SOFTWARE REQUIREMENT SPECIFICATIONS

FOR

VINAYA (VERSION: 0.0.1)

A FREE EDUCATIONAL SOFTWARE.

Table of Contents

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1.0 ABSTRACT

In this project, a free, educational software is proposed to demonstrate and verify basic Operating System Concepts such as Process Scheduling, Process Synchronization, Deadlock and Page Replacement Algorithms along with designing a Website where the software and its documentation is available to be downloaded by the general public.

This software aims to transform the theory of above topics into programmable code so as to graphically represent the concepts. As well as test and compare the results for each simulation. The software provides graphical aid to the user to perform- scheduling of processes to view turnaround time, waiting time and Gantt chart (FCFS, SJF, RR, Priority), simulation of Producer and Consumer Problem to see use of buffer and synchronization, testing on a process sequence for deadlock (Banker’s Algorithm) and implementation of FIFO, LRU and Optimal page replacement algorithm on a page sequence. The website will be the medium of access to the software; it is dynamic and also provides information on how to use the software. Users can provide feedback and updates and bug fixes can be released on the website.

The software is coded in Java and Java Swing while the website is coded in HTML, CSS, JavaScript and Bootstrap (Possible inclusion: Angular JS, Wordpress). The UI design is based on Microsoft’s Fluent and Metro UI Design languages. Web domain and hosting is done by help of AWS.

The targeted audiences for this software are students and teachers, who have basic understanding of Operating System concepts, for practical implementation of topics. The benefit of using this software is that it will provide understanding of the theory behind a topic, proper visualization, use of code for simulation and comparison of found results. Like a calculator provides correct answer for any calculation, this software may serve as digital truth for every simulation by the user.

The results for all simulations are viewed in separate java frames with interactive UI. Data provided during simulation may be stored as files in local system for future reference and comparison.

The software strives for educational visualization i.e. to help learners to practically implement the concepts as it is used inside the Operating System.

2.0 INTRODUCTION

This Software Requirements Specification specifies the requirements of Vinaya (Version 0.0.1), a free, GUI based, educational software that can be downloaded from its official deployment website using which users can simulate various Operating System concepts such as:-

* Process Creation
* Process Scheduling (FCFS, SJF, RR, Priority)
* Process Synchronization (Producer Consumer Problem )
* Deadlock Avoidance (Banker’s Algorithm)
* Page Replacement Algorithms (FIFO, LRU, Optimal)

While the deployment website shall include:-

* Launch / Home Page
* Software download Page
* Documentation and User Manual Page
* User Forum and Feedback Page
* Version Update and Release Page
* Credits / About Page

2.1 PURPOSE

The purpose of this Software Requirements Specification is to verify that all the specifications are verified. This document also serves to ensure that the software is traceable throughout its software development life cycle and that it adheres to the Scrum framework under Agile Methodology.

2.2 INTENDED AUDIENCE

This SRS would be used by the following people:-

* Product Owner:
* Scrum Master:
* Developers: The developers would use this document to implement the functionalities and to ensure traceability of the software.
* Testers: The testers would use this document to know the interfaces and to test the software accordingly.
* Users: The users would use this document to verify if the requirements specified satisfy their needs.

2.3 SCOPE OF THE PROJECT

The purpose of this proposed educational software is to allow users to demonstrate and verify earlier mentioned Operating System concepts through simulation. The GUI of this software shall allow ease of access to the user and help understand the working of algorithms behind these topics.

The software promotes:

* Education – via demonstration of concepts and verification of OS problems.
* Student Research – via comparing functionalities of various algorithms on a particular data set.

The deliverables of the software should be a document/text/image file of conducted simulation. This file may be used in another simulation or be saved for future reference.

This project has a lot of scope for future development. Features such as Pre-emptive Scheduling, Classical Synchronization Paradigms, RAG for Deadlock and many more such functionalities can be implemented in the later versions of this software.

2.4 OVERVIEW OF THE DOCUMENT

The first section of the document gave a brief description about Vinaya and the various benefits that it provides. In the following sections we will describe the requirements, assumptions, dependencies, constraints and other such concepts about the software.

3.0 OVERALL DESCRIPTION

3.1 PRODUCT PERSPECTIVE

Vinaya can be downloaded from its official deployment website and can be used to demonstrate and verify earlier mentioned basic Operating System concepts. The simulation may be saved as document, text or image file for future reference. These files may later be used as comparison between different primitive algorithms as part of student research.

3.2 PRODUCT FEATURES

The project showcases simulation of mentioned OS concepts using features such as:

* An interactive, dynamic, deployment website to allow users to:
  + download the software and user manual
  + check update and release log
  + give feedback and discuss common issues in a forum
* Accept user defined processes and store it for future use in a database.
* Schedule a set of processes to view Gantt chart or scheduling details.
* Simulate synchronization in between Producer and Consumer.
* Input Resource type, instances, processes along with Allocation, Available and Max matrix to implement Banker’s Algorithm and deem system safe or unsafe.
* Input a page sequence and implement FIFO, LRU, Optimal to simulate page replacement and show page faults, page hit, efficiency ratios.

3.3 USER CLASSES AND CHARACTERISTICS

There will be primarily two types of targeted audience for this software:

* Teachers: To demonstrate working of various basic concepts inside an OS and implementation of primitive algorithms behind topics.
* Students: To verify the results obtained in each simulation; To compare between simulation results and document findings.

3.4.1 OPERATING ENVIRONMENT

3.4.2 HARDWARE REQUIREMENTS

3.4.3 SOFTWARE REQUIREMENTS

3.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

* For ensuring platform independence of the software the implementation will be JAVA so the end users system must have a JAVA run time environment.
* The software may schedule at most 10 processes.
* FCFS, SJF and Priority shall be Non-Pre-emptive scheduling Algorithms.
* For process synchronization, size of producer, consumer and buffer shall be at most 10 items.
* In case of Banker’s Algorithm, storage of at most 10 resource types and 10 processes.
* For page replacement, page sequence of 15 pages, frame size of 5 and Virtual Memory size of 20 shall be set.

3.6 USER DOCUMENTATION

* A user document should be provided at the end of the development. It should have the following:
* A ReadMe file to help the user with the installation of the software.
* A well-documented User Manual.

3.7 ASSUMPTIONS AND DEPENDENCIES

ASSUMPTIONS:

* User already has basic idea about the concepts of OS being demonstrated in this software.
* User has experience of working with simulators.
* User knows how to use a software manual to run the software.

DEPENDENCIES:

* This software would need Java Runtime Environment for its execution. (Possible inclusion : Apache Derby and Glass Fish Server for Backend Support)
* It will need a web browser and an active Internet connection to download the software and user manual.
* It will require an approximate storage space of 50Mb.

3.8 DATA REQUIREMENTS

INPUT

OUTPUT

3.9 USER VIEW OF THE SOFTWARE

This software will have Graphical User Interface that will provide ease of access. The software will via frames/windows. The user can perform simulation by simply entering input into the text fields and clicking a label or button to begin the operation.

Possible types of Screens:

* Launch/Landing Frame
* Home Frame
* Instructions Frame
* Credits Frame
* Concept Window (For every Module)
* Simulation Window
* Dialog Boxes (Confirmation/ Warning)
* Exit Page

3.10 EXTERNAL INTERFACE REQUIREMENT

Hardware Peripherals: Computer System

Keyboard, Mouse (Preferred)

3.10.1 HARDWARE INTERFACE

* Vinaya depends on User Input to perform simulations. Hence a keyboard and mouse is preferred for proper functioning.
* Screen resolution of at least 800 X 600 or above will be preferable for viewing the frames / windows.

3.10.2 SOFTWARE INTERFACE

* It should be possible for Vinaya to be implemented across all platforms and Operating System Environments.
* The entire software is to be coded in Java and the GUI is to be designed using Java Swing. It should embed certain useful hyperlinks to its official website.
* The software and user manual will need a web browser for downloading it.

4.0 SYSTEM FUNCTIONALITIES

4.1 PROVISION OF BASIC INFORMATION REGARDING CONCEPTS

4.2 PROCESS CREATION

4.3 PROCESS SCHEDULING

4.4 PROCESS SYNCHRONIZATION

4.5 DEADLOCK AVOIDANCE

4.6 PAGE REPLACEMENT

4.7 SAVING A SIMULATION

5.0 DATA FLOW DIAGRAM

6.0 NON FUNCTIONAL REQUIREMENTS

6.1 PERFORMANCE RQUIREMENTS

This software should perform ­the same way irrespective to its Operating System environments. Time taken for importing previously saved files and performing simulation should be negligible. The software should suffer from minimum to no crashes. The software must provide accurate calculation of results and proper visualization.

6.2 SAFETY REQUIREMENTS

The software available online for downloading should be free from any malware and should pose no threat to the system.

6.3 SECURITY REQUIREMENTS

As all the operations are to be done within a single system with no handling of personal information, security is not an issue for this software.

6.4 QUALITY REQUIREMENTS

Quality has a number of attributes. Some of the important attributes for this software are:

Portability: As this software is to work on multiple platforms, portability is an essential attribute and we ensure this by using JAVA as our programming language.

User Training: We assume that the user already has some previous experience in working with simulators. So the users will not need any specific training for using this software.

Testability: As a basic characteristic the software needs to be testable to ensure correctness. Manual testing with certain test cases and automated JUnit Testing shall be used.

7.0 WORKFLOW

8.0 SEQUENCE DIAGRAM

9.0 BIBLIOGRAPHY