|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Practicum Final Exam – Odd Semester Year 2021/2022** | | | | | | |
| **Subject** | | | **COMP6153001 – Operating System** | | |  |
| **Class** | **:** | **B601 / B701 / BC07** | | **Start Date** | **: 18 January 2022** |
| **Lecturer** | **:** | **D2518 - Rony Baskoro Lukito, S.Kom., M.Kom**  **D5214 - Martin Suhartana, ST., MM., M.TI.** | | **Start Time** | **: 09:20 WIB** |
| **End Date** | **: 18 January 2022** |
| **End Time** | **: 11:20 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 **Network Link** to send and receive the data.
5. Use **KThread**.

**Soal**

*Case*

**e-Wallet**

**e-Wallet** is an application to transfer money and view history transaction for the users. This application is inspired by cashless wallet that was used in smart phones. As a programmer, you are asked to create an application that does the required features for **e-Wallet** using **nachOS in Java Programming Language**.

At the start of application, the program **will be shown with delay 3000 milliseconds** (use **Thread sleep**). The program will **give the user a starting balance of 20.000** and show the main menu where the application will show the **user’s balance** and **address** (based on the network link address) along with **4 menu options** which are:

* **Transfer Money**
* **History Transaction**
* **Top Up**
* **Exit**

Text

Description automatically generated

**Figure 1. Main menu**

1. **Transfer Money**

The first option of the menu allows the user to send a money into another user. This requires the user to fill in a few inputs along with its validation which are listed below:

* **Validate that the user’s balance** must be **more than or equals to 10.000**. If the user’s balance is less than 10000 then show an error message and redirect the user back to the main menu**.**

A screenshot of a computer

Description automatically generated with low confidence

**Figure 2. Low balance message**

* Otherwise, ask the user to **input the destination e-Wallet address.**
* Ask the user to **input money**. The money must be **between 10.000 and current balance (Inclusive).**
* Ask user to **input description**. The description length must be **between 5 and 100 chars (Inclusive).**

Text

Description automatically generated

**Figure 3. Transfer money**

* After fulfilling all validations, the application will **send** the money to receiver, then save the transaction with the **type ‘Send’ for the sender** **and** save the transaction with type **‘Receive’ for the receiver** and **update balance of both users**. The transaction will be sent with the following format:

|  |
| --- |
| **[Transaction’s Sender]#[Transaction’s Receiver]#[Transaction’s Description]#[Transaction’s Money]** |

* Then, a notification will appear in the receiver’s main menu.

A picture containing company name

Description automatically generated

**Figure 4. Notification**

1. **Transaction History**

The second option **Transaction History** allows the user to view transaction history that was sent / received from another user. Every view printing process involves a process with the following criteria:

* **Validate** if the current user **has any transaction**.If the user **has no transaction**, then the application will **show an error message**.

Chart, scatter chart

Description automatically generated with medium confidence

**Figure 5. Empty transaction history**

* Otherwise, if the current user **has any transactions,** then the application will **show all transactions** using **KThread and Scheduler (FIFO concept)** with the following criteria:
  + **Type, print either ‘Receive’ or ‘Send’ based on the transaction type.**
  + **From, print the sender’s address if the transaction type is ‘Receive’.**
  + **To, print the receiver’s address if the transaction type is ‘Send’.**
  + **Money, print the money that was sent or received on the transaction.**
  + **Description, print the description that was created by the sender.**

A picture containing graphical user interface

Description automatically generated

**Figure 6. History transaction for “Send” type**

A picture containing graphical user interface

Description automatically generated

**Figure 7. History transaction for “Receive” type**

1. **Top Up**

The third menu is a feature that allows the user to top up money. Process that will run whenever a user chooses this option are:

* Ask the user **to input money and validate that the** **money must be more than or equals to 10.000**.

Chart, scatter chart

Description automatically generated

**Figure 8. Top up process.**

* After fulfilling all validations, the application will **update the user’s balance**, then save the transaction with the **type ‘Receive’ with the sender’s address taken from the user own address and description will be filled with ‘Top Up’** and **redirect the user back to the main menu.**

A picture containing background pattern

Description automatically generated

**Figure 9. History transaction for top up.**

1. **Exit**

If the user chooses this menu, the system **will print ticks of time passed using Timer**. After that, the system will exit.

Text

Description automatically generated

**Figure 10. Exit**

**Must be collected:**

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