

Studiarea bibliotecii PyQt5 pentru limbajul de programare Python si crearea unui aplicatii de tip music player

ELABORAT: ST.GR.TI-194, ZAVOROT DANIEL

Analiza domeniului

- Aplicația care a fost realizată este de tip desktop creată în limbajul Python
- Pentru realizarea acestei aplicații s-a utilizat biblioteca Qt (PyQt5). De asemenea a fost folosită aplicația Qt Designer ce permite crearea GUI-urilor în real-time cu ajutorul funcției (drag 'n drop).

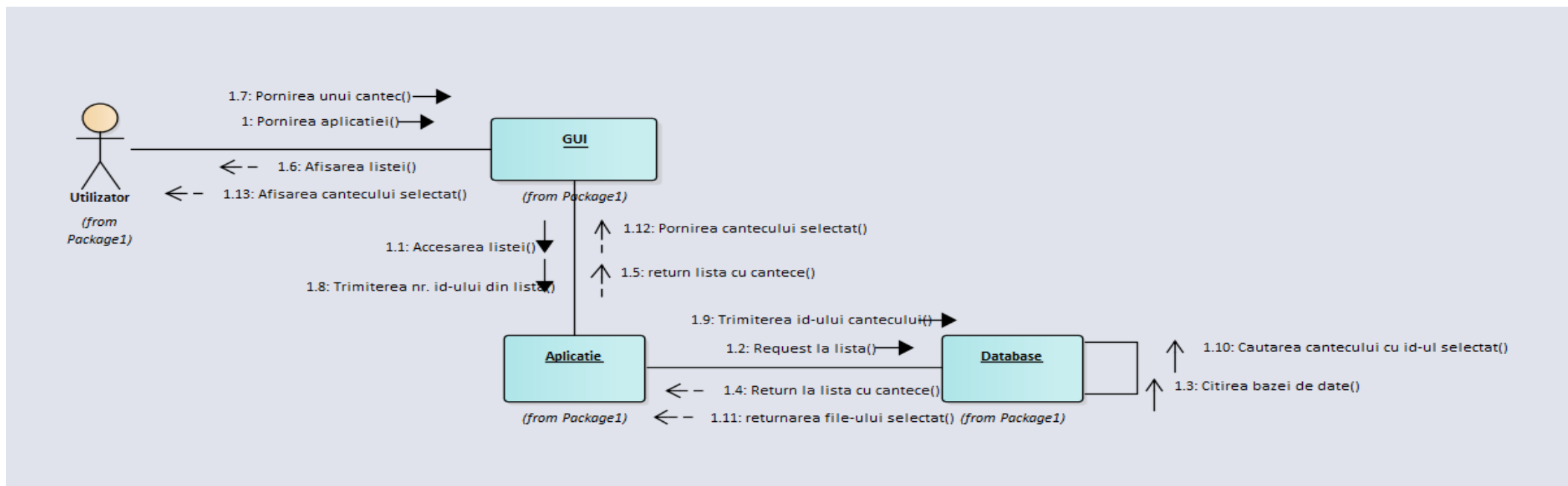
Scopul si Obiectivele

Principalul scop al acestui proiect este crearea unei aplicații de tip music player cu un funcțional de baza mediu. Design-ul va fi de culoare inchisa, cu culori predominante negru/gri/verde. De asemenea va avea window bar-ul sau propriu, care va fi de culoare inchisa, deoarece windows nu permite schimbarea bar-ului care sta la baza tuturor aplicatiilor. Dupa functional va avea incarcarea, stergerea, editarea fisierilor cu extensia .mp3. De asemenea va avea posibilitatea incarcarii unui cover pentru piesa aleasa. Ca si orice alt player pentru muzica, va avea posibilitatea de controlare a pieselor, adica va contine shuffle mode, repeat once, repeat this si desigur normal mode. Va contine slide pentru volum si un slide pentru controlarea cantecului curent.

Obiectivele de baza sunt:

1. Realizarea unei aplicații de tip music player
2. Optimizarea maximala a codului
3. Executarea non-stop a programului chiar daca a aparut vreo greseala

Imaginea generala a sistemului



Interacțiunea user-ului cu sistemul cu ajutorul GUI-ului realizat

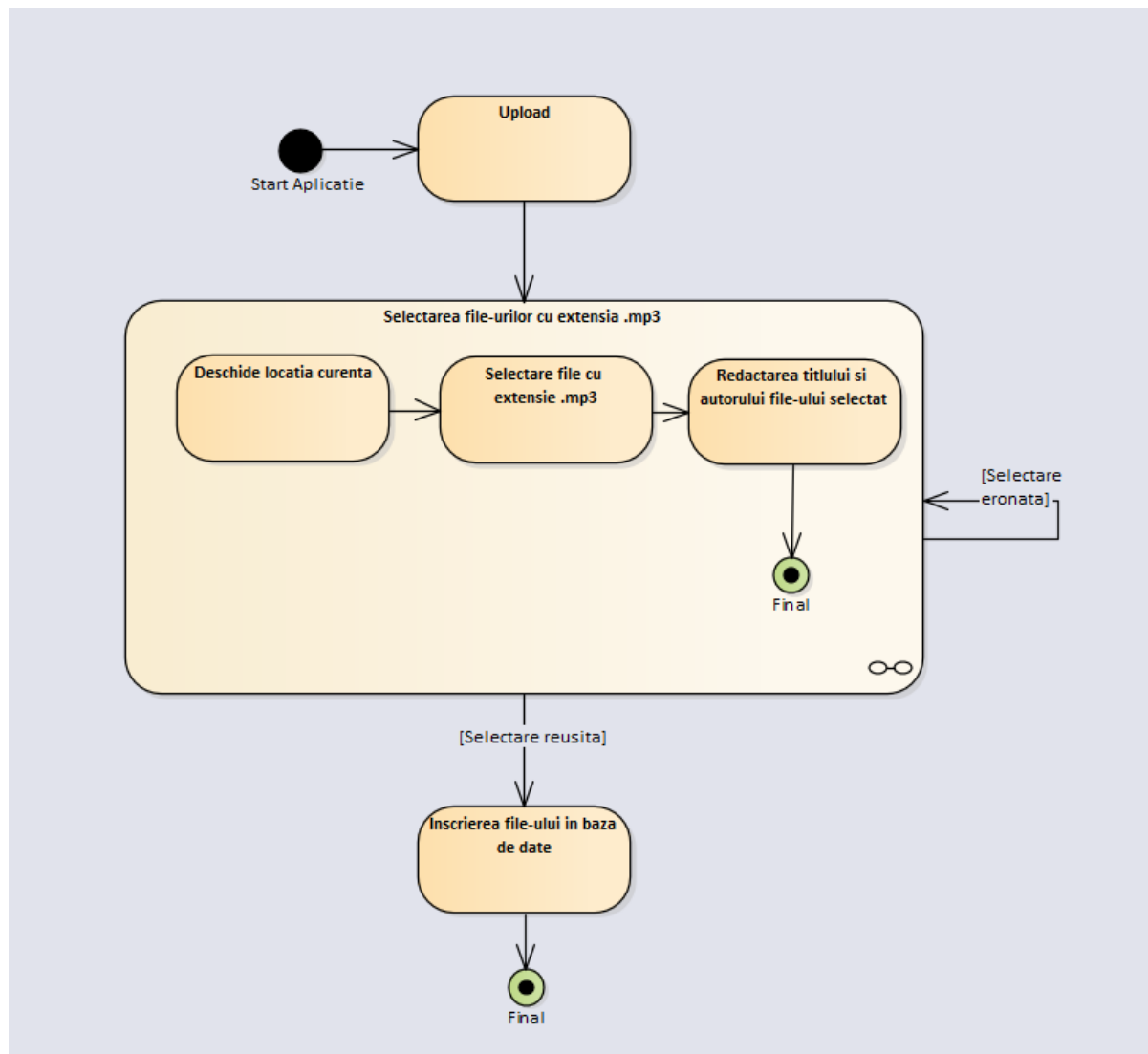


Diagrama de stare a procesului de selectare file

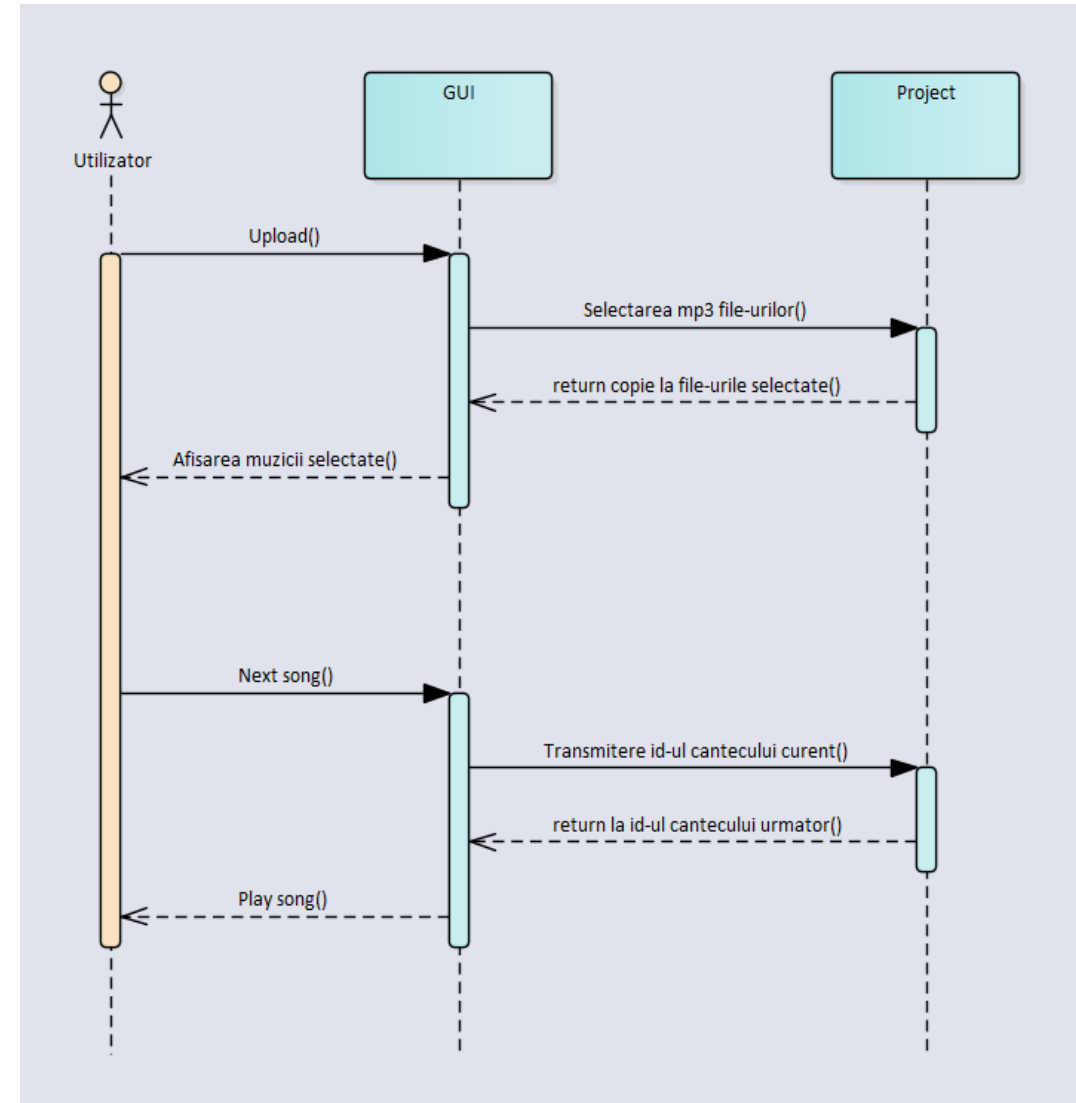
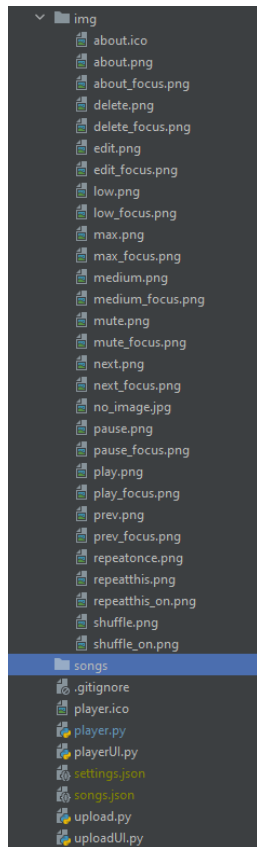


Diagrama de secvență a incarcarii unui cantec in aplicatie

Structura proiectului



```
# read songs from songs.json
def read_songs_from_json(self):
    if not os.path.exists('songs'):
        os.makedirs('songs')
    self.player = QMediaPlayer()
    self.playlist = QMediaPlaylist(self.player)
    try:
        with open("songs.json", "r", encoding="utf-8") as file:
            data = json.load(file)
        self.titles.clear()
        self.artists.clear()
        self.covers.clear()
        for i in data["Songs"]:
            # title
            self.titles.append(i["title"])
            # artist
            self.artists.append(i["artist"])
            # cover
            if i["cover"] == "Undefined":
                self.covers.append("no_image.jpg")
            else:
                self.covers.append(i["cover"])
    except Exception as e:
        print(e)
    self.read_files_songs()
```

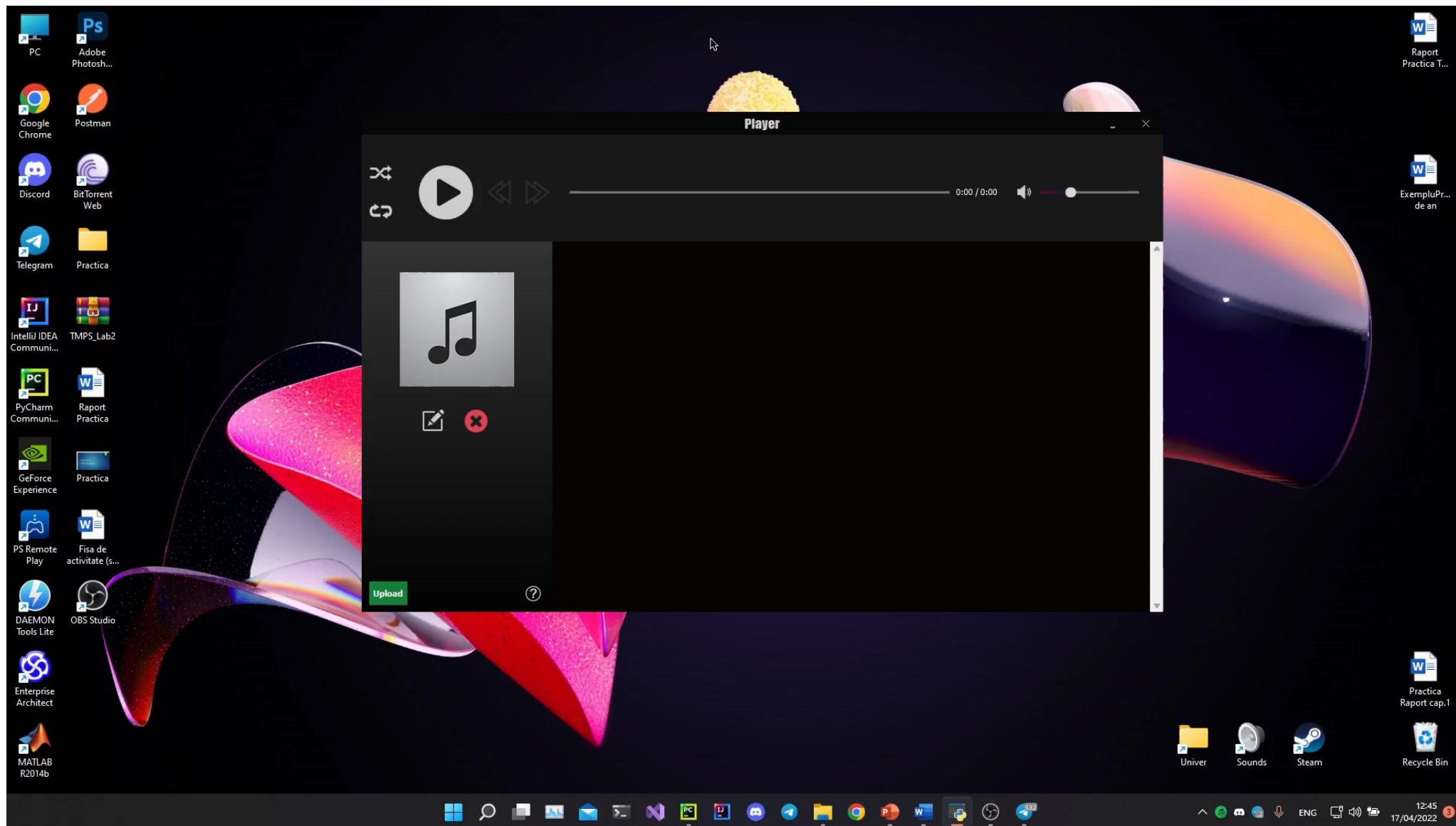
```
{
  "Songs": [
    {
      "id": 0,
      "title": "All Night",
      "artist": "2Scratch",
      "cover": "Undefined"
    },
    {
      "id": 1,
      "title": "Sober",
      "artist": "2Scratch",
      "cover": "Undefined"
    },
    {
      "id": 2,
      "title": "Channel Dark Side",
      "artist": "Blind",
      "cover": "Undefined"
    },
    {
      "id": 3,
      "title": "Livin' la Vida Loca",
      "artist": "COMETRON",
      "cover": "Undefined"
    }
  ]
}
```

Instrumente si tehnologii utilizate

Pentru realizarea acestei aplicatii am ales limbajul de programare Python si biblioteca PyQt5. De asemenea am folosit file-urile cu extensia .json care servesc ca baza de date NoSQL pentru aplicatia data. Nu in ultimul rand a fost utilizat aplicatia Qt Designer care permite crearea GUI-ului mai usor, de asemenea GUI creat in C++ putem sa-l convertim pentru limbajul Python cu ajutorul unei comenzi in consola, fiind un foarte mare plus pentru programatorii ce utilizeaza aceasta librerie.



Video prezentare



Mulumesc pentru atentie!