Software Requirements Specification

for

Hostel Management System (custom)

**Version 1.0 (pending approval)**

Triple SSsolutions LLC.

Tel.: +92 324 2624354

Email: [hsm3.solutions@gmail.com](mailto:hsm3.solutions@gmail.com)

**Moatasim bin Hisham Sayyid, Founder**

**October 16th, 2022**

**Table of Contents**

**Table of Contents** [**ii**](#_heading=h.gjdgxs)

[**Revision History**](#_heading=h.gjdgxs) [**ii**](#_heading=h.30j0zll)

[**1. Introduction**](#_heading=h.30j0zll) [**1**](#_heading=h.1fob9te)

[1.1 Purpose](#_heading=h.1fob9te) [1](#_heading=h.3znysh7)

[1.2 Document Conventions](#_heading=h.3znysh7) [1](#_heading=h.2et92p0)

[1.3 Intended Audience and Reading Suggestions](#_heading=h.2et92p0) [1](#_heading=h.tyjcwt)

[1.4 Product Scope](#_heading=h.tyjcwt) [1](#_heading=h.3dy6vkm)

[1.5 References](#_heading=h.3dy6vkm) [1](#_heading=h.4d34og8)

[**2. Overall Description**](#_heading=h.4d34og8) [**2**](#_heading=h.2s8eyo1)

[2.1 Product Perspective](#_heading=h.2s8eyo1) [2](#_heading=h.17dp8vu)

[2.2 Product Functions](#_heading=h.17dp8vu) [2](#_heading=h.3rdcrjn)

[2.3 User Classes and Characteristics](#_heading=h.3rdcrjn) [2](#_heading=h.26in1rg)

[2.4 Operating Environment](#_heading=h.26in1rg) [2](#_heading=h.lnxbz9)

[2.5 Design and Implementation Constraints](#_heading=h.lnxbz9) [2](#_heading=h.35nkun2)

[2.6 User Documentation](#_heading=h.35nkun2) [2](#_heading=h.1ksv4uv)

[2.7 Assumptions and Dependencies](#_heading=h.1ksv4uv) [3](#_heading=h.44sinio)

[**3. External Interface Requirements**](#_heading=h.44sinio) [**3**](#_heading=h.2jxsxqh)

[3.1 User Interfaces](#_heading=h.2jxsxqh) [3](#_heading=h.z337ya)

[3.2 Hardware Interfaces](#_heading=h.z337ya) [3](#_heading=h.3j2qqm3)

[3.3 Software Interfaces](#_heading=h.3j2qqm3) [3](#_heading=h.1y810tw)

[3.4 Communications Interfaces](#_heading=h.1y810tw) [3](#_heading=h.4i7ojhp)

[**4. System Features**](#_heading=h.4i7ojhp) [**4**](#_heading=h.2xcytpi)

[4.1](#_heading=h.2xcytpi) Undo Feature [4](#_heading=h.1ci93xb)

[4.2](#_heading=h.1ci93xb) Redo Feature

4.3 History Feature

4.[4 Animation Feature.](#_heading=h.3whwml4)

[**5. Other Nonfunctional Requirements**](#_heading=h.3whwml4) [**4**](#_heading=h.2bn6wsx)

[5.1 Performance Requirements](#_heading=h.2bn6wsx) [4](#_heading=h.qsh70q)

[5.2 Safety Requirements](#_heading=h.qsh70q) [5](#_heading=h.3as4poj)

[5.3 Security Requirements](#_heading=h.3as4poj) [5](#_heading=h.1pxezwc)

[5.4 Software Quality Attributes](#_heading=h.1pxezwc) [5](#_heading=h.49x2ik5)

[5.5 Business Rules](#_heading=h.49x2ik5) [5](#_heading=h.2p2csry)

[**6. Other Requirements**](#_heading=h.2p2csry) [**5**](#_heading=h.147n2zr)

[**Appendix A: Glossary**](#_heading=h.147n2zr) [**5**](#_heading=h.3o7alnk)

[**Appendix B: Analysis Models**](#_heading=h.3o7alnk) [**5**](#_heading=h.23ckvvd)

[**Appendix C: To Be Determined List**](#_heading=h.23ckvvd) [**6**](#_heading=h.ihv636)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this document is to give a detailed description of the functional and non-functional requirements of the Custom Hostel Management software requested by our client Mr. Agha Sikandar Shah. Tentative name for the software is Hostel-Insight.

## Document Conventions

This document follows IEEE template for System Requirement Specification. The font used to write this document is Times New Roman. For main headings a bold font of size 18, for sub headings a bold font of size 14 and for description a normal font of size 12 has been used. Italicized text is used to label diagrams and to provide additional description to explain a particular point. Every requirement statement in the document has its own priority

## Intended Audience and Reading Suggestions

The intended audience of this documents are:

* The end users of Hostel Insight, which may include Hostel managers, Admin (Mr. Sikandar himself), and System maintainers (devs).
* Software testers and documentation writers.

Reading Suggestions:

* For an overall description of the project, see section 2 (mostly our own use).
* For an overview of the software interface see section 3 (for development guidelines/integrity standards)
* For an explanation of requirements and implementation of System features, see section 4.
* For an explanation of non-functional requirements, see section 5.
* For explanation of the vocabulary used in the document, see the glossary in Appendix A.

## Product Scope

Hostel Insight is meant for use by the client and the people working under him managing the operation of the hostels under the client’s name. These managers must be able to add expenses/liabilities used by their respective hostels, keep a record of the students living there and their individual security deposits and hostel fees

## References

* System requirements for installing python

[Python-installation-instructions-1.pdf (shahandanchor.com)](https://www.shahandanchor.com/home/wp-content/uploads/Python-installation-instructions-1.pdf):

# Overall Description

## Product Perspective

Pyaint is a digital software which provides a digital drawing experience to the end user. In real life, once we have erased anything by mistake, we cannot draw it again with the same perfection. Our software provides Undo and Redo functionality which gives a chance to the end user to correct his action. As we know, in practice, remembering what we have drawn in sequence is impossible. This software provides the solution to this problem. We can maintain and clear the history. This helps us with the new feature, showing animations, which provides the recording experience to the end user. He/She can experience their creativity by watching a sequential animation.

## Product Functions

The Perspective is to add the following functionalities in our digital drawing software:-

- History will have a checkbox for each action a user has performed, Therefore if a user wants to delete a particular action, it can delete that specific action from the history. It can also delete multiple actions from the history. Users can clear the whole history as well. If the user cleared the whole history, undo and redo functionality will be disabled.

- User can undo its action once or multiple times until it reaches its first action.

- Users can redo till its last action performed once or multiple times.

- Users can run the animation from the start/very first action or it can show the animation from a specific index against action from the history.

## User Classes and Characteristics

Features available for the user:-

* User can view all his actions using the history feature.
* Users will have an undo button which can be used once or multiple times to reverse the last action.
* User will have a redo button which can be used once or multiple times to revert the undoed action.
* User will have a option to show the animation from the beginning
* Users will have an option to show the animation from a particular index or action.

## Operating Environment

The Pyaint software will run on any computer that has a Windows operating system version 8 or above.

## Design and Implementation Constraints

**Minimum Hardware Requirements:**

* 2 GB RAM.
* Intel core i3 or above processor.

**Software Constraint:**

* All the objects on the screen, and user actions, will be stored in a linked list.
* The number of objects that can be saved in the linked list will be limited by the available RAM in the system.
* Software must be responsive to the user screen.
* Python language will be used to develop the features.
* Every feature must have a button in the software’s GUI.

## User Documentation:

* A pdf and a word file with a pictorial manual or guide will be provided which will show how certain features work separately.
* The user guide will contain all the features like undo, redo and clearing the history be it one action at a time or all the actions together.
* Users will also be guided about the system requirements via the user guide.
* The manual will also contain an email address for the online help if required, although a certain time frame will be required for the user to wait until he gets the help.

## Assumptions and Dependencies

* **Assumptions:**
  + The operating system of the PC must be Windows 7 or above, Linux: Ubuntu 16.04 or above
  + Intel core i3 or above processor.
  + Python is installed on the user’s computer.
  + Pygame library is installed on the user’s computer.
  + The PC must have at least 500KB of free storage.
  + Users should have basic knowledge of how to interact with the software GUI.
  + User understands the English Language.
* **Dependencies:**
  + **Pygame**
    - For the GUI of the software.

# External Interface Requirements

## User Interfaces

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

## Hardware Interfaces

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## Software Interfaces

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

## Communications Interfaces

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

# System Features

**Priority High:** The function won’t work without this requirement.

**Priority Medium:** The function will not work completely without this requirement.

**Priority Low:** The function will work without this requirement but won’t be as good.

## Undo an action

**4.1.1 Description and Priority**

In our Pyaint software we need to include the option for the user to undo his actions.

This feature is very important to make sure that if a user makes a mistake he can go back to his original action and preserve his work.

**4.1.2 Stimulus/Response Sequences**

When the user is using Pyaint he will have a button of undo in his taskbar. When he clicks on that button his last performed action will be reversed to the action that was previously performed before the last action and the changes will be displayed on the screen.

**STEP 1: Click on the undo button**

**STEP 2: The current action on the screen will be removed.**

**STEP 3: The previous action performed will be displayed on screen.**

**4.1.3 Functional Requirements**

**REQ-1:** The user must have performed at least one action before being able to use the undo feature. **Priority High** (The feature won’t work without this)

## Redo the action

**4.2.1 Description and Priority**

We are including a redo feature in the Pyaint software to make sure that if users accidentally undo something they can go back to their original action.

**4.2.2 Stimulus/Response Sequences**

The user will have the option to redo the actions they undid by using a button on their task bar. As soon as they click on the button the redo action will be performed and displayed on the screen.

**STEP 1: Click on the redo button**

**STEP 2: The current action on the screen will be removed**

**STEP 3: The action you performed and undid will now be displayed on the screen.**

**4.2.3 Functional Requirements**

**REQ 1:** The user must have performed any action on the Pyaint software. **Priority High**

**REQ 2:** The user must have performed the undo function on any action. **Priority High**

## History

**4.3.1 Description and Priority**

The history feature will allow the user to view all the actions he has performed on the Pyaint software from the start till the end, including all his undo and redo actions. All the actions performed by the user will be given an index and that way the history function will be able to store all the actions and show the users a detail of all the actions they have performed.

**4.3.2 Stimulus/Response Sequences**

On the task bar the user will have the button for viewing their history. When they click on the button a new window will pop up which will display a sequence of actions the user has performed.

**STEP 1: Click on the history button**

**STEP 2: A new window will open**

**STEP 3: The new window will have a list of actions performed from start to the last action performed.**

4.3.3 Functional Requirements

**REQ 1:** The user must have performed any action on the Pyaint software. **Priority High**

## Animation

4.4.1 Description and Priority

The animation feature will allow the user to view their actions to be performed from start to finish. The animation feature will make use of the history feature to find out all of the actions performed by the user and then make a video animation of performing all those actions. The animation feature will perform all the actions separately that the user performed and record it and that is what will be displayed.

4.4.2 Stimulus/Response Sequences

On the task bar the user will have the button for animation. When they click on the button a new window will pop up which will show a kind of slideshow animation of all the user actions performed on the software.

**STEP 1: Click on the animation button**

**STEP 2: A new window will open**

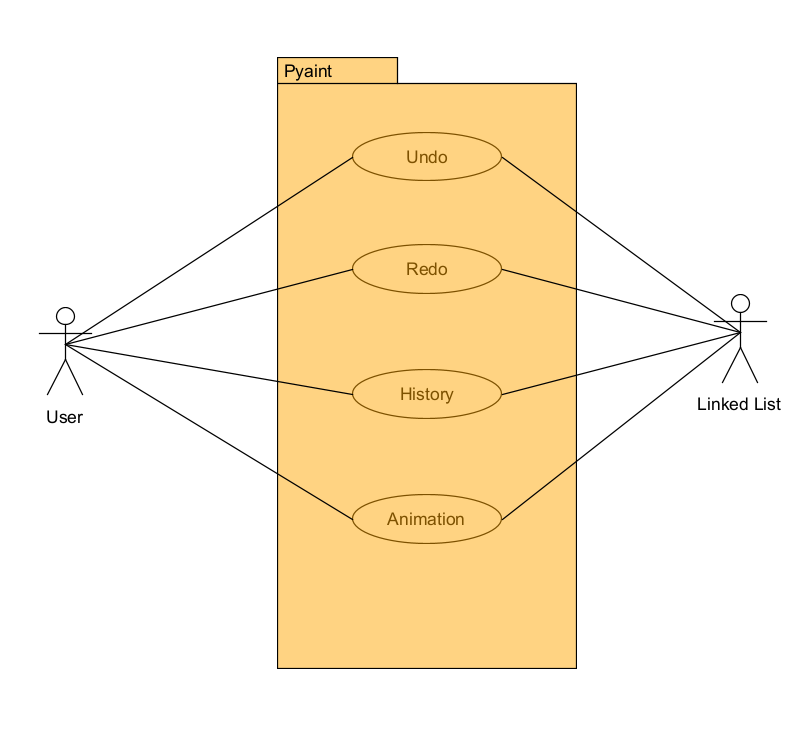
**STEP 3: The new window will display a timelapse video of performing all the user actions from start to the last action performed.**

4.4.3 Functional Requirements

**REQ 1:** The user must have performed any action on the Pyaint software. **Priority High**

**REQ 3:** The user performed actions must be documented in the history feature. **Priority High**

**USE CASE DIAGRAM:**



|  |  |
| --- | --- |
| **Use Case Name** | **Undo** |
| Related Requirements | **REQ 1** |
| Goal in Context | To be able to undo any action |
| Preconditions | 1. The user must have opened up the pyaint software 2. The user must have performed any action |
| Successful End Condition | The last performed action will be undone. |
| Failed End Condition | The last performed action will not be undone. |
| Primary Actors | User |
| Secondary Actors | None |
| Trigger | The user clicks on the undo button |
| Included Cases | None |
| Main Flow | |  |  | | --- | --- | | Step | Action | | 1 | The undo feature will begin when the user clicks on the undo button | | 2 | The function is called and it deletes the last action performed | | 3 | The action performed before the last action is now shown and is now the latest action performed | |

|  |  |
| --- | --- |
| **Use Case Name** | **Redo** |
| Related Requirements | **REQ 1** |
| Goal in Context | To be able to redo any action |
| Preconditions | 1. The user must have opened up the pyaint software 2. The user must have performed any action |
| Successful End Condition | The last performed action will be redo. |
| Failed End Condition | The last performed action will not be redo. |
| Primary Actors | User |
| Secondary Actors | None |
| Trigger | The user clicks on the redo button |
| Included Cases | None |
| Main Flow | |  |  | | --- | --- | | Step | Action | | 1 | The redo feature will begin when the user clicks on the undo button | | 2 | The function is called and it will go to the next action of the current action performed. | | 3 | The action performed next to the current action will be displayed. | |

|  |  |
| --- | --- |
| **Use Case Name** | **History** |
| Related Requirements | **REQ 1** |
| Goal in Context | To be able to view all the user action that led to the current state of the screen |
| Preconditions | 1. The user must have opened up the pyaint software 2. The user must have performed any action |
| Successful End Condition | A window opens with a list of all the actions done by the user in the order that the actions were performed. |
| Failed End Condition | History window does not open.  Incorrect order of action in the list.  List does not contain all the actions. |
| Primary Actors | User |
| Secondary Actors | None |
| Trigger | The user clicks on the history button |
| Included Cases | None |
| Main Flow | |  |  | | --- | --- | | Step | Action | | 1 | The history feature will begin when the user clicks on the history button | | 2 | The function is called and it fetches all the actions performed from the linked list. | | 3 | All the actions performed are now shown in the history window in the form of an ordered list. | |

|  |  |
| --- | --- |
| **Use Case Name** | **Animation** |
| Related Requirements | **REQ 4**  **Dependent requirements: REQ 1 & REQ 3** |
| Goal in Context | To render a real time animation of drawing sequences piling up from the starting (blank) screen, to the end drawing (last performed action). |
| Preconditions | 1. The user must have opened up the pyaint software 2. The user must have performed any action |
| Successful End Condition | A window opens and each action replays from start to finish, detailing how the drawing was made. Note: animation does not draw out the actions as made by the user, instead, each action performed from the start is displayed (like the redo function) one by one till the finished drawing. |
| Failed End Condition | The animation does not reach the final drawing. |
| Primary Actors | User |
| Secondary Actors | None |
| Trigger | The user clicks on the animation button |
| Included Cases | None |
| Main Flow | |  |  | | --- | --- | | Step | Action | | 1 | The animation feature will begin when the user clicks on the undo button | | 2 | The function is called and it fetches all the actions performed from the history | | 3 | A new window opens and it displays all the actions from start to finish | |

# Other Nonfunctional Requirements

## Performance Requirements

* The design and interface of the Undo/Redo/Animation tool must be minimal and highlight-able (when the cursor hovers over it).
* All three functions must have the least delay possible, with less than a few milliseconds of delay with Undo/Redo.
* In the case that there are more than 1000 sequences used to draw a particular image using Pyaint, the delay in rendering the Animation when the animation function is clicked, must not be greater than 5 seconds.
* Requires efficient memory management/caching.

## Safety Requirements

* Auto-save feature so as to ensure that after every action a user performs, the Undo/Redo/Animation feature can be clicked and used, and no loss of work occurs.

## Security Requirements

* Application does not require any specific user credentials or any other identity information that could be a potential security hazard.
* No online features so no chance of IP addresses being stolen/ no chance of phishing.
* Any drawings created by the product must be kept safe and secure at the user’s end.

## Software Quality Attributes

Since all three features are hard-coded into the application in the Python 3 language, the features will have interoperability, portability, maintainability and usability over different platforms, provided that Python 3 and Pygame are installed and running.

**Usability:**

Buttons clearly defined with visible size.

Work as expected.

**Portability:**

Every new user must have a copy of the source code.

Python 3 installed on their machine.

**Interoperability:**

Each function must perform the same on one machine as it would on another.

All features must work on any operating system with Python 3 installed.

**Maintainability:**

New versions pushed to GitHub.

Maintenance logs on GitHub.

## Business Rules

**GNU General Public License v3.0**Permissions of this strong copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights.

# Other Requirements

* The software will contain legal requirements about the consent of the information of the users that they allow the owner to use and store.
* Every information will be limited to the owner of this software and will not be shared.
* In the case of any unforeseen circumstance, the owners and the developers of this software will not be blamed for any miscariage of the personal information of the users that may have been due to any unethical access or hacking.

**Appendix A: Glossary**

* **Product Scope:** Future of the software.
* **History:** Past actions applied by the user.
* **Undo:** Reverting back to an action that the user feels is not right.
* **Redo**: Performing an action again which was redone by the user.
* **Assumptions:** Basic requirements of this software to run smoothly on any device..
* **Dependencies:** Software on which the features are dependent on.
* **Animation:** Slideshow of all the functions.
* **Usability:** The degree to which something is able or fit to be used.
* **Portability:** The ability to be easily carried or moved.
* **Interoperability:** The ability of computer systems or software to exchange and make use of information.
* **Maintainability:** The ability to maintain the services provided by the software and all providing all the relevant updates.
* **GNU General Public License v3.0:** Public license to make the source code available to the public.

**Appendix B: Analysis Models**

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

**Appendix C: To Be Determined List**

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*