|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Practicum Final Exam – Odd Semester Year 2021/2022** | | | | | | |
| **Subject** | | | **COMP6153001 – Operating System** | | |  |
| **Class** | **:** | **BA03 / BA08 / BB03** | | **Start Date** | **: 12 January 2022** |
| **Lecturer** | **:** | **D2518 - Rony Baskoro Lukito, S.Kom., M.Kom**  **D3778 - Benfano Soewito, M.Sc., Ph.D** | | **Start Time** | **: 15:20 WIB** |
| **End Date** | **: 12 January 2022** |
| **End Time** | **: 17:20 WIB** |

**PERATURAN UJIAN:**

*Exam Regulations:*

* Mahasiswa tidak diperbolehkan berdiskusi dan/atau bekerja sama dengan peserta ujian lainnya

*Student is not allowed to discuss and/or work together with other exam participants*

* Mahasiswa tidak diperbolehkan untuk membuka dan menyalin dari **BUKU** atau **CATATAN**, **VIDEO** dari pengajar (recording kelas, VBL, Youtube, dsb) dan **REFERENSI** lainnya

*Student isn't allowed to open and copy from any resources such as notes, videos (class recording, VBL, Youtube, etc) and other references*

* Mahasiswa tidak diperbolehkan membuka dan menyalin jawaban dari internet (google, stackoverflow, dsb)

*Student isn't allowed to open and copy answer from the internet (google, stackoverflow, etc)*

* Asisten **BERHAK** memberi nilai 0 **(NOL)** bagi peserta ujian yang melakukan segala bentuk kecurangan

*Assistant is able to give 0 (ZERO) score for exam participant who does any cheating actions*

* Kumpulkan jawaban tepat pada waktunya, apabila terlambat mengumpulkan maka jawaban tidak akan dikoreksi dan nilai mahasiswa adalah 0

*Submit the answer on time, if not, then the answer will not be checked, and the students will receive 0 (ZERO)*

* Bila Anda tidak membaca peraturan ini, maka Anda dianggap telah membaca dan menyetujuinya

*If you have missed to read these regulations, so you are considered to have read and agreed on it*



**SOFTWARE YANG DIGUNAKAN:**

*Software will be used:*

* Java 8
* Eclipse 2020.6
* NachOS 5.0j

**FILE YANG DIKUMPULKAN:**

*File must be collected:*

* JAVA
* CLASS

**PERHATIAN!**

*Attention!*

* Bagi yang mengerjakan tidak sesuai dengan soal, maka akan diberikan nilai **NOL (0)**

*For those who do not work in accordance with the exam case will be marked as* ***ZERO (0)***

* Bagi yang mengerjakan tidak sesuai dengan software dan versi yang telah ditetapkan, maka akan tetap dikoreksi dengan software dan versi yang telah ditetapkan

*For those who do not work in accordance with the software and specific version will be corrected by the predefined software and version*

* Kompres semua jawaban yang akan diunggah. Pastikan format pengumpulan nama file dan ekstensi sesuai dengan format berikut: **[NIM]-[NAMA].zip**

*Compress all file that will be uploaded. Make sure the format for collecting file name and extension according to the following format:* ***[NIM]-[NAME].zip***

**Important Notes**:

1. Use your **own Console** and **Scheduler (FIFO)**.
2. Youmustuse **Semaphore** to **control** operation.
3. You must use **Timer** to generate tick of time.
4. You must use **File System** to save the data.
5. Use **KThread**.

**Soal**

*Case*

**spoJiFy**

**spoJiFy** is a platform that connects artists and audiophiles. It is a place for everyone to write and publish their own music. Not only that, spoJiFy also provides a streaming service for their published songs. As they are still new in this industry, their first goal is to create a dependable storage system using **nachOS**’ **file system** in **Java programming language**.

First, when the application start, the program will look for a file named **“songList.txt”** which will be **converted into an acceptable song list data** and load it into the system with the help of **file system** if the file is **available**.

After the process has been completed, the application will show 4 menus which represent the features in the application. Which are:

* + **Write Song**
  + **Listen All Songs**
  + **Delete Song**
  + **Exit**

A picture containing application

Description automatically generated

**Figure 1. Menu**

1. **Write Song (Menu 1)**

The first menu allows user to **insert a song into the list**. The inserting process follow the following steps:

* + - * When user choose this menu, the program will ask user to **insert song name**.



**Figure 2. Insert Song Name**

* + - * After that, the program will ask user to fill the song lyrics. **Validate** the song lyrics must be between **10 – 50 characters (Inclusive)**.



**Figure 3. Insert Song Lyrics**

* + - * Lastly, the user must input the song duration. The **song duration** must be between **1 – 360 seconds (Inclusive).**



**Figure 4. Insert Song Duration**

* + - * After the user provides all the necessary inputs, **store** the song information into a file named “**songList.txt**”. Please store the all the information using **“#”** **as a separator with the following format**.

**[song name]#[song duration]#[song lyrics]**

1. **Listen All Songs (Menu 2)**

The second menu allows user to **listen to all songs inside the list**. The listening process follow the following steps:

* + - * If there are **no songs stored** inside the song list**, show error message**.
      * Otherwise, the program will play all the songs stored in it using **KThread** and **Scheduler (FIFO Concept).** For every song, display the **song name**, **lyrics**, **for x minutes** **(obtained from the song duration converted to application minutes)**. The song will be shown with a delay of **1000 milliseconds**, **whereas 1000 milliseconds** **represent a single application minute**.

Background pattern

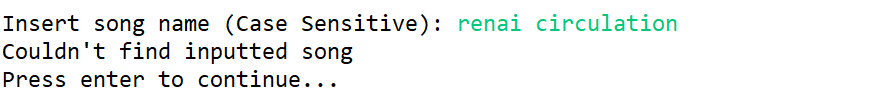
Description automatically generated with low confidence

**Figure 5. Listen All Songs**

**3. Delete Song (Menu 3)**

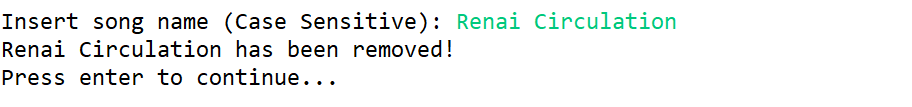
The third menu allows the user to **delete a song from the list**. The deleting process follow the following steps:

* + - * If there are **no songs stored** inside the song list**, show error message**.
      * Otherwise, the program will ask user to **input the song name** to delete **(Case Sensitive)**.
      * Validate if the inputted song exists if the inputted **song doesn’t exist** **show error message**.



**Figure 6. Input Song Name (Delete)**

* + - * However, if the **inputted song exists**, **remove** the song from the **file system** and **show success message**.



**Figure 7. Success Message**

* + - * After the song deletion has been **completed successfully**. **Update the data** inside **“songList.txt”** and **redirect the user** to the **main menu.**

**4. Exit (Menu 4)**

* + - * When user choose this menu, the program **will show the time** in **real time minutes passed** using a **timer**. Minutes could be calculated from the following formula:

|  |
| --- |
| **1 Minute = 60 Seconds**  **1 Second = 10.000.000 Timer Ticks** |

**Background pattern

Description automatically generated with low confidence**

**Figure 8. Exit Message**

**Must be collected:**

1. Java Project (Including NachOS and student’s code) compressed (.zip)