Logging Document

CECS 491
TheBoiZ

Lead:	014073642	Fa Fu
Members:	015188925	Brian Nguyen
	013122991	Louie Yonzon
	015774900	Kevin Simon
	015938089	Dylan Thorin
	015260958	Kevin Phan
Date:	November 7, 2019	

Logging

Functional Requirements:

- Definition -

At every feature operation (successful/unsuccessful), a log is made of that event by the system.

- Preconditions -

A feature operation occurs on the web application (excluding logging and archiving).

- Required Inputs -

- Type of Operation Performed
- Error Message ("" if no error occurred)
- User/System Unique Identification

- Successful Postconditions -

- Logs are saved separate from application information.
- A log is comprised of:
  - Timestamp (Date: YYYY-MM-DD, Time: 00:00:00:00 UTC)
  - Type of Operation Performed
  - The Error Message (if applicable)
  - The User/System Unique Identification
- All logs are saved within the file: logs.csv
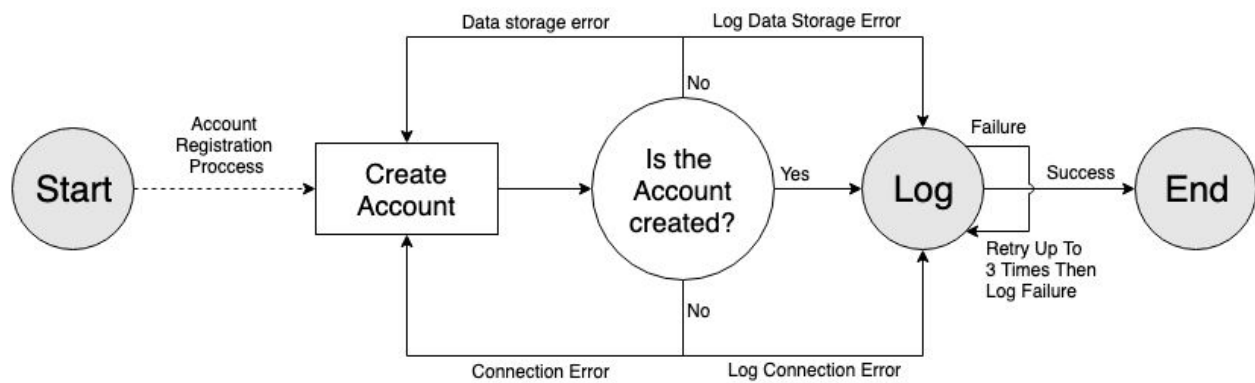
- Failure Postconditions -

- Any of the successful postconditions failing.
- Failure to log.
  - Attempts to log a maximum of 3 times, then notifies the system administrator. System prevents further logging of that specific entry.
  - This operation is then logged.

Non-Functional Requirements:

- Logs are sorted by Timestamp.
- Log Timestamps should be accurate.
- The logging process should take a maximum of 3 seconds.

Logging Examples:

Account Registration Error Logging:

Logging Low Level:

:Event   :Logging   :File

new Logging(string doc, int runs)

constructor

return instance

bool Logging.Log(op, msg, id)

string log

log = time + op + msg + id

for int runs try:

File.AppendAllText(doc, log)

write

false

exception

true

void