Kivy) -
<https://kivymd.readthedocs.io/en/1.1.1/>

Selenium (кросс-платформенный инструмент для управления веб браузерами) -
<https://www.selenium.dev/>

Requests (модуль для языка Python, который используют для упрощения работы с HTTP-запросами) -
<https://requests.readthedocs.io/en/latest/>

Pandas (программная библиотека на языке Python для обработки и анализа данных) -
<https://pandas.pydata.org/>

PuPi (каталог программного обеспечения, написанного на языке программирования Python) -
<https://pypi.org/>

Habr (Русскоязычный веб-сайт в формате системы тематических коллективных блогов) -
<https://habr.com/ru/all/>

StackOverflow (система вопросов и ответов о программировании) -
<https://stackoverflow.com/>

Figma (Графический редактор) - <https://figma.com/>

ЕМИАС (Единая Московская Информационно-Аналитическая Система) -
<https://emias.info/>

Медкарта ЕМИАС – <https://lk.emias.mos.ru/>

MOS.RU (Сайт Мэра Москвы) – <https://mos.ru/>