Dsa Group Project

Group Memebers

Enata Shikongo- 222098147

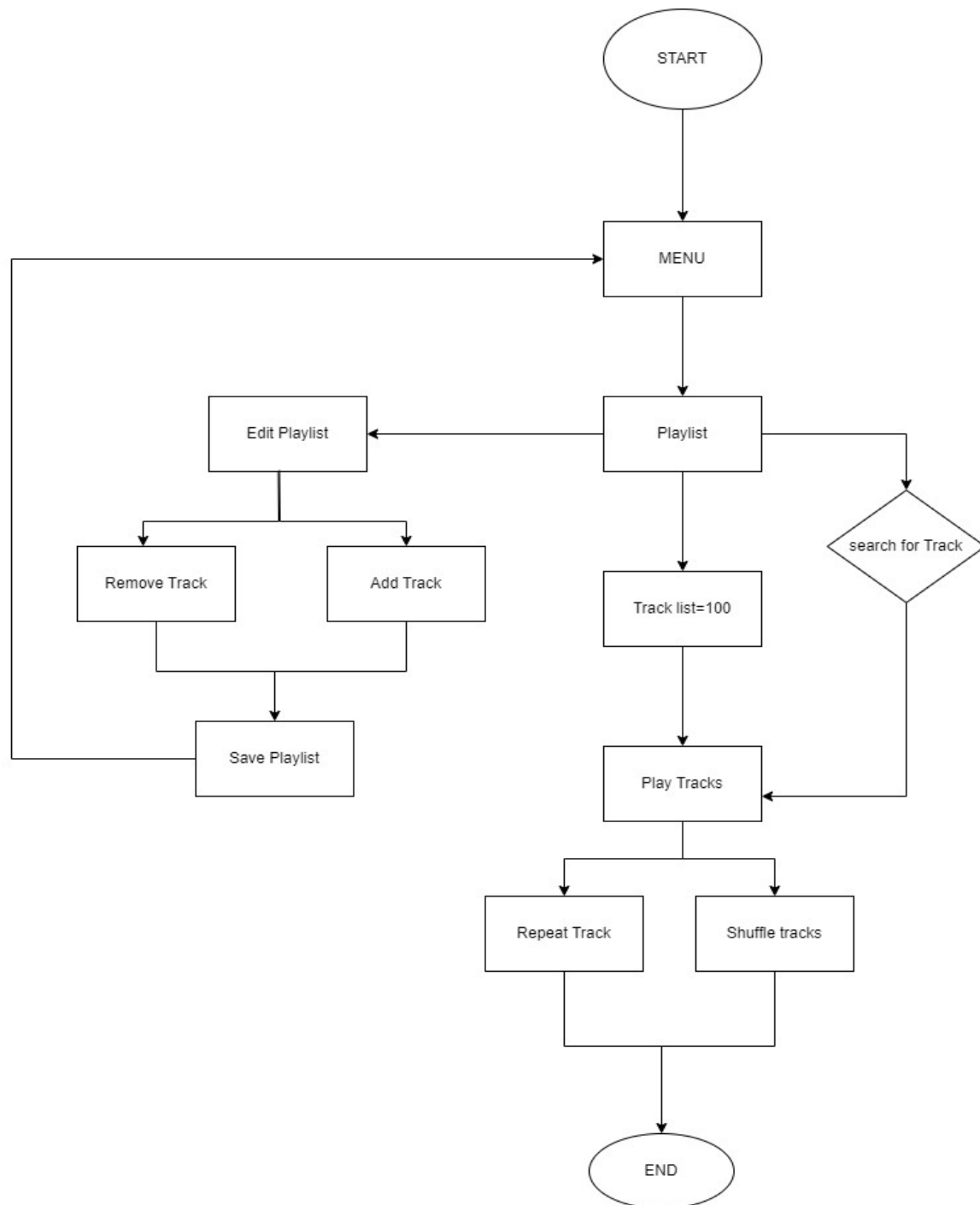**Job M Fuuma -222005017**

Nguvitjita Zemburuka -222084731

Johannes Petrus -222064501

Jones Neville- 221085432

SECTION A:

Flow chart

```
                              ┌──────────┐
                              │  START   │
                              └────┬─────┘
                                   │
                                   ▼
                              ┌──────────┐
              ┌──────────────▶│  MENU    │
              │               └────┬─────┘
              │                    │
              │                    ▼
    ┌──────────────┐          ┌──────────┐              ◇
    │ Edit Playlist │◀────────│ Playlist │──────▶ search for Track
    └──────┬────────┘          └────┬─────┘              │
           │                        │                    │
      ┌────┴────┐                   ▼                    │
      ▼         ▼             ┌──────────────┐           │
┌──────────┐ ┌──────────┐     │ Track list=100│          │
│Remove    │ │Add Track │     └──────┬───────┘           │
│Track     │ │          │            │                   │
└────┬─────┘ └────┬─────┘            ▼                    ▼
     └─────┬──────┘           ┌──────────┐
           ▼                  │Play Tracks│◀─────────────┘
    ┌──────────────┐          └────┬─────┘
    │ Save Playlist│           ┌───┴────┐
    └──────────────┘           ▼        ▼
                       ┌──────────┐ ┌──────────┐
                       │Repeat    │ │Shuffle   │
                       │Track     │ │tracks    │
                       └────┬─────┘ └────┬─────┘
                            └─────┬──────┘
                                  ▼
                             ┌──────────┐
                             │   END    │
                             └──────────┘
```

A

The Algorithms representation of the different modules and/functions (Pseudocode)

2 Step By Step

Step 1

pseudo code below;

 initialize music player

initialize playlist

 initialize track list

 initialize repeat function

initialize search function

initialize add/remove

play music player

play tracks from playlist

 repeat tracks as necessary

search for specific tracks

 add/remove tracks as necessary

 exit

end

Explanation of the pseudo code

Step 2

```
 function create_playlist()

tracks := 100

 linked_list := create_new_linked_list()

 for i := 0 to tracks-1 do

 add_track(linked_list, i)

end for

 return linked_list

end function

 function play_track(linked_list, track_number)

 current_track := linked_list.head

 for i := 0 to track_number-1 do

 current_track := current_track.next

end for

play_audio(current_track.data)

end function

function add_track(linked_list, track_number)

new_track := create_new_track(track_number)

if linked_list.head = null then

linked_list.head := new_track

else

current_track := linked_list.head
```

```
    while current_track.next != null do

     current_track := current_track.next

     end while

     current_track.next := new_track

     end if

     end function

     function remove_track(linked_list, track_number)

     current_track := linked_list.head

     previous_track := null

     for i := 0 to track_number-1 do

      previous_track := current_track

    current_track := current_track.next

     end for

     if previous_track = null then

     linked_list.head := current_track.next

     else

    previous_track.next := current_track.next

     end if

     end function

     function search_playlist(linked_list, track_number)

     current_track := linked_list.head

     while current_track != null do
```

```
    if current_track.data = track_number then

return true

    end if

current_track := current_track.next

end while

    return false

    end function
```

 The music player, playlist, track list, repeat feature, search feature, and add/remove feature are all initialised by the code above. Then, songs from the playlist are played on the music player. Searches for specific songs may be done, and the tracks are replayed as necessary. The last option is to add or delete tracks as needed.

SECTION B:

```
public class musicplayer {
public static void main(String[] args) {

}
public final class DoubleClass<Music> {

private Music data;
private DoubleClass<Music> next;
private DoubleClass<Music> prev;
public DoubleClass(final DoubleClass<Music> prev, final Music data, final
DoubleClass<Music> next) {
this.data = data;
this.next = next;
this.prev = prev;
}


public DoubleClass(final Music data) {
this(null, data, null);
}

public Music getData() {
return data;
}

public DoubleClass<Music> getNext() {
return next;
}


public void setTheNextTrack(final DoubleClass<Music> next) {
this.next = next;
}

public void setThePreviousTrack(final DoubleClass<Music> prev) {
this.prev = prev;
}

public DoubleClass<Music> getPrev() {
return prev;
}
```

```
public void addTracks(final Music data) {
this.data = data;
```