

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук  
Департамент программной инженерии

СОГЛАСОВАНО

УТВЕРЖДАЮ

Доцент департамента программной  
инженерии факультета компьютерных  
наук

Академический руководитель  
образовательной программы  
«Программная инженерия» профессор  
департамента программной инженерии,  
канд. техн. наук

\_\_\_\_\_ Х. М. Салех  
«\_\_\_\_\_» \_\_\_\_\_ 2020 г.

\_\_\_\_\_ В. В. Шилов  
«\_\_\_\_\_» \_\_\_\_\_ 2020 г.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ СООБЩЕНИЙ О  
БЫТОВЫХ ПРОБЛЕМАХ СТУДЕНТОВ В ОБЩЕЖИТИИ

Клиент  
Текст программы

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.02.07-01 12 01-1-ЛУ

Исполнитель: Студент группы БПИ-194  
\_\_\_\_\_ В. А. Анненков  
«\_\_\_\_\_» \_\_\_\_\_ 2020 г.

УТВЕРЖДЁН  
RU.17701729.02.07-01 12 01-1-ЛУ

## МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ СООБЩЕНИЙ О БЫТОВЫХ ПРОБЛЕМАХ СТУДЕНТОВ В ОБЩЕЖИТИИ

Клиент  
Текст программы

RU.17701729.02.07-01 12 01-1

Листов 50

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

# Содержание

<b>1</b>	<b>Текст программы</b>	<b>4</b>
1.1	Views	4
1.1.1	AboutPage.xaml	4
1.1.2	AboutPage.xaml.cs	5
1.1.3	ChatPage.xaml	6
1.1.4	AboutPage.xaml.cs	7
1.1.5	ItemDetailPage.xaml	10
1.1.6	ItemDetailPage.xaml.cs	12
1.1.7	ItemsPage.xaml	14
1.1.8	ItemsPage.xaml.cs	15
1.1.9	LoginPage.xaml	17
1.1.10	LoginPage.xaml.cs	17
1.1.11	MainInfoPage.xaml	19
1.1.12	MainInfoPage.xaml.cs	22
1.1.13	MainPage.xaml	26
1.1.14	MainPage.xaml.cs	27
1.1.15	NewItemPage.xaml	27
1.1.16	NewItemPage.xaml.cs	28
1.1.17	NoticePage.xaml	29
1.1.18	NoticePage.xaml.cs	29
1.1.19	ProfilePage.xaml	30
1.1.20	ProfilePage.xaml.cs	31
1.1.21	WaitPage.xaml	32
1.1.22	WaitPage.xaml.cs	33
1.2	ViewModels	35
1.2.1	AboutViewModel.cs	35
1.2.2	BaseViewModel.cs	35
1.2.3	ChatViewModel.cs	36
1.2.4	ItemDetailViewModel.cs	37
1.2.5	ItemsViewModel.cs	37
1.2.6	MainInfoViewModel.cs	38
1.2.7	ProfileViewModel.cs	40
1.2.8	WaitViewModel.cs	41
1.3	Services	41
1.3.1	ApiService.cs	41
1.3.2	Extensions.cs	42
1.3.3	IHseSupporterApi.cs	42
1.4	Models	44
1.4.1	AuthResult.cs	44
1.4.2	Dormitory.cs	44
1.4.3	Event.cs	44
1.4.4	IModel.cs	45
1.4.5	MainPage.cs	45
1.4.6	MainQuestion.cs	46
1.4.7	Message.cs	46
1.4.8	Notice.cs	47

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1.4.9	Problem.cs . . . . .	48
1.4.10	Profile.cs . . . . .	48

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

# 1. Текст программы

## 1.1. Views

### 1.1.1. AboutPage.xaml

```
<?xml version="1.0" encoding="utf-8"?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms" xmlns:x="http://schemas.
microsoft.com/winfx/2009/xaml"
    xmlns:d="http://xamarin.com/schemas/2014/forms/design"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" mc:
        Ignorable="d"
    x:Class="HSESupporter.Views.AboutPage" xmlns:vm="clr-namespace:
        HSESupporter.ViewModels"
    Title="{Binding Title}">
    <ContentPage.BindingContext>
        <vm:AboutViewModel />
    </ContentPage.BindingContext>
    <ContentPage.Resources>
        <ResourceDictionary>
            <Color x:Key="Primary">#2196F3</Color>
            <Color x:Key="Accent">#96d1ff</Color>
            <Color x:Key="LightTextColor">#999999</Color>
        </ResourceDictionary>
    </ContentPage.Resources>
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>
        <StackLayout BackgroundColor="{StaticResource Accent}" VerticalOptions="
            FillAndExpand" HorizontalOptions="Fill">
            <StackLayout Orientation="Horizontal" HorizontalOptions="Center" VerticalOptions
                ="Center">
                <ContentView Padding="0,10,0,10" VerticalOptions="FillAndExpand">
                    <Image Source="hse_logo.png" VerticalOptions="Center" HeightRequest
                        ="200" />
                </ContentView>
            </StackLayout>
        </StackLayout>
        <ScrollView Grid.Row="1">
            <StackLayout Orientation="Vertical" Padding="16,40,16,40" Spacing="10">
                <Label FontSize="22">
                    <Label.FormattedText>
                        <FormattedString>
                            <FormattedString.Spans>
                                <Span Text="HSE Supporter" FontAttributes="Bold" FontSize
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        ="22" />
        <Span Text=" " />
        <Span Text="0.1" ForegroundColor="{StaticResource
            LightTextColor}" />
        </FormattedString.Spans>
    </FormattedString>
</Label.FormattedText>
</Label>
<Label>
    <Label.FormattedText>
        <FormattedString>
            <FormattedString.Spans>
                <Span TextЭто="
                    приложениепоможетвамбыстросвязатьсясадминистрациейобщее
                    " />
                <Span Text=" " />
                <Span TextВысшей=" ШколыЭкономики" FontAttributes="
                    Bold" />
                <Span Text="." />
            </FormattedString.Spans>
        </FormattedString>
    </Label.FormattedText>
</Label>
<Label>
    <Label.FormattedText>
        <FormattedString>
            <FormattedString.Spans>
                <Span TextПриложение=" работаетнаплатформах" />
                <Span Text=" " />
                <Span Text="iOS иAndroid" FontAttributes="Bold" />
                <Span Text="." />
            </FormattedString.Spans>
        </FormattedString>
    </Label.FormattedText>
</Label>
<Button Margin="0,10,0,0" TextУзнать=" больше" Command="{Binding
    OpenWebCommand}"
        BackgroundColor="{StaticResource Primary}" TextColor="White" />
</StackLayout>
</ScrollView>
</Grid>
</ContentPage>

```

### 1.1.2. AboutPage.xaml.cs

```

using System;
using System.ComponentModel;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
namespace HSESupporter.Views
{
    // Learn more about making custom code visible in the Xamarin.Forms previewer
    // by visiting https://aka.ms/xamarinforms-previewer
    [DesignTimeVisible(false)]
    public partial class AboutPage : ContentPage
    {
        public AboutPage()
        {
            InitializeComponent();
        }
    }
}
```

### 1.1.3. ChatPage.xaml

```
<?xml version="1.0" encoding="utf-8"?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:viewModels="clr-namespace:HSESupporter.ViewModels;assembly=
        HSESupporter"
    xmlns:elements="clr-namespace:HSESupporter.Views.Elements;assembly=
        HSESupporter"
    Title="{Binding Title}"
    x:Class="HSESupporter.Views.ChatPage">
    <ContentPage.BindingContext>
        <viewModels:ChatViewModel />
    </ContentPage.BindingContext>
    <ContentPage.Content>
        <ScrollView
            x:Name="ScrollView">
            <StackLayout x:Name="StackLayout">
                <StackLayout VerticalOptions="Fill"
                    x:Name="Messages"
                    Padding="12,8,12,0">
                    <!--<elements:Message Margin="42,0,-8,0" />
                    <elements:Message Margin="-8,0,42,0" />
                    <elements:Message Margin="42,0,-8,0" />-->
                </StackLayout>
            <Grid
                Margin="8, 12, 8, 8"
                HorizontalOptions="FillAndExpand"
                Padding="1">
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="30" />
                    <ColumnDefinition Width="*" />
                    <ColumnDefinition Width="30" />
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
</Grid.ColumnDefinitions>
```

```
<Image x:Name="RefreshButton"
      Grid.Column="0"
      HeightRequest="30"
      WidthRequest="30"
      Source="refresh.png"
      Margin="0,0,0,0">
  <Image.GestureRecognizers>
    <TapGestureRecognizer
      Tapped="Refresh_OnTapped"
      NumberOfTapsRequired="1" />
  </Image.GestureRecognizers>
</Image>
```

```
<ActivityIndicator Grid.Column="0" x:Name="LoadingIndicatorRefresh"
      HeightRequest="30"
      WidthRequest="30"
      IsRunning="False"
      IsVisible="False" />
```

```
<Frame Grid.Column="1" BorderColor="DarkGray"
      HasShadow="False"
      HorizontalOptions="FillAndExpand"
      CornerRadius="5">
  <Editor
    x:Name="MessageEditor"
    AutoSize="TextChanges"
    PlaceholderВведите=" сообщение..." />
</Frame>
```

```
<Image Grid.Column="2" x:Name="SendButton"
      HeightRequest="30"
      WidthRequest="30"
      HorizontalOptions="End"
      Source="send.png"
      Margin="0,0,0,0">
  <Image.GestureRecognizers>
    <TapGestureRecognizer
      Tapped="SendButton_OnClicked"
      NumberOfTapsRequired="1" />
  </Image.GestureRecognizers>
</Image>
<ActivityIndicator Grid.Column="2" x:Name="LoadingIndicatorSend"
      HeightRequest="30"
      WidthRequest="30"
      IsRunning="False"
      IsVisible="False" />
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```

        </Grid>
    </StackLayout>
</ScrollView>
</ContentPage.Content>
</ContentPage>

```

#### 1.1.4. AboutPage.xaml.cs

```

using System;
using System.Linq;
using System.Net.Http;
using System.Threading.Tasks;
using HSESupporter.Services;
using HSESupporter.ViewModels;
using HSESupporter.Views.Elements;
using Xamarin.Essentials;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace HSESupporter.Views
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class ChatPage : ContentPage
    {
        private readonly ChatViewModel _viewModel;

        public ChatPage(ChatViewModel viewModel)
        {
            InitializeComponent();

            InitChat(true);

            BindingContext = _viewModel = viewModel;
        }

        public ChatPage()
        {
            InitializeComponent();

            InitChat(true);

            _viewModel = new ChatViewModel();
            BindingContext = _viewModel;
        }

        private async Task InitChat(bool isScrollDown)
        {
            try
            {

```

ИЗМ.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

var api = new ApiService().HseSupporterApi;
var result = await api.GetDormitory(Preferences.Get("dormitory_id", 1));
var messages = result.Messages;

Messages.Children.Clear();

foreach (var message in messages)
    Messages.Children.Add(new Message(Preferences.Get("id", 0)
        .Equals(message.Author))
    {
        PAuthor = {Text = $"{message.AuthorFirstName} {message.
            AuthorLastName}"},
        PText = {Text = message.Text}, PTime = {Text = message.CreatedAt.
            GetDateTimeText()}
    });

if (isScrollDown)
{
    await Task.Delay(50);
    await ScrollView.ScrollToAsync(0,
        StackLayout.Children.LastOrDefault().Y,
        false );
}
}
catch (HttpRequestException e)
{
    await DisplayAlertОшибка("", "Возникла" ошибкаприподключениииксайту "",
        "ОК");
}
catch (Exception e)
{
    await DisplayAlertОшибка("", "Возникла" ошибка", "ОК");
}
}

private async void SendButton_OnClicked(object sender, EventArgs e)
{
    SendButton.IsVisible = false;
    LoadingIndicatorSend.IsVisible = true;
    LoadingIndicatorSend.IsRunning = true;

    try
    {
        if (MessageEditor.Text.Equals("")) return;

        var text = MessageEditor.Text;
        MessageEditor.Text = "";
    }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

var firstName = Preferences.Get("first_name", "");
var lastName = Preferences.Get("last_name", "");

var message = new Models.Message
{
    IsRead = false,
    IsFromStudent = true,
    Dormitory = Preferences.Get("dormitory_id", 1),
    Text = text
};

var api = new ApiService().HseSupporterApi;
await api.SendMessage(message.GetDictionaryParams());

Messages.Children.Add(new Message(true)
{
    PAuthor = {Text = $"{firstName} {lastName}"}, PText = {Text = text},
    PTime = {Text = DateTime.Now.ToString("HH:mm")}
});

await Task.Delay(1);
await ScrollView.ScrollToAsync(StackLayout.Children.LastOrDefault(),
    ScrollToPosition.MakeVisible,
    false);
}
catch (Exception ex)
{
    // Nothing
}

LoadingIndicatorSend.IsVisible = false;
LoadingIndicatorSend.IsRunning = false;
SendButton.IsVisible = true;
}

private async void Refresh_OnTapped(object sender, EventArgs e)
{
    RefreshButton.IsVisible = false;
    LoadingIndicatorRefresh.IsVisible = true;
    LoadingIndicatorRefresh.IsRunning = true;

    await InitChat(true);

    await Task.Delay(1);
    await ScrollView.ScrollToAsync(StackLayout.Children.LastOrDefault(),
        ScrollToPosition.MakeVisible,
        false);
}

```

ИЗМ.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        LoadingIndicatorRefresh.IsVisible = false;
        LoadingIndicatorRefresh.IsRunning = false;
        RefreshButton.IsVisible = true;
    }
}
}

```

#### 1.1.5. ItemDetailPage.xaml

```

<?xml version="1.0" encoding="utf-8"?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms" xmlns:x="http://schemas.
microsoft.com/winfx/2009/xaml"
    xmlns:d="http://xamarin.com/schemas/2014/forms/design"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    x:Class="HSESupporter.Views.ItemDetailPage" Title="{Binding Title}">
    <RefreshView x:Name="RefreshView" IsRefreshing="{Binding IsBusy}"
        Command="{Binding RefreshCommand}">
        <ScrollView>
            <StackLayout Spacing="20" Padding="10">
                <StackLayout x:Name="LoadingStatus"
                    Orientation="Vertical"
                    VerticalOptions="CenterAndExpand">
                    <ActivityIndicator
                        IsRunning="True"
                        VerticalOptions="Center"
                        Scale="1.5" />
                    <Label TextИдёт=" загрузка..."
                        VerticalOptions="Center"
                        HorizontalOptions="CenterAndExpand" />
                </StackLayout>

                <StackLayout x:Name="Content" IsVisible="False">
                    <Label x:Name="Title" d:Text="Item name" FontSize="20" />
                    <Label x:Name="Description" Padding="4"
                        BackgroundColor="AliceBlue"
                        d:Text="Item description" FontSize="16" />
                    <StackLayout Orientation="Horizontal">
                        <Label TextСтатус=":" FontSize="14" />
                        <Label Textв=" обработке" Padding="1" BackgroundColor="
                            LightGoldenrodYellow" FontSize="14" />
                    </StackLayout>

                    <StackLayout VerticalOptions="Fill"
                        x:Name="Messages"
                        Padding="8,8,8,0">
                        <!--<local:Message Margin="42,0,-8,0" />
                        <local:Message Margin="-8,0,42,0" />

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

<local:Message Margin="42,0,-8,0" />-->
</StackLayout>

<Grid
    Margin="8, 12, 8, 8"
    HorizontalOptions="FillAndExpand"
    Padding="1">
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="*" />
        <ColumnDefinition Width="30" />
    </Grid.ColumnDefinitions>

    <Frame Grid.Column="0" BorderColor="DarkGray"
        HasShadow="False"
        HorizontalOptions="FillAndExpand"
        CornerRadius="5">
        <Editor
            x:Name="MessageEditor"
            AutoSize="TextChanges"
            PlaceholderВведите=" сообщение..." />
        </Frame>

        <Image Grid.Column="1" x:Name="SendButton"
            HeightRequest="30"
            WidthRequest="30"
            HorizontalOptions="End"
            Source="send.png"
            Margin="0,0,0,0">
            <Image.GestureRecognizers>
                <TapGestureRecognizer
                    Tapped="SendButton_OnClicked"
                    NumberOfTapsRequired="1" />
            </Image.GestureRecognizers>
        </Image>
        <ActivityIndicator Grid.Column="1" x:Name="LoadingIndicatorSend"
            HeightRequest="30"
            WidthRequest="30"
            IsRunning="False"
            IsVisible="False" />
    </Grid>
</StackLayout>
</StackLayout>
</ScrollView>
</RefreshView>
</ContentPage>

```

#### 1.1.6. ItemDetailPage.xaml.cs

using System;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

using System.ComponentModel;
using HSESupporter.Models;
using HSESupporter.Services;
using HSESupporter.ViewModels;
using Xamarin.Essentials;
using Xamarin.Forms;
using Message = HSESupporter.Views.Elements.Message;

namespace HSESupporter.Views
{
    // Learn more about making custom code visible in the Xamarin.Forms previewer
    // by visiting https://aka.ms/xamarinforms-previewer
    [DesignTimeVisible(false)]
    public partial class ItemDetailPage : ContentPage
    {
        private readonly ItemDetailViewModel _viewModel;

        public ItemDetailPage(ItemDetailViewModel viewModel)
        {
            InitializeComponent();

            BindingContext = _viewModel = viewModel;
            _viewModel.Load += InitItem;

            _viewModel.OnLoad();
        }

        public ItemDetailPage()
        {
            InitializeComponent();

            var item = new Problem
            {
                Title = "Item 1",
                Description = "This is an item description."
            };

            _viewModel = new ItemDetailViewModel(item);
            BindingContext = _viewModel;
        }

        private async void InitItem(object sender, EventArgs e)
        {
            _viewModel.IsBusy = true;
            try
            {
                var api = new ApiService().HseSupporterApi;
                var result = await api.GetProblem(_viewModel.Item.Id);
            }
            catch { }
        }
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

_viewModel.Title = result.Title;
Title.Text = result.Title;
Description.Text = result.Description;

Messages.Children.Clear();
foreach (var message in result.Messages)
    Messages.Children.Add(new Message(Preferences.Get("id", 0)
        .Equals(message.Author))
    {
        PAuthor = {Text = $"{message.AuthorFirstName} {message.
            AuthorLastName}"},
        PText = {Text = message.Text}, PTime = {Text = message.CreatedAt.
            GetDateTimeText()}
    });

if (LoadingStatus.IsVisible)
    LoadingStatus.IsVisible = false;
if (!Content.IsVisible)
    Content.IsVisible = true;
}
catch (Exception ex)
{
}
finally
{
    _viewModel.IsBusy = false;
}
}

private async void SendButton_OnClicked(object sender, EventArgs e)
{
    try
    {
        if (MessageEditor.Text.Equals("")) return;

        var text = MessageEditor.Text;
        MessageEditor.Text = "";

        var firstName = Preferences.Get("first_name", "");
        var lastName = Preferences.Get("last_name", "");
        Messages.Children.Add(new Message(true)
        {
            PAuthor = {Text = $"{firstName} {lastName}"}, PText = {Text = text},
            PTime = {Text = DateTime.Now.ToString("HH:mm")}
        });

        var message = new Models.Message

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

{
    Author = 1,
    IsRead = false,
    IsFromStudent = true,
    Text = text,
    Problem = _viewModel.Item.Id
};

var api = new ApiService().HseSupporterApi;
await api.SendMessage(message.GetDictionaryParams());
}
catch (Exception ex)
{
    // Nothing
}
}
}
}

```

#### 1.1.7. ItemsPage.xaml

```

<?xml version="1.0" encoding="utf-8"?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms" xmlns:x="http://schemas.
microsoft.com/winfx/2009/xaml"
    xmlns:d="http://xamarin.com/schemas/2014/forms/design"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" mc:
Ignorable="d"
    x:Class="HSESupporter.Views.ItemsPage" Title="{Binding Title}" x:Name="
BrowseItemsPage">
    <ContentPage.ToolbarItems>
        <ToolBarItem Text=Добавить="" Clicked="AddItem_Clicked" />
    </ContentPage.ToolbarItems>
    <StackLayout>
        <ListView x:Name="ItemsListView" ItemsSource="{Binding Items}" VerticalOptions="
FillAndExpand"
            HasUnevenRows="true" RefreshCommand="{Binding LoadItemsCommand}"
            IsPullToRefreshEnabled="true"
            IsRefreshing="{Binding IsBusy, Mode=OneWay}" CachingStrategy="
RecycleElement"
            ItemSelected="OnItemSelected">
            <d:ListView.ItemsSource>
                <x:Array Type="{x:Type x:String}">
                    <x:StringПервая> проблема</x:String>
                    <x:StringВторая> проблема</x:String>
                    <x:StringТретья> проблема</x:String>
                    <x:StringЧетвёртая> проблема</x:String>
                    <x:StringПятая> проблема</x:String>
                </x:Array>
            </d:ListView.ItemsSource>
        </ListView>
    </StackLayout>
</ContentPage>

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```

</d:ListView.ItemsSource>
<ListView.ItemTemplate>
    <DataTemplate>
        <ViewCell>
            <StackLayout Orientation="Horizontal" Padding="10"
                HorizontalOptions="FillAndExpand">
                <StackLayout>
                    <Label Text="{Binding Title}" d:Text="{Binding .}"
                        LineBreakMode="NoWrap"
                        Style="{DynamicResource ListItemTextStyle}" FontSize
                            ="16" />
                    <Label Text="{Binding Description}" d:TextОписание="
                        проблемы" LineBreakMode="NoWrap"
                        Style="{DynamicResource ListItemDetailTextStyle}"
                        FontSize="13" />
                </StackLayout>

                <Label Text="{Binding CreatedAtBeauty}" d:Text="{Binding .}"
                    LineBreakMode="NoWrap"
                    HorizontalOptions="EndAndExpand"
                    TextColor="Gray"
                    Style="{DynamicResource ListItemTextStyle}"
                    FontSize="13" />
            </StackLayout>
            <ViewCell.ContextActions>
                <MenuItem Clicked="MenuItem_OnDelete"
                    CommandParameter="{Binding .}"
                    TextУдалить="" IsDestructive="True" />
            </ViewCell.ContextActions>
        </ViewCell>
    </DataTemplate>
</ListView.ItemTemplate>
</ListView>
</StackLayout>
</ContentPage>

```

#### 1.1.8. ItemsPage.xaml.cs

```

using System;
using System.ComponentModel;
using HSESupporter.Models;
using HSESupporter.Services;
using HSESupporter.ViewModels;
using Xamarin.Forms;

```

```

namespace HSESupporter.Views
{
    [DesignTimeVisible(false)]
    public partial class ItemsPage : ContentPage

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

{
    private readonly ItemsViewModel _viewModel;

    public ItemsPage()
    {
        InitializeComponent();

        BindingContext = _viewModel = new ItemsViewModel();
    }

    private async void OnItemSelected(object sender, SelectedItemChangedEventArgs args)
    {
        var item = args.SelectedItem as Problem;
        if (item == null)
            return;

        await Navigation.PushAsync(new ItemDetailPage(new ItemDetailViewModel(item)));

        // Manually deselect item.
        ItemsListView.SelectedItem = null;
    }

    private async void AddItem_Clicked(object sender, EventArgs e)
    {
        await Navigation.PushAsync(new NewItemPage());
    }

    protected override void OnAppearing()
    {
        base.OnAppearing();

        _viewModel.LoadItemsCommand.Execute(null);
    }

    private async void MenuItem_OnDelete(object sender, EventArgs e)
    {
        try
        {
            var problem = ((MenuItem) sender).BindingContext as Problem;
            var api = new ApiService().HseSupporterApi;

            if (problem == null) return;
            await api.DeleteProblem(problem.Id);

            ItemsListView.BeginRefresh();
        }
        catch (Exception ex)
        {

```

ИЗМ.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    // Empty
  }
}
}
}

```

### 1.1.9. LoginPage.xaml

```

<?xml version="1.0" encoding="utf-8"?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="HSESupporter.Views.LoginPage">
    <ContentPage.Content>
        <StackLayout VerticalOptions="Center">
            <Label HorizontalTextAlignment="Center"
                Text="HSE Supporter"
                FontSize="30"
                FontAttributes="Bold"
                TextColor="#5B69FF" />
            <Label HorizontalTextAlignment="Center"
                TextВведите=" вашEmail имыотправимнанегокодподтверждения ."
                FontSize="16"
                FontAttributes="Bold"
                TextColor="DarkGray" />

            <Entry x:Name="Login" Placeholder="Email" Margin="12,0" />
            <Entry x:Name="Code" PlaceholderКод=" подтверждения" Margin="12,0"
                IsVisible="False" />

            <Button TextВойти="" FontSize="22" Clicked="Button_OnClicked" />
            <ActivityIndicator x:Name="LoadingIndicator" IsRunning="False" />

            <Label x:Name="ResultText" HorizontalTextAlignment="Center"
                TextУспешно=""
                FontSize="20"
                IsVisible="False"
                FontAttributes="Bold"
                TextColor="#5B69FF" />
        </StackLayout>
    </ContentPage.Content>
</ContentPage>

```

### 1.1.10. LoginPage.xaml.cs

```

using System;
using System.Collections.Generic;
using HSESupporter.Services;
using Xamarin.Essentials;
using Xamarin.Forms;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
using Xamarin.Forms.Xaml;

namespace HSESupporter.Views
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class LoginPage : ContentPage
    {
        private bool _isConfirm;

        public LoginPage()
        {
            InitializeComponent();
        }

        private async void Button_OnClicked(object sender, EventArgs e)
        {
            try
            {
                if (!_isConfirm)
                {
                    var api = new ApiService().HseSupporterApi;

                    var values = new Dictionary<string, object>
                    {
                        {"email", Login.Text},
                    };

                    LoadingIndicator.IsRunning = true;
                    var result = await api.Login(values);
                    LoadingIndicator.IsRunning = false;

                    ResultText.IsVisible = true;
                    ResultText.Text = "Проверочный" кодотправлен !";

                    Code.IsVisible = true;

                    _isConfirm = true;
                }
                else
                {
                    var api = new ApiService().HseSupporterApi;

                    var values = new Dictionary<string, object>
                    {
                        {"email", Login.Text},
                        {"code", Code.Text}
                    };
                }
            }
        }
    }
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

LoadingIndicator.IsRunning = true;
var result = await api.LoginConfirm(values);
LoadingIndicator.IsRunning = false;

ResultText.IsVisible = true;
ResultText.Text = "Успешный вход!";

Code.IsVisible = true;

Preferences.Set("is_accept", result.IsAccept);
Preferences.Set("id", result.Profile.Id);
Preferences.Set("first_name", result.Profile.FirstName);
Preferences.Set("last_name", result.Profile.LastName);
Preferences.Set("info", result.Profile.Info);
Preferences.Set("is_student", result.Profile.Role.Equals("student"));
Preferences.Set("token", result.Token);

if (result.IsAccept)
    await Navigation.PushModalAsync(new MainPage());
else
    await Navigation.PushModalAsync(new WaitPage());
}
}
catch (Exception ex)
{
    LoadingIndicator.IsRunning = false;

    ResultText.IsVisible = true;
    ResultText.Text = "Произошла ошибка.";
}
}
}
}

```

#### 1.1.11. MainInfoPage.xaml

```

<?xml version="1.0" encoding="utf-8"?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:local="clr-namespace:HSESupporter.Views.Elements"
    xmlns:viewmodels="clr-namespace:HSESupporter.ViewModels"
    Title="{Binding Title}"
    x:Class="HSESupporter.Views.MainInfoPage">
    <ContentPage.BindingContext>
        <viewmodels:MainInfoViewModel />
    </ContentPage.BindingContext>
    <ContentPage.ToolbarItems>
        <ToolbarItem Text="Профиль" Clicked="MenuItem_OnClicked" />
    </ContentPage.ToolbarItems>

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

</ContentPage.ToolbarItems>
<ContentPage.Content>
  <RefreshView x:Name="RefreshView"
    IsRefreshing="{Binding IsBusy}"
    Command="{Binding RefreshCommand}">
    <ScrollView>
      <StackLayout>
        <StackLayout x:Name="LoadingStatus"
          Orientation="Vertical"
          VerticalOptions="CenterAndExpand">
          <ActivityIndicator
            x:Name="ActivityIndicator"
            IsRunning="True"
            VerticalOptions="Center"
            Scale="1.5" />
          <Label x:Name="ResultText" TextИдёт=" загрузка..."
            VerticalOptions="Center"
            HorizontalOptions="CenterAndExpand" />
        </StackLayout>
        <StackLayout x:Name="Content" IsVisible="False">
          <StackLayout VerticalOptions="Fill"
            x:Name="StackImportantMessages"
            Padding="8,8,8,0">
            <!--<local:ImportantNotice />-->
          </StackLayout>

          <StackLayout HorizontalOptions="CenterAndExpand">
            <Label
              x:Name="Name"
              HorizontalTextAlignment="Center"
              Margin="10,0,10,0"
              FontSize="22"
              Text="" />
            <Label
              x:Name="DormitoryName"
              HorizontalTextAlignment="Center"
              Margin="10,0,10,0"
              FontSize="16"
              Text="" />
            <Label
              x:Name="DormitoryAddress"
              HorizontalTextAlignment="Center"
              Margin="10,0,10,0"
              FontSize="12"
              Text="" />
          </StackLayout>

          <Label

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

x:Name="DormitoryUsersCount"
Margin="10,0,10,0"
FontSize="16"
IsVisible="False"
Text="" />
<Label
  x:Name="AgentsOnlineCount"
  Margin="10,0,10,0"
  FontSize="16"
  IsVisible="False"
  Text="" />

<CarouselView x:Name="CarouselView"
  PeekAreaInsets="26" VerticalOptions="Center"
  HeightRequest="230">
  <CarouselView.GestureRecognizers>
    <TapGestureRecognizer Tapped="
      TapGestureRecognizer_OnTapped" />
  </CarouselView.GestureRecognizers>
  <CarouselView.ItemTemplate>
    <DataTemplate>
      <StackLayout>
        <Frame HasShadow="False"
          BorderColor="DarkGray"
          CornerRadius="5"
          Margin="15, 5"
          HorizontalOptions="FillAndExpand"
          VerticalOptions="Center">
          <StackLayout>
            <Label Text="{Binding Title}"
              FontAttributes="Bold"
              FontSize="Large"
              HorizontalOptions="Center"
              VerticalOptions="Center" />
            <Label Text="{Binding Description}"
              FontAttributes="Italic"
              HorizontalOptions="Center"
              MaxLines="5"
              LineBreakMode="TailTruncation" />
            <Label Text="{Binding CreatedAtBeauty}"
              HorizontalOptions="EndAndExpand"
              FontSize="13"
              TextColor="Gray"
              MaxLines="5"
              LineBreakMode="TailTruncation" />
          </StackLayout>
        </Frame>
      </StackLayout>
    </DataTemplate>
  </CarouselView.ItemTemplate>
</CarouselView>

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        </DataTemplate>
    </CarouselView.ItemTemplate>
</CarouselView>

    <StackLayout Orientation="Horizontal"
        x:Name="EventsTitle"
        IsVisible="False"
        Margin="10,0,10,0">
        <Image HeightRequest="22"
            WidthRequest="22"
            Source="calendar-month.png"
            Margin="0,0,0,0" />
        <Label TextСобытия=":"
            FontSize="18"
            FontAttributes="Bold" />
    </StackLayout>
    <StackLayout x:Name="Events" />

    <StackLayout Orientation="Horizontal"
        x:Name="MainQuestionsTitle"
        IsVisible="False"
        Margin="10,12,10,0">
        <Image HeightRequest="22"
            WidthRequest="22"
            Source="help-circle-outline.png"
            Margin="0,0,0,0" />
        <Label TextГлавные=" вопросы:"
            FontSize="18"
            FontAttributes="Bold" />
    </StackLayout>
    <StackLayout x:Name="MainQuestions"
        Margin="10,0,10,0" />
    </StackLayout>
</StackLayout>
</ScrollView>
</RefreshView>
</ContentPage.Content>
</ContentPage>

```

#### 1.1.12. MainInfoPage.xaml.cs

```

using System;
using System.Linq;
using HSESupporter.Models;
using HSESupporter.ViewModels;
using HSESupporter.Views.Elements;
using Xamarin.Essentials;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```

namespace HSESupporter.Views
{
    /// <summary>
    /// Классглавнойстраницыприложения .
    /// </summary>
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class MainInfoPage : ContentPage
    {
        public MainInfoPage()
        {
            InitializeComponent();
            if (!(BindingContext is MainInfoViewModel vm)) return;

            vm.Load += InitProfile;
            vm.Load += InitMainNotices;
            vm.Load += InitEvents;
            vm.Load += InitMainQuestions;

            vm.Error += OnError;

            vm.IsBusy = true;
        }

        /// <summary>
        /// Инициализируетблокспрофилемпользователя .
        /// </summary>
        private void InitProfile(object sender, EventArgs e)
        {
            if (!(BindingContext is MainInfoViewModel vm)) return;

            if (LoadingStatus.IsVisible)
                LoadingStatus.IsVisible = false;

            if (!Content.IsVisible)
            {
                Content.IsVisible = true;
                CarouselView.SetBinding(ItemsView.ItemsSourceProperty, "Notices");
            }

            if (!EventsTitle.IsVisible)
                EventsTitle.IsVisible = true;

            if (!MainQuestionsTitle.IsVisible)
                MainQuestionsTitle.IsVisible = true;

            Name.Text = $"{vm.Profile.FirstName} {vm.Profile.LastName}";
            DormitoryName.Text = vm.Profile.Dormitory.Name;
        }
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

DormitoryAddress.Text = vm.Profile.Dormitory.Address;

Preferences.Set("id", vm.Profile.Id);
Preferences.Set("first_name", vm.Profile.FirstName);
Preferences.Set("last_name", vm.Profile.LastName);
Preferences.Set("dormitory_id", vm.Profile.Dormitory.Id);
Preferences.Set("dormitory_name", vm.Profile.Dormitory.Name);
Preferences.Set("room", vm.Profile.Room);
}

/// <summary>
/// Инициализирует блок главными сообщениями
/// </summary>
private void InitMainNotices(object sender, EventArgs e)
{
    CarouselView.Position = 0;

    StackImportantMessages.Children.Clear();

    if (!(BindingContext is MainInfoViewModel vm)) return;
    var importantNotices = vm.AllNotices.Where(notice => notice.IsImportant);

    foreach (var notice in importantNotices)
        StackImportantMessages.Children.Add(new ImportantNotice { PTitle = {Text =
            notice.Title}});
}

/// <summary>
/// Инициализирует блок событиями
/// </summary>
private void InitEvents(object sender, EventArgs e)
{
    Events.Children.Clear();

    if (!(BindingContext is MainInfoViewModel vm)) return;
    foreach (var eEvent in vm.Events)
        Events.Children.Add(new EventView
        {
            PTitle =
            {
                Text = eEvent.Title
            },
            PTargetDate = eEvent.TargetDate
        });
}

/// <summary>
/// Инициализирует блок главными вопросами

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

/// </summary>
private void InitMainQuestions(object sender, EventArgs e)
{
    MainQuestions.Children.Clear();

    if (!(BindingContext is MainInfoViewModel vm)) return;
    foreach (var mainQuestion in vm.MainQuestions)
    {
        var mainQuestionView = new MainQuestionView
        {
            PTitle = {Text = mainQuestion.Question},
            Answer = mainQuestion.Answer
        };

        mainQuestionView.Click += async (o, args) =>
        {
            await DisplayAlertОтвет("", mainQuestionView.Answer, "OK");
        };

        MainQuestions.Children.Add(mainQuestionView);
    }
}

/// <summary>
/// Вызываетсяпривозникновенииошибки .
/// </summary>
private void OnError(object sender, EventArgs e)
{
    ActivityIndicator.IsRunning = false;
    ResultText.Text = "Произошла" ошибка.";
}

/// <summary>
/// Открываетстраницуспрофилем .
/// </summary>
private async void MenuItem_OnClicked(object sender, EventArgs e)
{
    await Navigation.PushAsync(new ProfilePage());
}

/// <summary>
/// Срабатываетпринажатиинакарусельсновостями .
/// Получаетактивнуююновостьиоткрываетподробности .
/// </summary>
private async void TapGestureRecognizer_OnTapped(object sender, EventArgs e)
{
    if (!(BindingContext is MainInfoViewModel vm)) return;
    var currentItem = (Notice) CarouselView.CurrentItem

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

?? vm.Notices.First(notice => !notice.IsImportant);

```
await Navigation.PushAsync(new NoticePage
{
    PTitle =
    {
        Text = currentItem.Title
    },
    PDescription =
    {
        Text = currentItem.Description
    },
    PDate =
    {
        Text = currentItem.CreatedAtBeauty
    }
});
}
```

### 1.1.13. MainPage.xaml

```
<?xml version="1.0" encoding="utf-8"?>

<TabbedPage xmlns="http://xamarin.com/schemas/2014/forms" xmlns:x="http://schemas.
microsoft.com/winfx/2009/xaml"
    xmlns:d="http://xamarin.com/schemas/2014/forms/design"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" mc:
        Ignorable="d"
    xmlns:views="clr-namespace:HSESupporter.Views" x:Class="HSESupporter.Views.
        MainPage">
    <TabbedPage.Children>
        <NavigationPage TitleГлавная="">
            <NavigationPage.Icon>
                <OnPlatform x:TypeArguments="FileImageSource">
                    <On Platform="iOS" Value="tab_feed.png" />
                </OnPlatform>
            </NavigationPage.Icon>
            <x:Arguments>
                <views:MainInfoPage />
            </x:Arguments>
        </NavigationPage>
        <NavigationPage TitleОбращения="">
            <NavigationPage.Icon>
                <OnPlatform x:TypeArguments="FileImageSource">
                    <On Platform="iOS" Value="problems.png" />
                </OnPlatform>
            </NavigationPage.Icon>
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        <x:Arguments>
            <views:ItemsPage />
        </x:Arguments>
    </NavigationPage>
    <NavigationPage TitleБеседа="">
        <NavigationPage.Icon>
            <OnPlatform x:TypeArguments="FileImageSource">
                <On Platform="iOS" Value="messages.png" />
            </OnPlatform>
        </NavigationPage.Icon>
        <x:Arguments>
            <views:ChatPage />
        </x:Arguments>
    </NavigationPage>
    <NavigationPage TitleO=" приложения">
        <NavigationPage.Icon>
            <OnPlatform x:TypeArguments="FileImageSource">
                <On Platform="iOS" Value="tab_about.png" />
            </OnPlatform>
        </NavigationPage.Icon>
        <x:Arguments>
            <views:AboutPage />
        </x:Arguments>
    </NavigationPage>
</TabPage.Children>
</TabPage>

```

#### 1.1.14. MainPage.xaml.cs

```

using System.ComponentModel;
using Xamarin.Essentials;
using Xamarin.Forms;

```

```

namespace HSESupporter.Views
{

```

```

    // Learn more about making custom code visible in the Xamarin.Forms previewer
    // by visiting https://aka.ms/xamarinforms-previewer
    [DesignTimeVisible(false)]

```

```

    public partial class MainPage : TabbedPage
    {

```

```

        public MainPage()
        {

```

```

            if (Preferences.Get("is__accept", true).Equals(false))
                Navigation.PushModalAsync(new WaitPage());
            else if (Preferences.Get("token", "").Equals(""))
                Navigation.PushModalAsync(new LoginPage());
            else
                InitializeComponent();

```

```

        }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    }
}

```

### 1.1.15.NewItemPage.xaml

```

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

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms" xmlns:x="http://schemas.
microsoft.com/winfx/2009/xaml"
    xmlns:d="http://xamarin.com/schemas/2014/forms/design"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" mc:
        Ignorable="d"
    x:Class="HSESupporter.Views.NewItemPage" TitleНовая=" проблема">
    <ContentPage.ToolbarItems>
        <!--<ToolbarItem TextОтмена="" Clicked="Cancel_Clicked" />-->
        <ToolbarItem TextСоздать="" Clicked="Save_Clicked" />
    </ContentPage.ToolbarItems>
    <ContentPage.Content>
        <StackLayout Spacing="20" Padding="15">
            <Label TextЧто=" васбеспокоит?" FontSize="Medium" />
            <Entry x:Name="Title" PlaceholderОпишите=" в двух словах" FontSize="Small" />
            <Label TextРасскажите=" подробнее..." FontSize="Medium" />
            <Editor x:Name="Description" AutoSize="TextChanges"
                PlaceholderНачните=" рассказывать о подробностях тут ..." FontSize="
                Small" Margin="0" />
            <Label x:Name="ResultText" HorizontalTextAlignment="Center"
                TextУспешно=""
                FontSize="20"
                IsVisible="False"
                FontAttributes="Bold"
                TextColor="#5B69FF" />
        </StackLayout>
    </ContentPage.Content>
</ContentPage>

```

### 1.1.16. NewItemPage.xaml.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using HSESupporter.Models;
using HSESupporter.Services;
using Xamarin.Forms;

namespace HSESupporter.Views
{
    // Learn more about making custom code visible in the Xamarin.Forms previewer
    // by visiting https://aka.ms/xamarinforms-previewer
    [DesignTimeVisible(false)]
    public partial class NewItemPage : ContentPage

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
{
    public NewItemPage()
    {
        InitializeComponent();

        Item = new Item
        {
            Text = "Item name",
            Description = "This is an item description."
        };

        BindingContext = this;
    }

    public Item Item { get; set; }

    private async void Save_Clicked(object sender, EventArgs e)
    {
        try
        {
            MessagingCenter.Send(this, "AddItem", Item);

            var api = new ApiService().HseSupporterApi;
            await api.SaveProblem(new Dictionary<string, object>
            {
                {"title", Title.Text},
                {"description", Description.Text},
                {"status", "O"}
            });
            await Navigation.PopAsync();
        }
        catch (Exception ex)
        {
            ResultText.IsVisible = true;
            ResultText.Text = "Произошла ошибка.";
        }
    }

    private async void Cancel_Clicked(object sender, EventArgs e)
    {
        await Navigation.PopAsync();
    }
}
```

#### 1.1.17. NoticePage.xaml

<?xml version="1.0" encoding="utf-8"?>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:d="http://xamarin.com/schemas/2014/forms/design"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d"
  TitleНовость=""
  x:Class="HSESupporter.Views.NoticePage">
  <ContentPage.Content>
    <ScrollView>
      <StackLayout Spacing="20" Padding="10">
        <Label x:Name="Title" d:Text="Item name" FontSize="22" FontAttributes="
          Bold" />
        <Label x:Name="Description" Padding="4" BackgroundColor="AliceBlue"
          d:Text="Item description" FontSize="18" />
        <StackLayout Orientation="Horizontal">
          <Label TextДата=" добавления:" FontSize="16" />
          <Label x:Name="Date" Textв=" обработке" Padding="1" FontSize="16"
            />
        </StackLayout>
      </StackLayout>
    </ScrollView>
  </ContentPage.Content>
</ContentPage>

```

#### 1.1.18. NoticePage.xaml.cs

```

using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace HSESupporter.Views
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class NoticePage : ContentPage
    {
        public NoticePage()
        {
            InitializeComponent();
        }

        public Label PTitle
        {
            get => Title;
            set => Title = value;
        }

        public Label PDescription
        {
            get => Description;
            set => Description = value;
        }
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```

    }

    public Label PDate
    {
        get => Date;
        set => Date = value;
    }
}

```

### 1.1.19. ProfilePage.xaml

```

<?xml version="1.0" encoding="utf-8"?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:viewmodels="clr-namespace:HSESupporter.ViewModels"
    TitleПрофиль=""
    x:Class="HSESupporter.Views.ProfilePage">
    <ContentPage.BindingContext>
        <viewmodels:ProfileViewModel />
    </ContentPage.BindingContext>
    <ContentPage.Content>
        <StackLayout Padding="16">
            <Label TextВаши=" имяифамилия" FontSize="Medium" />
            <Entry x:Name="FirstName"
                PlaceholderВаше=" имя"
                Text="{Binding FirstName}"
                IsReadOnly="True"
                FontSize="Small" />
            <Entry x:Name="LastName"
                PlaceholderВаше=" имя"
                Text="{Binding LastName}"
                IsReadOnly="True"
                FontSize="Small" />
            <Label TextОбщежитие="" FontSize="Medium" />
            <Picker x:Name="DormitoryList" />
            <Label TextНомер=" комнаты" FontSize="Medium" />
            <Entry x:Name="RoomNumber"
                PlaceholderНомер=" комнаты"
                Text="{Binding Room}"
                IsReadOnly="False"
                FontSize="Small" />

            <Button
                HorizontalOptions="CenterAndExpand"
                FontSize="16"
                Clicked="SaveButton_OnClicked"
                TextСохранить="" />

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

<ActivityIndicator x:Name="LoadingIndicator"
    IsRunning="False" />

<Label x:Name="ResultText" HorizontalTextAlignment="Center"
    TextУспешно=""
    FontSize="20"
    IsVisible="False"
    FontAttributes="Bold"
    TextColor="#5B69FF" />
</StackLayout>
</ContentPage.Content>
</ContentPage>

```

### 1.1.20. ProfilePage.xaml.cs

```

using System;
using System.Collections.Generic;
using HSESupporter.Services;
using HSESupporter.ViewModels;
using Xamarin.Essentials;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace HSESupporter.Views
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class ProfilePage : ContentPage
    {
        public ProfilePage()
        {
            InitializeComponent();
            if (!(BindingContext is ProfileViewModel vm)) return;

            DormitoryList.ItemsSource = vm.Dormitories;
            DormitoryList.SelectedIndex = vm.Dormitories.IndexOf(Preferences.Get("dormitory_name", ""));
        }

        private async void SaveButton_OnClicked(object sender, EventArgs e)
        {
            try
            {
                var values = new Dictionary<string, object>
                {
                    {"dormitory", DormitoryList.SelectedItem},
                    {"room", RoomNumber.Text}
                };

                LoadingIndicator.IsRunning = true;
            }
        }
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

var api = new ApiService().HseSupporterApi;
var result = await api.SetProfile(values);
LoadingIndicator.IsRunning = false;

ResultText.Text = "Сохранено";
ResultText.IsVisible = true;
}
catch (Exception ex)
{
    LoadingIndicator.IsRunning = false;

    ResultText.Text = "Произошла ошибка";
    ResultText.IsVisible = true;
}
}
}
}
}

```

#### 1.1.21. WaitPage.xaml

```

<?xml version="1.0" encoding="utf-8"?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:viewModels="clr-namespace:HSESupporter.ViewModels;assembly=
        HSESupporter"
    x:Class="HSESupporter.Views.WaitPage">
    <ContentPage.BindingContext>
        <viewModels.WaitViewModel />
    </ContentPage.BindingContext>
    <ContentPage.Content>
        <StackLayout
            Padding="12"
            VerticalOptions="Center">

            <Label Text="Здравствуйте!"
                HorizontalTextAlignment="Center"
                FontSize="26"
                VerticalOptions="Center"
                HorizontalOptions="Center" />
            <Label Text="Владислав", поделитесь вашими данными ."
                HorizontalTextAlignment="Center"
                HorizontalOptions="CenterAndExpand"
                FontSize="16" />
            <Label Text="После сохранения мы дополнительно всё проверим ,
                чтобы не было ошибок ."
                HorizontalTextAlignment="Center"
                HorizontalOptions="CenterAndExpand"
                FontSize="14" />

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

<Label TextОбщежитие="" FontSize="Medium" />
<Picker x:Name="DormitoryList"
        TitleВыберите=" из списка " />
<Label TextНомер=" комнаты" FontSize="Medium" />
<Entry x:Name="RoomNumber"
        PlaceholderВведите=" его здесь "
        Text=""
        IsReadOnly="False"
        FontSize="Small" />

<Button
        Clicked="SaveButton_OnClicked"
        HorizontalOptions="Center"
        FontSize="18"
        TextПодтвердить="" />

<ActivityIndicator x:Name="LoadingIndicator"
        IsRunning="False" />

<Label x:Name="ResultText" HorizontalTextAlignment="Center"
        TextУспешно=""
        FontSize="20"
        IsVisible="False"
        FontAttributes="Bold"
        TextColor="#5B69FF" />

</StackLayout>
</ContentPage.Content>
</ContentPage>

```

#### 1.1.22. WaitPage.xaml.cs

```

using System;
using System.Collections.Generic;
using System.IO;
using HSESupporter.Services;
using HSESupporter.ViewModels;
using Xamarin.Essentials;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

```

```

namespace HSESupporter.Views
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class WaitPage : ContentPage
    {
        public WaitPage()
        {

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

InitializeComponent();
if (!(BindingContext is WaitViewModel vm)) return;

DormitoryList.ItemsSource = vm.Dormitories;
}

private async void SaveButton_OnClicked(object sender, EventArgs e)
{
    try
    {
        if (DormitoryList.SelectedIndex == -1 || RoomNumber.Text.Equals(""))
            throw new InvalidDataException();

        var values = new Dictionary<string, object>
        {
            {"dormitory", DormitoryList.SelectedItem},
            {"room", RoomNumber.Text}
        };

        LoadingIndicator.IsRunning = true;
        var api = new ApiService().HseSupporterApi;
        var result = await api.SetProfile(values);

        Preferences.Set("dormitory_id", result.Dormitory.Id);

        ResultText.Text = "Сохранено";

        Preferences.Set("is_accept", true);
        await Navigation.PushModalAsync(new MainPage());
    }
    catch (InvalidDataException ex)
    {
        ResultText.Text = "Заполните все поля";
    }
    catch (Exception ex)
    {
        ResultText.Text = "Произошла ошибка";
    }
    finally
    {
        LoadingIndicator.IsRunning = false;
        ResultText.IsVisible = true;
    }
}
}
}
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 1.2. ViewModels

### 1.2.1. AboutViewModel.cs

```
using System.Windows.Input;
using Xamarin.Essentials;
using Xamarin.Forms;

namespace HSESupporter.ViewModels
{
    public class AboutViewModel : BaseViewModel
    {
        public AboutViewModel()
        {
            Title = "О" + " приложении";
            OpenWebCommand = new Command(async () => await Browser.OpenAsync("https
                ://www.hse.ru/"));
        }

        public ICommand OpenWebCommand { get; }
    }
}
```

### 1.2.2. BaseViewModel.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Runtime.CompilerServices;
using HSESupporter.Models;
using HSESupporter.Services;
using Xamarin.Forms;

namespace HSESupporter.ViewModels
{
    public class BaseViewModel : INotifyPropertyChanged
    {
        private bool _isBusy;

        private string _title = string.Empty;
        public IDataStore<Item> DataStore => DependencyService.Get<IDataStore<Item>>();

        public bool IsBusy
        {
            get => _isBusy;
            set
            {
                SetProperty(ref _isBusy, value);
                OnBusyChanged(value);
            }
        }
    }
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    }
}

public string Title
{
    get => _title;
    set => SetProperty(ref _title, value);
}

protected bool SetProperty<T>(ref T backingStore, T value,
    [CallerMemberName] string propertyName = "",
    Action onChanged = null)
{
    if (EqualityComparer<T>.Default.Equals(backingStore, value))
        return false;

    backingStore = value;
    onChanged?.Invoke();
    OnPropertyChanged(propertyName);
    return true;
}

public virtual void OnBusyChanged(bool value)
{
}

#region INotifyPropertyChanged

public event PropertyChangedEventHandler PropertyChanged;

protected void OnPropertyChanged([CallerMemberName] string propertyName = "")
{
    var changed = PropertyChanged;
    if (changed == null)
        return;

    changed.Invoke(this, new PropertyChangedEventArgs(propertyName));
}

#endregion
}

```

### 1.2.3. ChatViewModel.cs

```

namespace HSESupporter.ViewModels
{
    public class ChatViewModel : BaseViewModel
    {

```

ИЗМ.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        public ChatViewModel()
        {
            Title = "Беседа";
        }
    }
}

```

#### 1.2.4. ItemDetailViewModel.cs

```

using System;
using HSESupporter.Models;
using Xamarin.Forms;

namespace HSESupporter.ViewModels
{
    public class ItemDetailViewModel : BaseViewModel
    {
        public ItemDetailViewModel(Problem item = null)
        {
            Title = item?.Title;
            Item = item;

            RefreshCommand = new Command(InitItem);
        }

        public Command RefreshCommand { get; set; }

        public Problem Item { get; set; }

        public event EventHandler Load;

        private void InitItem()
        {
            OnLoad();
        }

        public void OnLoad()
        {
            Load?.Invoke(this, EventArgs.Empty);
        }
    }
}

```

#### 1.2.5. ItemsViewModel.cs

```

using System;
using System.Collections.ObjectModel;
using System.Diagnostics;
using System.Threading.Tasks;
using HSESupporter.Models;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```

using HSESupporter.Services;
using Xamarin.Forms;

namespace HSESupporter.ViewModels
{
    public class ItemsViewModel : BaseViewModel
    {
        public ItemsViewModel()
        {
            Title = "Обращения";
            Items = new ObservableCollection<Problem>();
            LoadItemsCommand = new Command(async () => await
                ExecuteLoadItemsCommand());
        }

        public ObservableCollection<Problem> Items { get; set; }
        public Command LoadItemsCommand { get; set; }

        private async Task ExecuteLoadItemsCommand()
        {
            if (IsBusy)
                return;

            IsBusy = true;

            try
            {
                Items.Clear();

                var api = new ApiService().HseSupporterApi;
                var items = await api.GetProblems();
                foreach (var item in items) Items.Add(item);
            }
            catch (Exception ex)
            {
                Debug.WriteLine(ex);
            }
            finally
            {
                IsBusy = false;
            }
        }
    }
}

```

#### 1.2.6. MainInfoViewModel.cs

```

using System;
using System.Collections.Generic;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

using System.Collections.ObjectModel;
using System.Linq;
using System.Net.Http;
using HSESupporter.Models;
using HSESupporter.Services;
using Xamarin.Forms;

namespace HSESupporter.ViewModels
{
    public sealed class MainInfoViewModel : BaseViewModel
    {
        public enum ErrorType
        {
            NetworkError,
            ServerError,
            UnknownError
        }

        public MainInfoViewModel()
        {
            Title = Главная"";
            Notices = new ObservableCollection<Notice>();
            Events = new ObservableCollection<Event>();
            MainQuestions = new ObservableCollection<MainQuestion>();

            RefreshCommand = new Command(InitMainPage);
        }

        public Command RefreshCommand { get; set; }

        public Profile Profile { get; private set; }
        public List<Notice> AllNotices { get; private set; }
        public ObservableCollection<Notice> Notices { get; }
        public ObservableCollection<Event> Events { get; }
        public ObservableCollection<MainQuestion> MainQuestions { get; }

        public event EventHandler Load;
        public event EventHandler Error;

        public async void InitMainPage()
        {
            try
            {
                var api = new ApiService().HseSupporterApi;
                var mainPage = await api.GetMainPage();

                Notices.Clear();
                Events.Clear();
            }
        }
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

MainQuestions.Clear();

AllNotices = mainPage.Notices;

Profile = mainPage.Profile;
foreach (var notice in mainPage.Notices.Where(notice => !notice.IsImportant))
    Notices.Add(notice);
foreach (var eEvent in mainPage.Events)
    Events.Add(eEvent);
foreach (var mainQuestion in mainPage.MainQuestions)
    MainQuestions.Add(mainQuestion);

IsBusy = false;
OnLoad();
}
catch (HttpRequestException e)
{
    OnError(ErrorType.ServerError);
}
catch (Exception e)
{
    OnError(ErrorType.UnknownError);
}
}

public void OnLoad()
{
    Load?.Invoke(this, EventArgs.Empty);
}

public void OnError(ErrorType type)
{
    Error?.Invoke(this, EventArgs.Empty);
}
}
}

```

### 1.2.7. ProfileViewModel.cs

```

using System.Collections;
using System.Collections.Generic;
using Xamarin.Essentials;

namespace HSESupporter.ViewModels
{
    public class ProfileViewModel : BaseViewModel
    {
        public readonly IList Dormitories = new List<string>
        {Общежитие

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        " №1", Общежитие
        " №2", Общежитие
        " №3", Общежитие
        " №4", Общежитие
        " №5", Общежитие
        " №6", Общежитие
        " №7", Общежитие
        " №8 «Трилистник»", Общежитие
        " №9", Общежитие
        " №10", Общежитие
        " №11", Студенческий
        " городок Дубки "
    };

    public ProfileViewModel()
    {
        Title = Профиль "";
        FirstName = Preferences.Get("first_name", "");
        LastName = Preferences.Get("last_name", "");
        Room = Preferences.Get("room", "");
    }

    public string FirstName { get; }
    public string LastName { get; }
    public string Room { get; }
}
}

```

#### 1.2.8. WaitViewModel.cs

```

using System.Collections;
using System.Collections.Generic;

namespace HSESupporter.ViewModels
{
    public class WaitViewModel : BaseViewModel
    {
        public readonly IList Dormitories = new List<string>
        {
            Общежитие
            " №1", Общежитие
            " №2", Общежитие
            " №3", Общежитие
            " №4", Общежитие
            " №5", Общежитие
            " №6", Общежитие
            " №7", Общежитие
            " №8 «Трилистник»", Общежитие
            " №9", Общежитие
            " №10", Общежитие
        }
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        " №11",Студенческий
        " городокДубки "
    };

    public WaitViewModel()
    {
        Title = Профиль"";
    }
}

```

### 1.3. Services

#### 1.3.1. ApiService.cs

```

using System.Threading.Tasks;
using Refit;
using Xamarin.Essentials;

namespace HSESupporter.Services
{
    public class ApiService
    {
        public readonly IHseSupporterApi HseSupporterApi;

        public ApiService()
        {
            var refitSettings = new RefitSettings
            {
                AuthorizationHeaderValueGetter = () => Task.FromResult(TokenHeader)
            };
            HseSupporterApi = RestService.For<IHseSupporterApi>("https://hse-supporter.
                herokuapp.com", refitSettings);
        }

        public static string TokenHeader => $"{Preferences.Get("token", "")}";
    }
}

```

#### 1.3.2. Extensions.cs

```

using System;

namespace HSESupporter.Services
{
    public static class Extensions
    {
        public static string GetDateTimeText(this string str)
        {
            var dateTime = DateTime.Parse(str);

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        return dateTime.ToString("HH:mm");
    }
}

```

### 1.3.3. IHseSupporterApi.cs

```

using System.Collections.Generic;
using System.Threading.Tasks;
using HSESupporter.Models;
using Refit;

namespace HSESupporter.Services
{
    [Headers("Authorization: Token")]
    public interface IHseSupporterApi
    {
        [Headers("Authorization")]
        [Post("/auth/register/")]
        Task<AuthResult> Login([Body(BodySerializationMethod.UrlEncoded)]
            Dictionary<string, object> data);

        [Headers("Authorization")]
        [Post("/auth/register/confirm-email/")]
        Task<AuthResult> LoginConfirm([Body(BodySerializationMethod.UrlEncoded)]
            Dictionary<string, object> data);

        [Get("/auth/accept-status/")]
        Task<AuthResult> CheckAcceptStatus();

        [Get("/main_page/")]
        Task<MainPage> GetMainPage();

        [Get("/profile/")]
        Task<Profile> GetProfile();

        [Post("/profile/")]
        Task<Profile> SetProfile([Body(BodySerializationMethod.UrlEncoded)]
            Dictionary<string, object> data);

        [Get("/dormitories/")]
        Task<List<Dormitory>> GetDormitories();

        [Get("/dormitories/{id}/")]
        Task<Dormitory> GetDormitory(int id);

        [Get("/problems/")]
        Task<List<Problem>> GetProblems();
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

[Get("/notices/")]
Task<List<Notice>> GetNotices();

[Get("/messages/")]
Task<List<Message>> GetMessages();

[Post("/problems/")]
Task<List<Problem>> SaveProblem([Body(BodySerializationMethod.UrlEncoded)]
    Dictionary<string, object> data);

[Get("/problems/{id}/")]
Task<Problem> GetProblem(int id);

[Delete("/problems/{id}/")]
Task<Problem> DeleteProblem(int id);

[Post("/messages/")]
Task<List<Problem>> SendMessage([Body(BodySerializationMethod.UrlEncoded)]
    Dictionary<string, object> data);
}
}

```

## 1.4. Models

### 1.4.1. AuthResult.cs

```

using Newtonsoft.Json;

namespace HSESupporter.Models
{
    public class AuthResult
    {
        [JsonProperty("token")] public string Token { get; set; }
        [JsonProperty("is_accept")] public bool IsAccept { get; set; }
        [JsonProperty("profile")] public Profile Profile { get; set; }
    }
}

```

### 1.4.2. Dormitory.cs

```

using System.Collections.Generic;
using Newtonsoft.Json;

namespace HSESupporter.Models
{
    public class Dormitory : IModel
    {
        [JsonProperty("id")] public int Id { get; set; }
        [JsonProperty("name")] public string Name { get; set; }
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

[JsonProperty("address")] public string Address { get; set; }
[JsonProperty("messages")] public List<Message> Messages { get; set; }

public Dictionary<string, object> GetDictionaryParams()
{
    return new Dictionary<string, object>
    {
        {"id", Id},
        {"name", Name},
        {"address", Address},
        {"messages", Messages}
    };
}
}
}

```

### 1.4.3. Event.cs

```

using System;
using Newtonsoft.Json;

namespace HSESupporter.Models
{
    public class Event
    {
        [JsonProperty("id")] public int Id { get; set; }
        [JsonProperty("title")] public string Title { get; set; }
        [JsonProperty("description")] public string Description { get; set; }
        [JsonProperty("target_date")] public string TargetDate { get; set; }
        [JsonProperty("created_at")] public string CreatedAt { get; set; }
        [JsonProperty("updated_at")] public string UpdatedAt { get; set; }

        /// <summary>
        /// Дата проведения в формате читабельной строки .
        /// </summary>
        public string TargetDateBeauty
        {
            get
            {
                var dateTime = DateTime.Parse(CreatedAt);
                return dateTime.ToString("dd.MM.yyyy HH:mm");
            }
        }

        /// <summary>
        /// Дата создания в формате читабельной строки .
        /// </summary>
        public string CreatedAtBeauty
        {

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```

        get
        {
            var dateTime = DateTime.Parse(CreatedAt);
            return dateTime.ToString("dd.MM.yyyy HH:mm");
        }
    }
}

```

#### 1.4.4. IModel.cs

```

using System.Collections.Generic;

namespace HSESupporter.Models
{
    public interface IModel
    {
        Dictionary<string, object> GetDictionaryParams();
    }
}

```

#### 1.4.5. MainPage.cs

```

using System.Collections.Generic;
using Newtonsoft.Json;

namespace HSESupporter.Models
{
    public class MainPage
    {
        [JsonProperty("profile")] public Profile Profile { get; set; }
        [JsonProperty("notices")] public List<Notice> Notices { get; set; }
        [JsonProperty("events")] public List<Event> Events { get; set; }
        [JsonProperty("main_questions")] public List<MainQuestion> MainQuestions { get; set; }
    }
}

```

#### 1.4.6. MainQuestion.cs

```

using Newtonsoft.Json;

namespace HSESupporter.Models
{
    public class MainQuestion
    {
        [JsonProperty("question")] public string Question { get; set; }
        [JsonProperty("answer")] public string Answer { get; set; }
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**1.4.7. Message.cs**

```

using System.Collections.Generic;
using Newtonsoft.Json;

namespace HSESupporter.Models
{
    public class Message : IModel
    {
        [JsonProperty("id")] public int Id { get; set; }
        [JsonProperty("author")] public int Author { get; set; }
        [JsonProperty("author_first_name")] public string AuthorFirstName { get; set; }
        [JsonProperty("author_last_name")] public string AuthorLastName { get; set; }
        [JsonProperty("text")] public string Text { get; set; }
        [JsonProperty("is_read")] public bool IsRead { get; set; }
        [JsonProperty("is_from_student")] public bool IsFromStudent { get; set; }
        [JsonProperty("created_at")] public string CreatedAt { get; set; }
        [JsonProperty("updated_at")] public string UpdatedAt { get; set; }
        [JsonProperty("problem")] public int? Problem { get; set; }
        [JsonProperty("dormitory")] public int? Dormitory { get; set; }

        public Dictionary<string, object> GetDictionaryParams()
        {
            var values = new Dictionary<string, object>
            {
                {"id", Id},
                {"name", Author},
                {"text", Text},
                {"is_read", IsRead},
                {"is_from_student", IsFromStudent},
                {"created_at", CreatedAt},
                {"updated_at", UpdatedAt}
            };

            if (Problem != 0) values.Add("problem", Problem);
            if (Dormitory != 0) values.Add("dormitory", Dormitory);

            return values;
        }
    }
}

```

**1.4.8. Notice.cs**

```

using System;
using System.Collections.Generic;
using Newtonsoft.Json;

namespace HSESupporter.Models

```

ИЗМ.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

{
    public class Notice : IModel
    {
        [JsonProperty("id")] public int Id { get; set; }
        [JsonProperty("main_text")] public string Title { get; set; }
        [JsonProperty("text")] public string Description { get; set; }
        [JsonProperty("is_important")] public bool IsImportant { get; set; }
        [JsonProperty("created_at")] public string CreatedAt { get; set; }
        [JsonProperty("updated_at")] public string UpdatedAt { get; set; }

        public string CreatedAtBeauty
        {
            get
            {
                var dateTime = DateTime.Parse(CreatedAt);
                return dateTime.ToString("dd.MM.yyyy HH:mm");
            }
        }

        public Dictionary<string, object> GetDictionaryParams()
        {
            return new Dictionary<string, object>
            {
                {"id", Id},
                {"title", Title},
                {"description", Description},
                {"is_important", IsImportant},
                {"created_at", CreatedAt},
                {"updated_at", UpdatedAt}
            };
        }
    }
}

```

#### 1.4.9. Problem.cs

```

using System;
using System.Collections.Generic;
using Newtonsoft.Json;

namespace HSESupporter.Models
{
    public class Problem : IModel
    {
        [JsonProperty("id")] public int Id { get; set; }
        [JsonProperty("author")] public string Author { get; set; }
        [JsonProperty("title")] public string Title { get; set; }
        [JsonProperty("description")] public string Description { get; set; }
        [JsonProperty("status")] public string Status { get; set; }
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

[JsonProperty("created_at")] public string CreatedAt { get; set; }
[JsonProperty("updated_at")] public string UpdatedAt { get; set; }
[JsonProperty("messages")] public List<Message> Messages { get; set; }

public string CreatedAtBeauty
{
    get
    {
        var dateTime = DateTime.Parse(CreatedAt);
        return dateTime.ToString("dd MMMM HH:mm");
    }
}

public Dictionary<string, object> GetDictionaryParams()
{
    return new Dictionary<string, object>
    {
        {"id", Id},
        {"name", Author},
        {"title", Title},
        {"description", Description},
        {"status", Status},
        {"created_at", CreatedAt},
        {"updated_at", UpdatedAt},
        {"message", Messages}
    };
}
}
}

```

#### 1.4.10. Profile.cs

```

using Newtonsoft.Json;

namespace HSESupporter.Models
{
    public class Profile
    {
        [JsonProperty("id")] public int Id { get; set; }
        [JsonProperty("first_name")] public string FirstName { get; set; }
        [JsonProperty("last_name")] public string LastName { get; set; }
        [JsonProperty("info")] public string Info { get; set; }
        [JsonProperty("role")] public string Role { get; set; }
        [JsonProperty("room")] public string Room { get; set; }
        [JsonProperty("dormitory")] public Dormitory Dormitory { get; set; }
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

# Лист регистрации изменений

[illegible]