WEIZMANN INSTITUTE OF SCIENCE

Holon Institute of Technology

Programming for Everyone:
An Introduction to Visual Languages

Student Details:

Name: Niv Neuvirth

Department: Computer Science

Self-Checkout System

- Self-checkouts, are machines that provide a mechanism for customers to complete their own transaction from a retailer without needing a traditional staffed checkout. When using, customers scan item barcodes before paying for their total shop without needing one-to-one staff assistance. Self-checkouts are used mainly in supermarkets. Most self-checkout areas are supervised by at least one staff member, often assisting customers process transactions, correcting prices, or otherwise providing service.

Inputs & Outputs

- Inputs:
  - Mode selection - ‘Guest’ or ‘Member’
  - Item scan – when the user scans the items barcode
  - User pressing on different buttons such as ‘Pay’, ‘Help’, ‘Add item’ and more
  - Item weight – when the user scans or adds an item that needs to be weighed

- Outputs:
  - Indicator light mode and color changes accordingly to the input
  - Screen display – changes between ‘Welcome’ to ‘Main’ to ‘Help’ and more
  - Total price (‘Guest’ input), total member price and points earned (‘Member input)
  - Receipt printing

Example of Reactivity

- **Item Scanning**: When a user scans an item, the system reacts by adding the item to the current transaction. The system updates the total cost of the transaction in real-time based on the scanned items.

- **User Assistance**: When a user encounters difficulties during the checkout process, he can press a certain button, and the machine reacts by providing options for user assistance, such as calling an attendant or displaying on-screen instructions.

- **User Feedback**: Throughout the checkout process, the self-checkout machine reacts to user feedback, such as button presses or touchscreen interactions. It provides immediate responses to user actions.

Specifications

1. When the system starts, it displays the welcome screen and awaits the user's selection between "guest" and "member" modes. The indicator light is set to "steady green" to signal its availability to customers. The magnetic card reader, scale, and scanner are in standby mode.

2. Once a mode is selected, the indicator light turns to "steady red" to indicate it is now occupied, and the main screen is presented. The system then waits for a barcode to be detected by the scanner or for the user to click the "select item" button.

3. Upon detecting a barcode, the item is scanned. If weighing is required, a message prompts the user, and the system waits for the scale to register the weight. Additionally, if the item is intended for individuals aged 18 or older, the help screen will be displayed.

4. When the user presses the "select item" button, the item screen appears, allowing the user to manually add items. After adding an item that does not require weighing, the display returns to the main screen. If an item needs to be weighed, the system automatically follows the same process as when scanning an item, then returns to the main screen.

5. At any point while the system is on the "purchase screen" or "items screen," if the user presses the "help" button, the indicator light will begin blinking to alert the manager, and the display will switch to the help screen. Only after the "magnetic stripe card reader" reads the store card will the display return to the previous screen, depending on which screen the help button was pressed from, and the indicator light will revert to steady mode.

Specifications

6. Upon pressing the "pay" button, the display will transition to the "payment screen," where the total price will be shown based on the user's mode (member or guest).

7. Once the user confirms the displayed price, a message will prompt them to select a payment type.

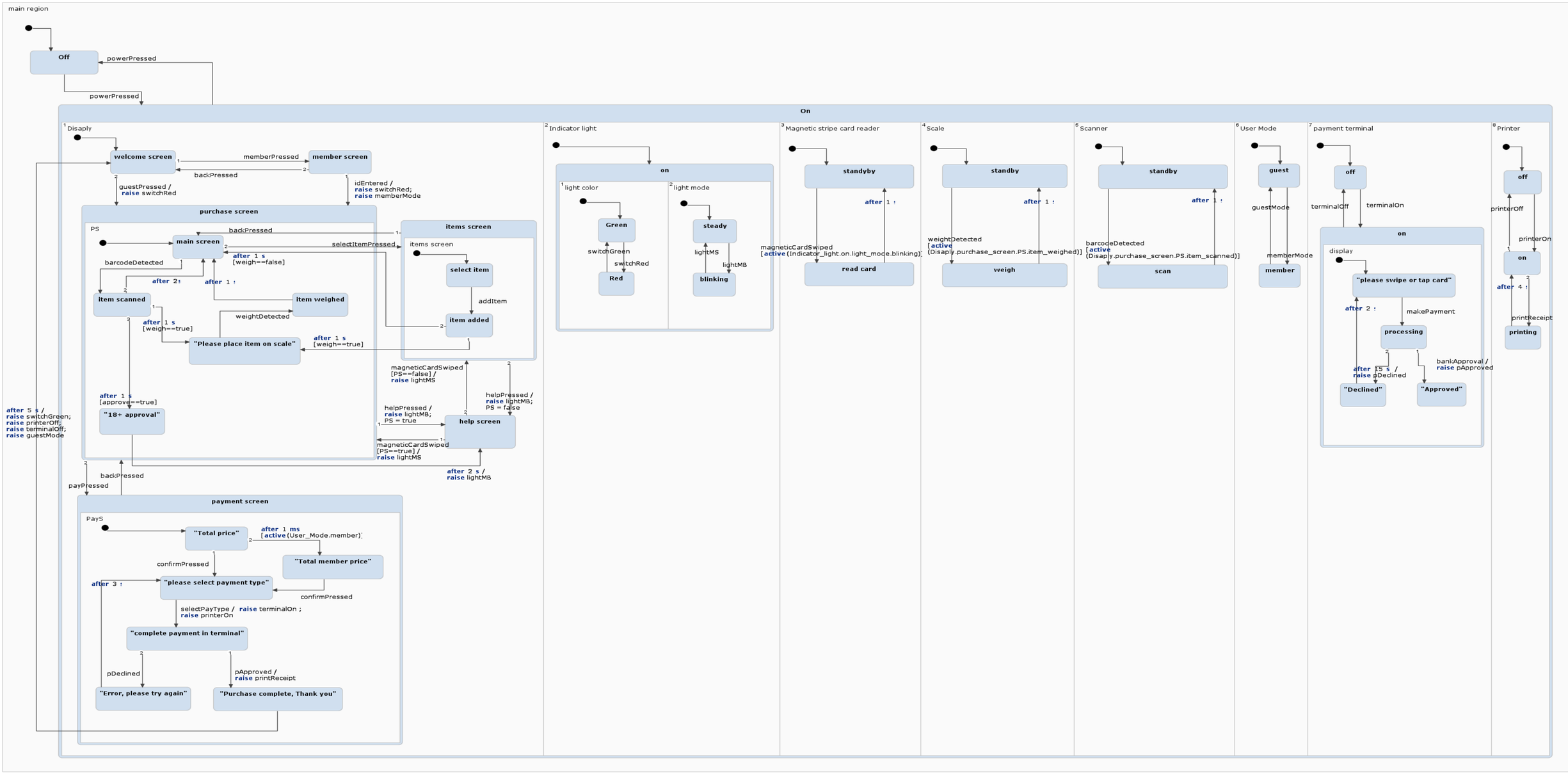
8. Upon selecting the payment type, the system will activate the "payment terminal" and "printer," displaying a message instructing the user to complete the payment using the terminal. Simultaneously, the payment terminal will prompt the user to swipe or tap their card.

9. Upon completing the payment, the terminal will process it, and depending on approval or decline, the system will display a corresponding message on the payment screen.

10. In case of a decline, the system will revert to the payment type selection phase, awaiting a new selection. Conversely, if the payment is approved, the system will print the receipt and display a message confirming the purchase completion on screen.

11. Following the completion of the purchase, the system will change the indicator light to green, power off the printer and payment terminal, and return to the welcome screen, ready for a new user to begin.

The Statechart



LSCs

Behavioral Requirements – Natural Language

- When the user presses the guest button, the display changes to the purchase screen and the indicator light changes color to red. (same as statecharts).
- When the user swipes his card in the payment terminal, the payment should be approved. When the payment is approved the printer prints a receipt, and a 'thank you' message appears on the display. (same as statecharts).
- When the user scans an item, the scanner processes the item, and a message stating item scanned will appear on the display. If the item requires approval, the display will transition to the help screen, and the indicator light will begin blinking. (same as statecharts).
- When the display is on the help screen and the manager swipes his card in the magnetic card reader, the display will revert to the purchase screen. (New).
- When the display shows a ‘thank you’ message and 5 seconds pass by, it returns to the welcome screen, the printer and payment terminal will power off, and the indicator light will turn green. (Same as statecharts).

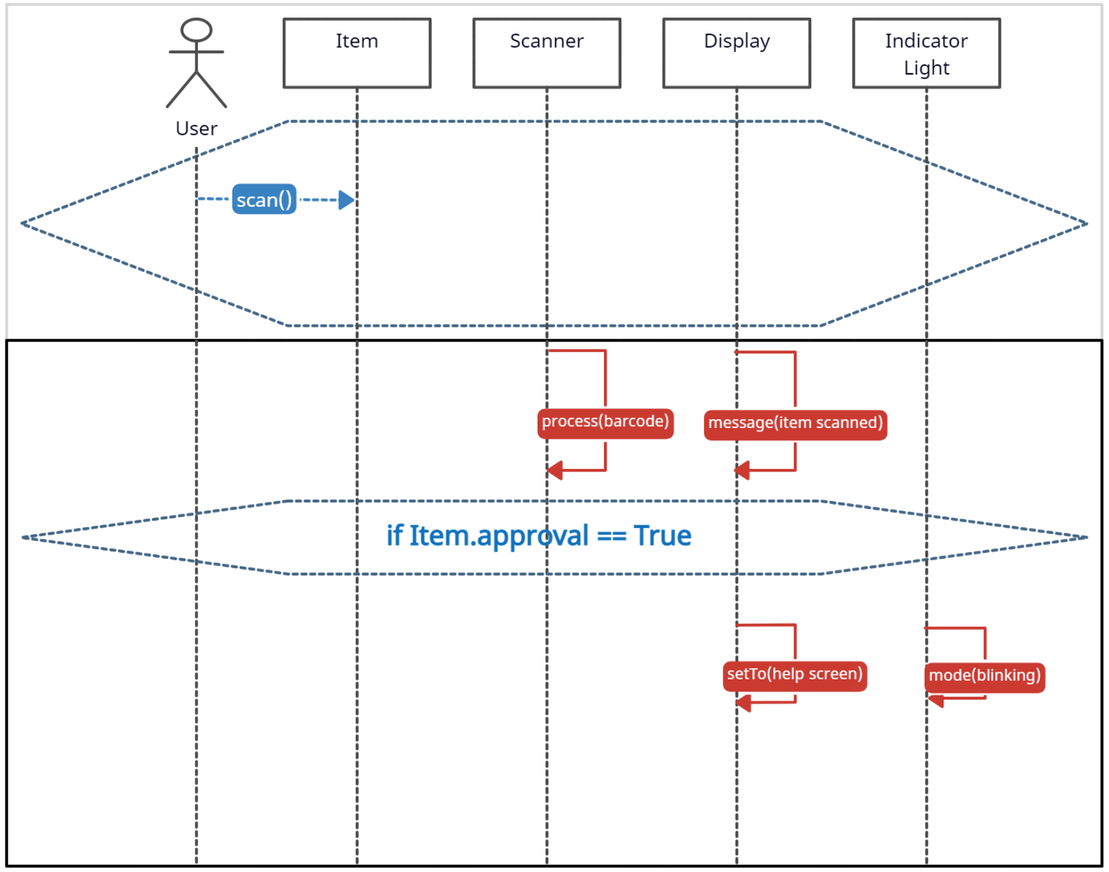
Behavioral Requirements – Controlled Natural Language

- When the guest button is pressed by the user, the display switches to the purchase screen, and the indicator light changes to red.
- When an item is scanned by the user, the scanner processes it, and a message indicating 'item scanned' will display on the display. If approval is required for the item, the display will transition to the help screen, and the indicator light will start blinking.
- When the help screen is displayed and the manager swipes his card in the magnetic card reader, the purchase screen will be displayed.
- When a 'thank you' message is displayed, and five seconds pass, the welcome screen will be displayed, the printer and the payment terminal will power off, and the indicator light will switch to green.

LSC Diagrams

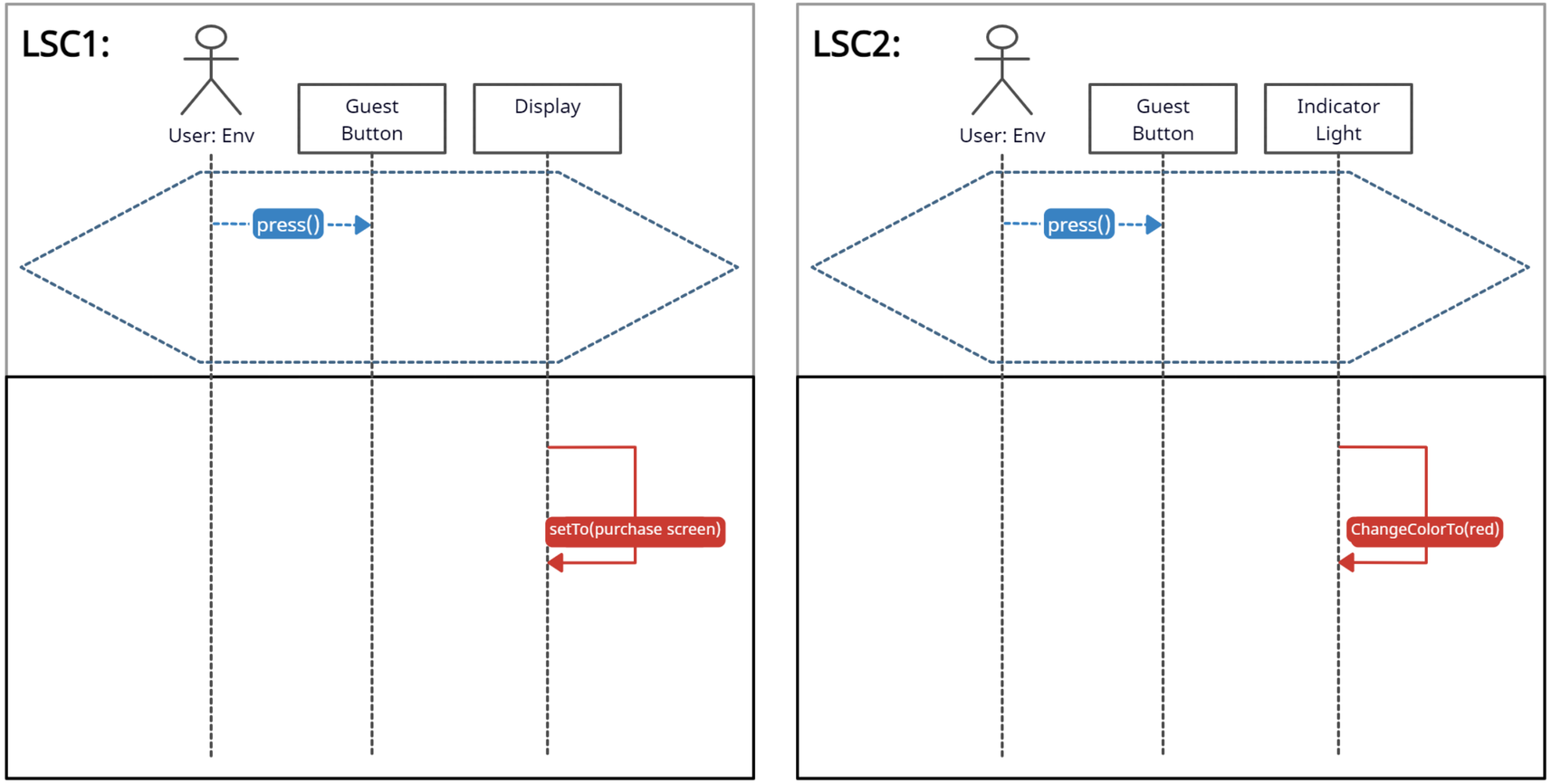
LSC Example

When the user scans an item, the scanner processes the items barcode, and a message stating item scanned will appear on the display. If the item requires approval, the display will transition to the help screen, and the indicator light will begin blinking.



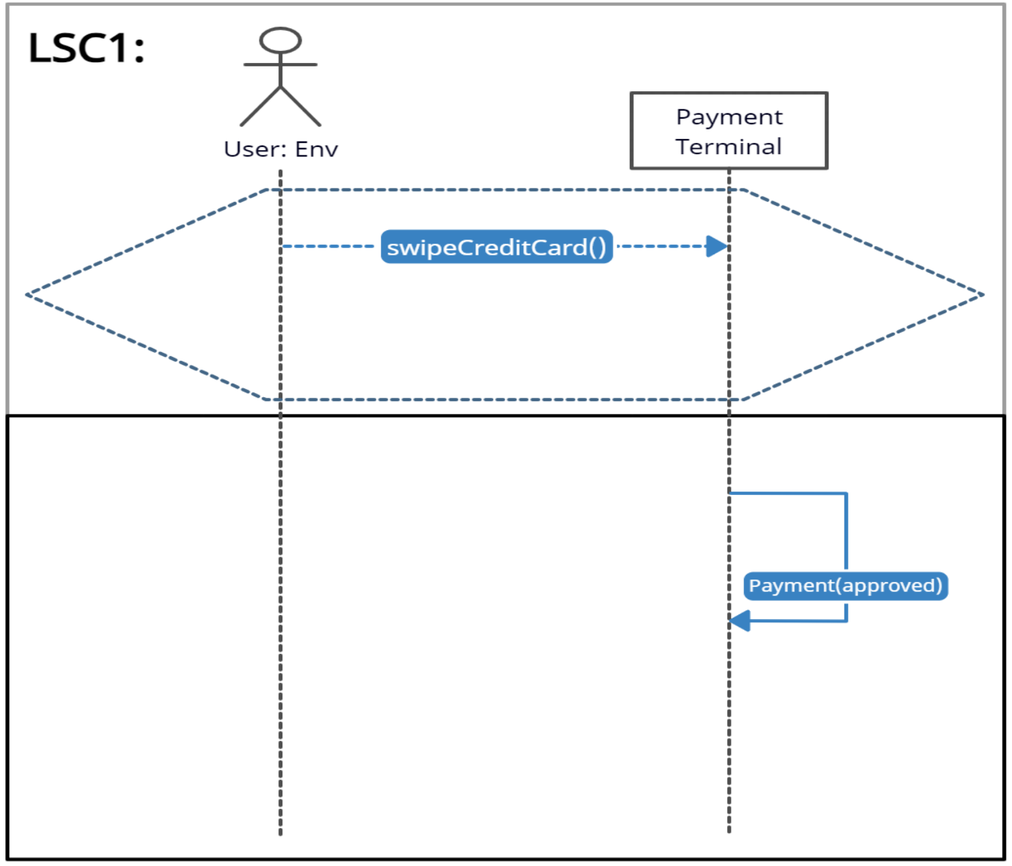
Unification in Prechart

When the user presses the guest button, the display changes to the purchase screen and the indicator light changes color to red.

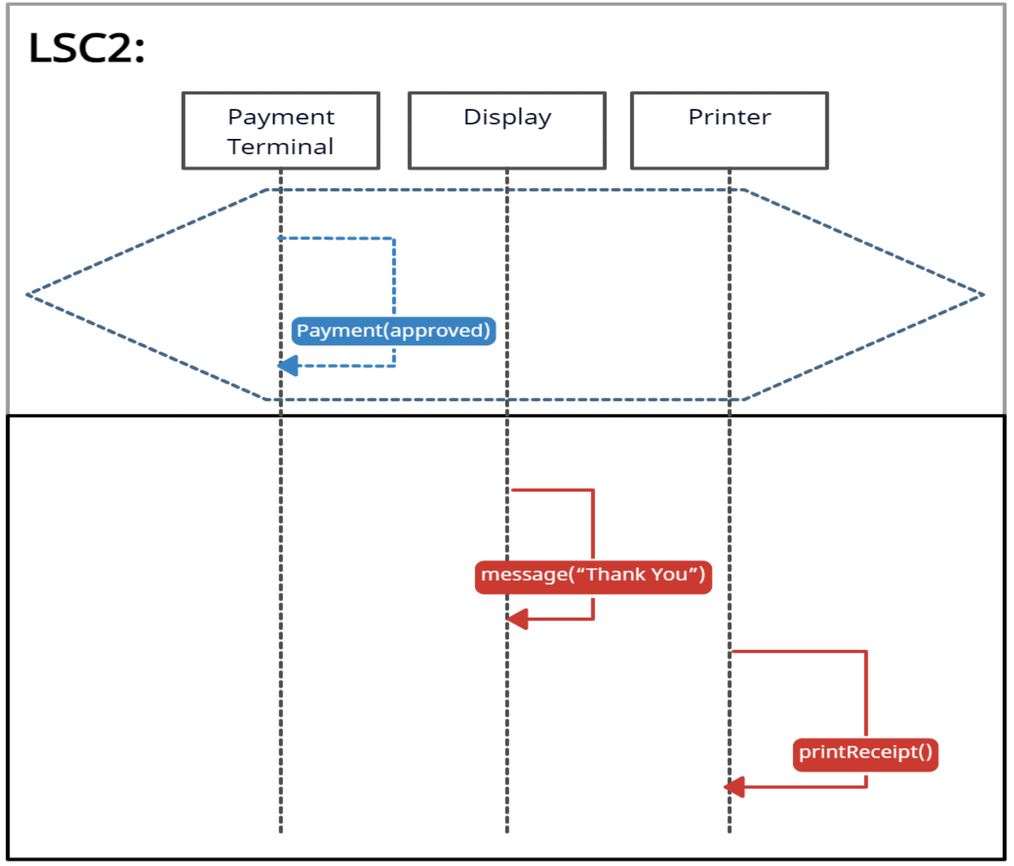


Unification – Mainchart to Prechart

When the user swipes his card in the payment terminal, the payment should be approved in the payment terminal.



When the payment is approved in the payment terminal, the printer prints a receipt, and a 'thank you' message appears on the display.

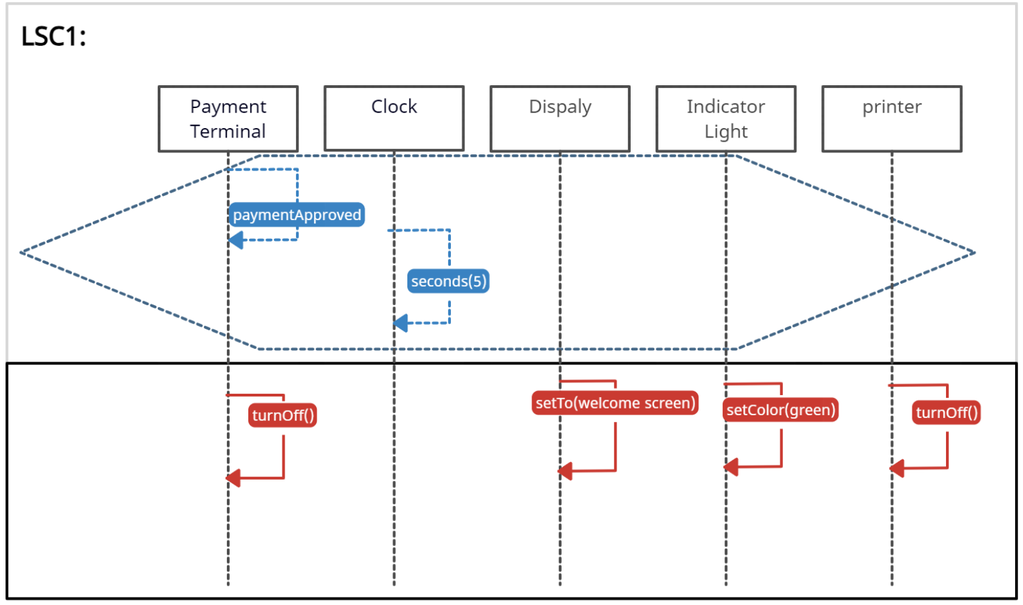


Unification – Mainchart to Mainchart

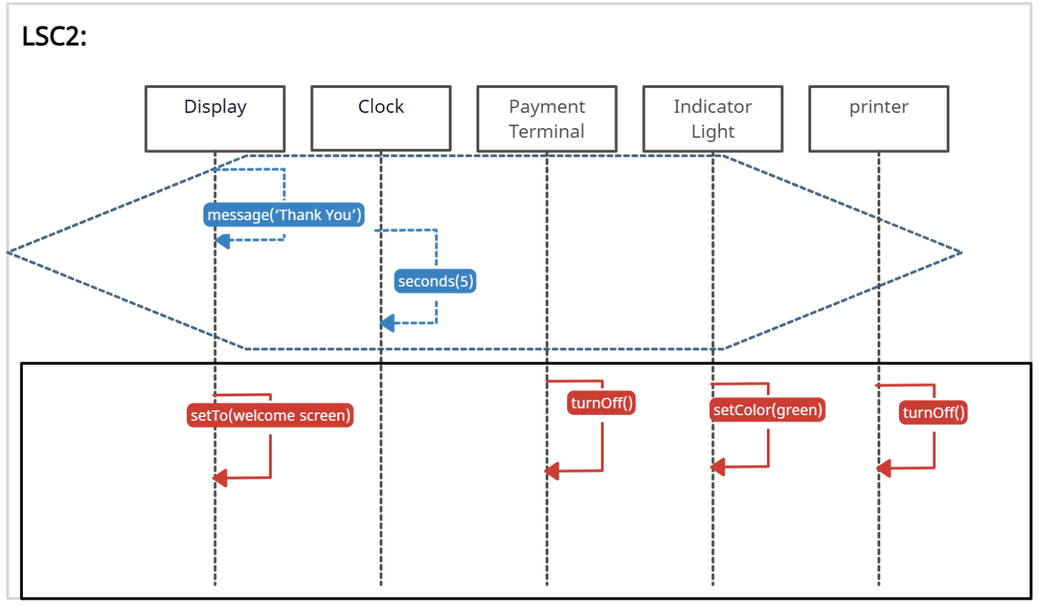
When the payment is approved in the payment terminal and 5 seconds pass by, it returns to the welcome screen, the printer and payment terminal will power off, and the indicator light will turn green.

When the display shows a ‘thank you’ message and 5 seconds pass by, it returns to the welcome screen, the printer and payment terminal will power off, and the indicator light will turn green.

LSC1:

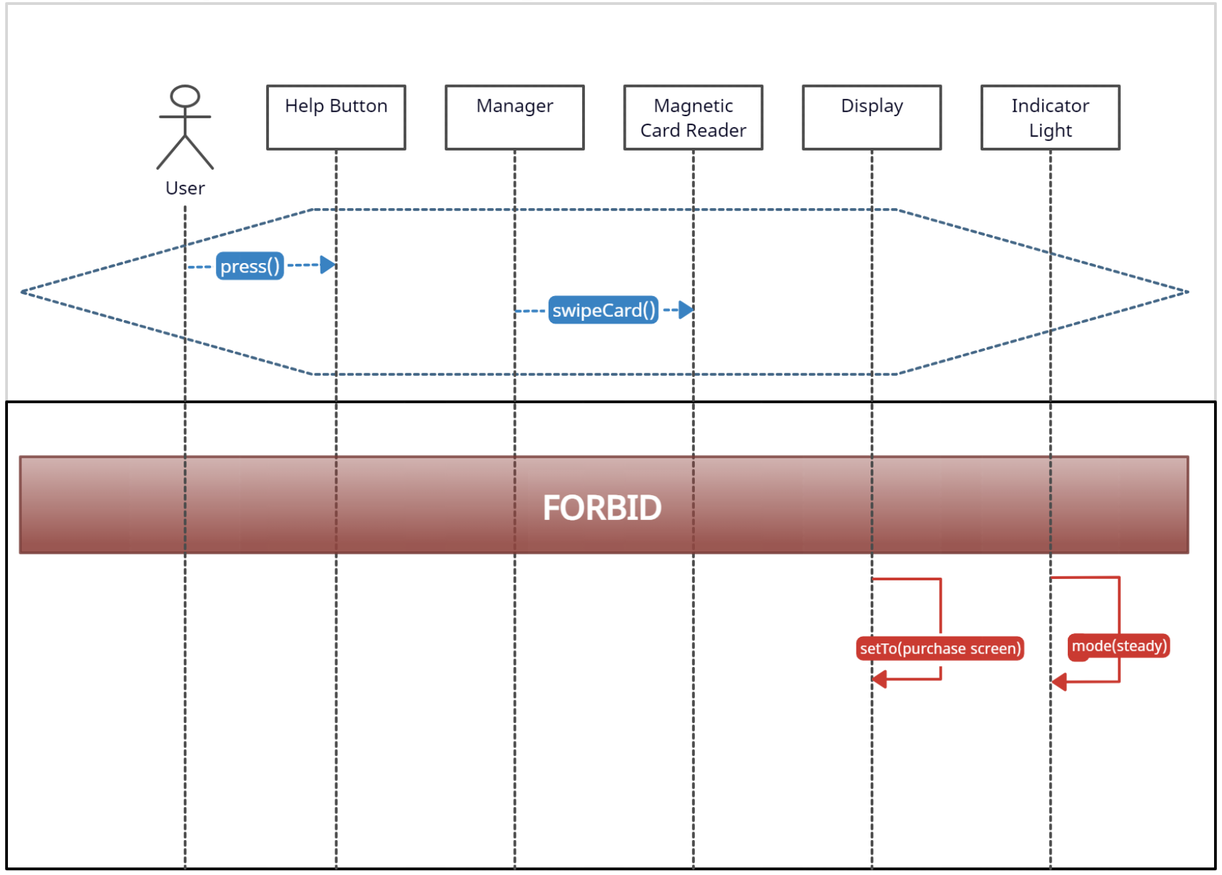


LSC2:



Forbidden LSC

When the user presses the Help Button, it is forbidden for the Display to switch to purchase screen and for the Indicator Light to switch to steady mode until the Manager swipes his card through the Magnetic Card Reader.



GUI and STATECHART Connection

- As part of the project, I developed a graphical user interface (GUI) for a self-checkout system using Python code with the Tkinter module.
- I designed a class called 'SelfCheckoutApp' to represent the GUI of the system. This class includes various GUI elements such as labels, buttons, ovals, and image displays, all of which are created and configured within its constructor. The constructor takes two parameters: 'master', which represents the root window of the GUI, and 'statechart', an instance of the Statechart class. This enables the 'SelfCheckoutApp' class to interact with the statechart, thereby reflecting the system's state in the GUI.
- Inside the class methods, there are references to methods from the Statechart class, such as 'raise_power_pressed()' and 'raise_guest_pressed()', which correspond to events raised within the statechart. These events trigger state changes and dictate the behavior of the system.
- I established the connection between the statechart behavior and the GUI using a main function, ensuring that the generated code of the statecharts remains unchanged. The main function coordinates the interaction between the GUI and the statechart behavior.
- In the main function, we initialize the statechart and connect it to the GUI using the 'SelfCheckoutApp' class. We then run a loop continuously while the GUI remains open. Within this loop, the GUI is updated based on the current state of the statechart. This involves checking the active states of the statechart and updating the GUI elements accordingly.
- During each iteration of the loop, the statecharts 'run_cycle()' method is invoked, allowing it to process events and transitions. Additionally, the 'root.update()' method is called to ensure that any changes made to the GUI are reflected on the screen.

Modeling Challenges

- In the statechart, it was challenging to think about the behavior of the system when an item is scanned and requires weighing or approval. My solution involved utilizing Boolean variables and transitioning to the appropriate state if they evaluate to true.

- Another challenge was to understand the interaction between the payment terminal and the display, whether within the statechart or the LSC.

Reflection

- Online learning was a positive experience for me. It was flexible and convenient, so I could make my own schedule that worked for me.

- Learning to write in visual programming languages was challenging and rewarding. Overall, it broadened my perspective on programming, making me think and look about it in new ways.

- I think the course material contributed to my training within the degree. I gained valuable skills and better understanding of the subject matter, which I believe will help me in the future.