**Worksheet 3: Non-Functional Requirements**

Thi Van Anh DUONG Student ID: 90023112

Diploma of Information Technology, Curtin College

ISEN1000: Introduction to Software Engineering

Coordinator: Khurram Hameed

14 March 2022

## **Student Declaration of Originality**

|  |  |
| --- | --- |
|  | This assignment is my own original work, and no part has been copied from another student’s work or source, except where it is clearly attributed. |
|  | All facts/ideas/claims are from academic sources are paraphrased and cited/referenced correctly. |
|  | I have not previously submitted this work in any form for THIS or for any other unit; or published it elsewhere before. |
|  | No part of this work has been written for me by another person. |
|  | I recognise that should this declaration be found to be false, disciplinary action could be taken and penalties imposed in accordance with Curtin College policy. |

**Electronic Submission:**

|  |  |
| --- | --- |
|  | I accept that it is my responsibility to check that the submitted file is the correct file, is readable and has not been corrupted. |
|  | I acknowledge that a submitted file that is unable to be read cannot be marked and will be treated as a non-submission. |
|  | I hold a copy of this work if the original is damaged, and will retain a copy of the Turnitin receipt as evidence of submission. |

1. **Verifying Non-Functional Requirements (NFRs)**
2. NFRs: valid

*Testing*: With loading, we can use different formats to store the file (PNG, JPG, GIF) and check whether or not the system can recognize or display it in the screen. With saving: trying to save file in different formats, when file is saved uploading them to another website or program (Email, Microsoft Word, so on) to check the format is correct or not.

1. Functional Requirements

*Flow of event:*

1. The customer upload screenshots
2. The system scan information of screenshots
3. The system saves it.
4. The system scan nearby camera devices via Bluetooth or Wifi
5. The system connects with camera.
6. NFRs: valid

Testing: Use different processors and CPUs, downloading an image from the Internet and check how long does it take to save the image.

1. NFRs: valid

Testing: Users can compare and test the image capacity every single time they upload images to system.

1. NFRs: invalid

Because some PC can waste longer time to process it and it also depends on what type of hardware involved.

1. NFRs: valid

Test: conducting an experiment to perform the full range of image manipulation functions based on standard criteria and the calculate the time which the system consumes and the number of crashed times.

1. Functional Requirements

When you recover an image, it means that the image had to saved previously. Furthermore, recover the image means you have to save it again and saving is the functional requirement for this system.

*Flow of event:*

1. The system check “Trash” and find the image that user wants to recover.
2. The system askes the user “Do you want to recover it?”
3. The user click “Confirm” option.
4. The system recovers the image.
5. Functional Requirements

It means that the system has to resize the image in order to suitable with photo album and resizing is the functional requirement for this system.

1. NFRs: valid

Test: we can rotate the number of images, then we test the result (within 3 degrees of level or not) and compare it with the time we took.

1. NFRs: valid

Test: we can try to open number of image and compare the time.

1. **Performance Requirements**
2. *A software system that enables a car to drive itself.*
   * **Response time**: response to accidents, time set (start driving and stop driving)
   * Data processing: recognize traffic signs, traffic jams and check the obstacles, hindrances in the roads.

* *This is the most useful and valid performance requirement.*
  + How efficient its use of resources is? Bluetooth & Wifi (for calling, send message, listen music and radios); GPS: locate the current position, necessary to use Google Map.

1. *A text-to-speech system that can read document out loud.*
   * Response time: time taken to read the document.
   * **Data processing**: the system must read up to 200 words per minute, the safe security system to avoid hacker (in some bank, they use text-to-speech system to provide OTP for customers)

* *This is the most useful and valid performance requirement.*
  + How efficient its use of resources is? Internet connection (Wifi, 3G, 4G, 5G..), speaker (read the document) and processor (to scan and recognize the document and images in it).

1. *An equation-proof system that tries to automatically prove that two sides of a mathematical equation are equal.*
   * **Response time**: time to calculate the equation and compare the result.

* *This is the most useful and valid performance requirement.*
  + Data processing: the comparation have to be done at lease 10 seconds.
  + How efficient its use of resources is? Calculator menu (make sure that we choose the appropriate equation, ex: logical equation).

1. **Usability Requirements**
2. *A software system that enables a car to drive itself.*

* The software must allow users control the car’s speed.
* The software must allow users maintain the safe distance with other vehicles and traffic participants.
* The software must allow users choose the shortest way to their desired destination.

1. *An online banking system.*

* The system must allow customers check their account balance.
* The system must allow customers transfer money to another account.
* The system must allow customers check their transaction history.

1. *A voice-activated “smart home” system (for remotely/ automatically controlling various devices around the house).*

* The system must adopt users’ different accents.
* The system must allow users control other connected smart devices in the house.
* The system must automatically recognize visitors to the house.

1. **Reliability Requirements**
2. *A software system that enables a car to drive itself.*

* Mean Time To/ Between Failure (MTTF/MTBF): The system must have a MTTF of at least 2 years.
* Availability (AVAIL): The system should be available 99,999% of the time.
* Probability of Failure on Demand (POFOD): The system’s auto drive feature should have a POFOD of 0.1% or less.
* Rate of Occurrence of Failure (ROCOF): The system must not make more than 1 accident in every 1000 hours.

1. *A face recognition system for spotting wanted suspects in a crowded public place.*
   * + Mean Time To/ Between Failure (MTTF/MTBF): The system must have a MTTF of a at least 6 months.
     + Availability (AVAIL): The system should be available 99,999% of the time.
     + Probability of Failure on Demand (POFOD): The system’s face recognition feature should have a POFOD of 0.1% or less.
     + Rate of Occurrence of Failure (ROCOF): The system must not recognize more than 1 incorrect face for every 1000 people.
2. *A software system for monitoring weather data to forecast cyclones and other storms.*

* Mean Time To/ Between Failure (MTTF/MTBF): The system must have a MTTF of at least 1 months.
* Availability (AVAIL): The system should be available 99,9% of the time.
* Probability of Failure on Demand (POFOD): The system’s weather monitoring data feature should have a POFOD of 0.1% or less.
* Rate of Occurrence of Failure (ROCOF): The system must not report more than 1 incorrect forecast in every 1000 hours.