3.46 Planning

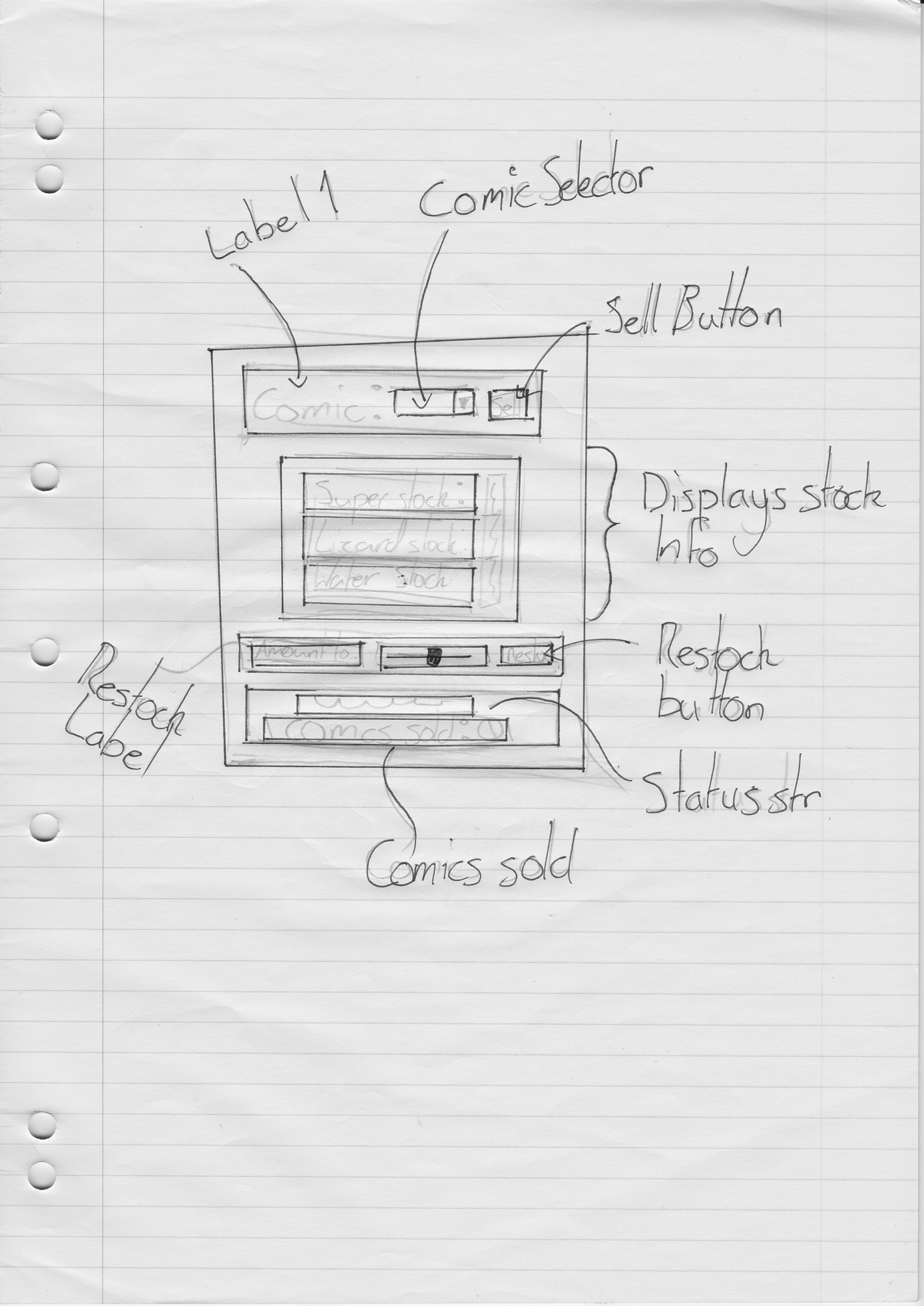
**User inputs:**

|  |  |  |
| --- | --- | --- |
| **Name:** | **Input Type:** | **Input Method (widget)** |
| Sell Button | Click (run function) | Button |
| Comic Selector | Option select (string) | Option box |
| Restock Button | Click (run function) | Button |
| Restock Slider | Integer | Slider |

Information Displays Needed:

|  |  |  |  |
| --- | --- | --- | --- |
| Name: | Information Type | What will it display? | Widget Used? |
| Water\_Label\_text | String/integer | How many water woman comics the user currently has in stock | Label |
| Lizard\_Label\_text | String/integer | How many lizard man comics the user currently has instock | Label |
| Super\_Label\_text | String/integer | How many Super Dude comics the user currently has in stock | Label |
| Status\_str | String | Wether or not the last action the user tried to do was successful. If not it will display an appropriate error message | Label |
| Restock\_label | String | “Amount to restock: “ | Label |
| Amount\_sold | Integer | The amount of comics sold | Label |
| Amount\_sold\_text | String | “amount of comics sold: | Label |
| Label1 | String | “comic: “ | Label |

**Design interface sketch:**



**Classes Required:**

Class: Comic

This class is used to make each of the comics, it stores things like the amount of stock and runs the sell and restock functions.

**Constraints or existing Data:**

The user will have to enter values into the data.txt file when they first run the program so it display’s their desired comics/stocks. After that the autosave features will allow the program what the user’s previous values were.

**Indexed Data Structures**

I have a dictionary (comics\_dict) that stores the name of the comic and its’ instance of the comic class.

**What calculations are nessescary?**

* How many comics have been sold: comics sold += 1 whenever sell button is clicked
* Can the user sell this comic?: check that the chosen comics stock is > 0
* Restock amount: chosen comic’s amount + restock slider value
* Sell a comic: chosen comic’s stock – 1

**Functions Used**:

Sell(): (note this is contained within the comic class)

Check that’s its stock > 0, if it is:

Minus 1 from its stock and update any labels

Set the status string to reflect the successful transaction

Update the comics sold value

Else:

Set the status string to notify the user of the failed attempt and the reason for the failure

Restock(): (note this is contained within the comic class)

Get the value from the restock slider

Add the restock slider value to the chosen comic’s stock

Update the chosen comics stock label

Update the status string to reflect the successful transactions

**Test cases**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variable being tested** | **Input** | **Expected Result** | **Result** | **Reason for test** | **Pass/Fail** |
| Stocks | Sell button when stock is at 0 | Error shown in status string | Error shown in status string | Too check that thee user cannot sell stock the do not have | Pass |
| Sell button when stock is not at 0 | Stock goes down by one for chosen comic, comics sold label is updated to reflect the new sale and the status string reflects the successful attempt | Stock goes down by one for chosen comic, comics sold label is updated to reflect the new sale and the status string reflects the successful attempt | To check the sell function works correctly | Pass |
| Comic Option box | Change the selected comic and try selling/ restocking | Updates the correct values for the correct comic | Updates the correct values for the correct comic and staus string reflects the changes | To check the option selector box return the correct values for the restock and sell functions. And to. Check that the status string updates on restock attempts | Pass |

**Images of testing:**

|  |  |
| --- | --- |
|  | Checking that the user cannot sell stock they do not have.  The stock did not change and the status string showed an easy to understand error message. |
|  | Checking that everything updates when when a comic is sold. the correct comics stock went down by one (default is 8)  The number of comics sold increased by the correct amount |
|  | Checking that when the selected comic changes, the correct values are updated.  It all works correctly |
|  | Check that the restock slider/button work correctly  The math was correct (12 Is default, 12 + 10 =22)  The correct labels including the status string updated. |