Составил: Садовский Р.В.

Практическая работа №27
Разработка приложения

1 Цель работы

1.1 Закрепить навыки разработки приложений на Avalonia UI.

2 Литература

2.1 Avalonia documentation – Текст : электронный // AvaloniaUI, 2024. – URL: https://docs.avaloniaui.net/

3 Подготовка к работе

3.1 Повторить теоретический материал (см.п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

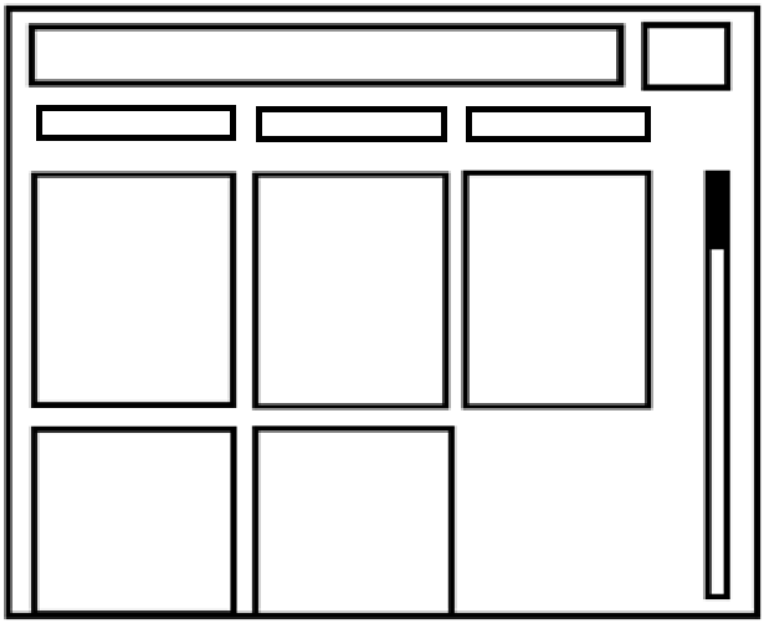
5 Задание

В работе используется приложение из ПР №27

5.1 Модификация интерфейса приложения

5.1.1 На странице расположите поле ввода, кнопку Поиск, список выбора городов и список прогнозов погоды.

5.1.2 Прогноз погоды должен отображаться в виде карточки, карточки отображаются в виде списка. Общий вид интерфейса должен соответствовать макету:



Также учтите адаптивность интерфейса под мобильные устройства

5.1.3 Для списка карточек используйте ItemsControl. В качестве значения для ItemsControl.ItemsPanel установите WrapPanel.

```
<ItemsControl.ItemsPanel>
    <ItemsPanelTemplate>
        <WrapPanel/>
    </ItemsPanelTemplate>
</ItemsControl.ItemsPanel>
```

5.1.4 ItemsControl расположите внутри ScrollViewer

5.2 Добавление записей в список

Пользователь должен вводить название города в поле ввода, нажимать на кнопку «Поиск» и после этого должен заполняться список городов на основе значений, которые вернуло API. Пользователь выбирает один из городов, который будет добавлен в список прогнозов погоды.

5.3 Сохранение данных

После добавления новой карточки необходимо сохранить на устройстве список городов, карточки которых отображаются в приложении

5.4 Для сохранения данных требуется использовать класс Preferences в проекте. Ознакомьтесь с его содержимым.

5.4.1 Для сохранения данных воспользоваться методом

```
await Preferences.Save("ключ", данные)
```

5.4.2 При запуске приложения загружайте данные из списка при помощи метода

```
var data = await Preferences.Load("ключ", знач_по_умолч)
```

5.4.3 Для выполнения асинхронных операций в конструкторе ViewModel используйте

```
TaskNotifier? _initializationTask;
public Task? InitializationTask
{
    get => _initializationTask;
    set      =>      SetPropertyAndNotifyOnCompletion(ref
_initializationTask, value);
}

public WeatherViewModel(WeatherService weatherService)
{
    _weatherService = weatherService;

    InitializationTask = Task.Run(async () =>
    {
        //Асинхронный код
    });
}
```

5.5 Обновление данных

5.5.1 Добавить на карточку кнопки «Обновить» и «Удалить»

5.5.2 При нажатии «Удалить» удаляйте карточку из списка прогнозов

5.5.3 При нажатии «Обновить» обновляйте значения данных прогноза до актуальных на текущий момент.

6 Порядок выполнения работы

6.1 Выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Как использовать асинхронные операции при инициализации объекта?

8.2 Как работает класс Preferences, предоставленный преподавателем?