



Универзитет „св. Кирил и Методиј“ - Скопје
Факултет за Информатички Науки и Компјутерско Инженерство

Проектна задача по предметот:

Визуелно Програмирање

Travel Planner

Професор:
д-р Дејан Ѓорѓевиќ

Изработила:
Андреа Лапевска 181065

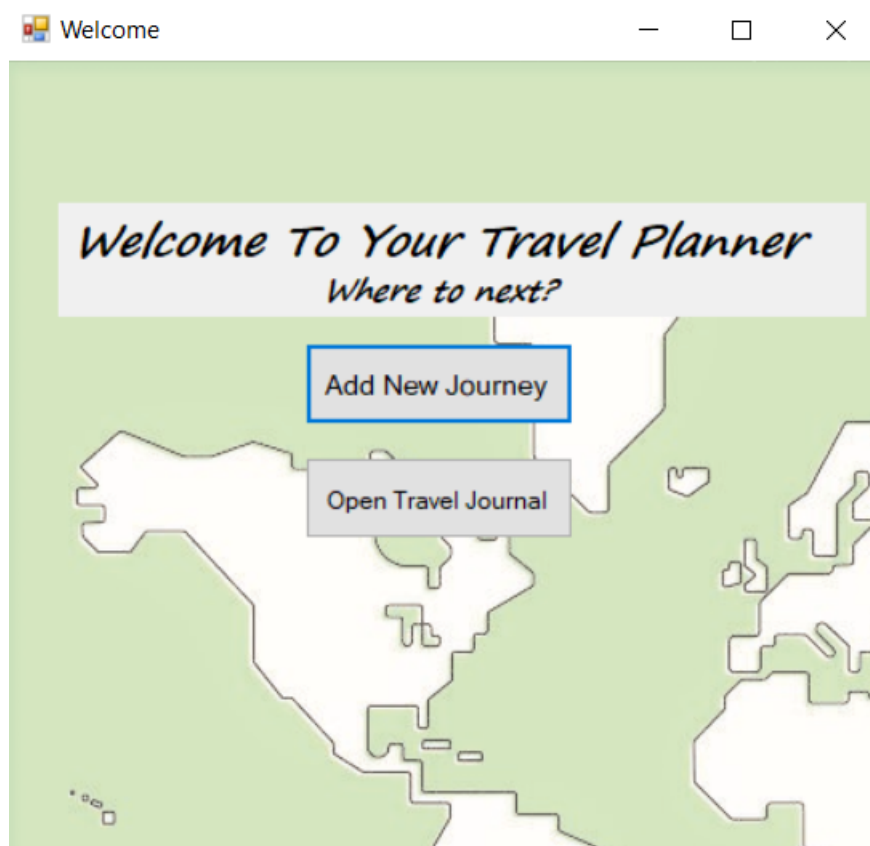
Асистент:
м-р Стефан Андонов

Септември 2020

Опис на апликацијата

Апликацијата Travel Planner е едноставна алатка за организација на патувања. Таа претставува еден вид на бележник (т.е. планер) во кој се запишуваат најбитните аспекти при организирање на едно патување.

При стартување на апликацијата, на корисникот му се прикажува почетната форма која му нуди избор за внесување на детали за ново патување или пак отварање на Travel Journal, односно дневникот кој ги содржи информациите за сите претходни патувања.



При клик на копчето Add New Journey се отвара форма за внесување на податоци за следното патување, односно од каде до каде се патува, датумот на поаѓање и враќање, каде ќе биде сместен и соодветните трошоци. Исто така, корисникот може да направи листа од предмети кои треба да ги спакува како и листа со места кои треба да ги посети. Изгледот на формата е прикажан на следната слика:

New Journey

Starting Point:

Destination:

Accommodation:

Packing List:

Add To Packing

Sights to see:

Add To List

Save

Transport cost:

Accommodation cost:

Pocket Money:

Date:

September 2020

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

Today: 9/5/2020

Копчето Save го враќа корисникот на почетниот прозорец од каде при кликување на копчето Open Travel Journal може да пристапи до листа од претходните патувања и при избирање на елемент од листата се прикажуваат внесените детали за тоа патување.

My journeys

Skopje-Paris, 9/20/2020-9/26/2020

Berovo -Ohrid, 8/10/2020-8/15/2020

Skopje-Ljubljana, 11/10/2020-11/17/2020

Back

Starting Point: Skopje

Destination: Paris

Accommodation: AirBnB

Cost: 800

From: 9/20/2020 To: 9/26/2020

Packing list:

- ☐ passport
- ☐ sunglasses
- ☐ wallet
- ☐ toothbrush

Places to visit:

- ☐ Eiffel Tower
- ☐ Louvre
- ☐ Jardins de Lux

Close

Како работи апликацијата?

Апликацијата е составена од 4 Windows Forms форми за приказ на корисничкиот интерфејс. Back-end делот на апликацијата е пишуван со програмскиот јазик C#. Во склоп на апликацијата е направена и класата Journey која има својства (properties) кои опишуваат едно патување.

```
[Serializable()]
8 references
public class Journey
{
    3 references
    public string StartingPoint { get; set; }
    3 references
    public string Destination { get; set; }
    2 references
    public string Accomodation { get; set; }
    2 references
    public float Budget { get; set; }
    3 references
    public string From { get; set; }
    3 references
    public string To { get; set; }
    3 references
    public List<string> ToPack { get; set; }
    3 references
    public List<string> ToVisit { get; set; }
    0 references
    public Journey() { }
```

Како што може да се забележи, класата Journey е сериализибилна, односно објект од оваа класа и неговата состојба можеме да ги зачуваме во датотека со цел податоците кои ги внесува корисникот да не се изгубат при исклучување на апликацијата. Сите патувања кои ги внесува корисникот се зачуваваат во листа со објекти од тип Journey која подоцна се сериализира. Тоа го правиме со помош на методите Serialize() и Deserialize(). Во оваа апликација се користи XmlSerializer, иако истото може да се постигне и со BinaryFormatter.

```
public static void Serialize()
{
    try
    {
        XmlSerializer serializer = new XmlSerializer(journeys.GetType());
        using (StreamWriter sw = new StreamWriter("journeys.xml"))
        {
            serializer.Serialize(sw, journeys);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

```
public static void Deserialize()
{
    try
    {
        if (File.Exists("journeys.xml"))
        {
            XmlSerializer serializer = new XmlSerializer(journeys.GetType());
            using (StreamReader sr = new StreamReader("journeys.xml"))
            {
                journeys = (List<Journey>)serializer.Deserialize(sr);
            }
        }
        else throw new FileNotFoundException("File not found");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

За прикажување на податоците, премин од една форма до друга како и внесување на податоци се користат алатки од Toolbox кои се програмираат со соодветните настани како што се `mouseClick`, `SelectedIndexChanged`, `Load` и слично.