# User stories

**Note**
In all the test cases if an Overwrite warning popup appears when loading or creating machines simply click the “Continue” button. If another test has been run the current state of the previous machine will have been temporarily stored.

**Test:**
Create new machine - from application launch

**Rationale:**
Users should be able to start a new, blank Turing Machine on launching the application.

**Setup:**
No setup required

**Test Procedure:**
1. Launch the application
2. Click on “New Machine”
3. Enter the name “Test New” into the text field
4. Click the “Create” button
5. Type “2” to make the text field contain “_012”
6. Click the “Define” button

**Expected Results:**
1. Display a menu screen with the options: “New Machine”, “Load Machine”, “Import Machine” and “Delete Machine”
2. Display a popup for creating a New Turing machine with a text field for naming it and “Create” and “Cancel” buttons
3. Entered text appears in the text field
4. Display the Build screen and display a popup for defining the alphabet with a text field and “Define” and “Cancel” buttons
5. Entered text appears in the text field
6. Popup disappears and the build screen is enabled

**Test:**
Create new machine - while another is open

**Rationale:**
The user should be able to create a new, blank Turing Machine when they have been working on another.

**Setup:**
Follow the steps in the "Load Machine" test case

**Test Procedure:**

1. Click the “Menu” button on the toolbar
2. Click on “New Machine”
3. Enter the name “Test New 2” into the text field
4. Click the “Create” button
5. Add a “2” to the text field so it contains “_012”
6. Click the “Define” button
7. Click the Create State button in the toolbar
8. Click on the canvas to create a state

**Expected Results:**

1. Display the menu screen with the options: “New Machine”, “Load Machine”, “Import Machine” and “Delete Machine” and “Reopen: Test Machine”
2. Display a popup for creating a New Turing machine with a text field for naming it and “Create” and “Cancel” buttons
3. Entered text appears in the text field
4. Display the Build screen and display a popup for defining the alphabet with a text field and “Define” and “Cancel” buttons
5. Entered text appears in the text field
6. Popup disappears and the build screen is enabled
7. Create State button highlighted
8. State created where the user clicks

**Test:**
Set the initial contents of the tape.

**Rationale:**
Setting the initial contents of the type will be vital to creating useful Turing Machines.

**Setup:**
Follow the steps in the "Load Machine" test case

**Test Procedure:**
1. Click on the first cell in the tape (above the number 0)
2. Enter “0”
3. Click on the second cell in the tape
4. Enter “_”
5. Click on the seventh cell in the tape
6. Enter “2”
7. Enter “1”

**Expected Results:**
1. Typing cursor appears in the first cell
2. First cell contains “0”
3. Typing cursor appears in the second cell
4. Second cell is blank. (Blanks in the machine are displayed as empty.)
5. Typing cursor appears in the seventh cell
6. The number won’t appear as it is not in the alphabet
7. Seventh cell contains “1”

**Test:**
Add a state

**Rationale:**
Users need to be able to add states to build new Turing Machines.

**Setup:**
Follow the steps in the "Load Machine" test case.

**Test Procedure:**
1. Click on the Create State button in the toolbar
2. Click on a blank part of the canvas
3. Click on another blank part of the canvas

**Expected Results:**
1. The Create State button is highlighted
2. A new state appears where the user clicked
3. A new state appears where the user clicked

**Test:**
Add a transition - between two states

**Rationale:**
Users need to be able to add transitions to build new Turing Machines.

**Setup:**
Follow the steps in the "Load Machine" test case.

**Test Procedure:**
1. Click on the Create Transition button in the toolbar
2. Click and hold on state 0
3. Drag the mouse over to state 2 and release the mouse button
4. Select the first text field and enter “_”
5. Select the second text field and enter “5”
6. Enter “2” into the second text field
7. Select the third text field and enter “1”
8. Enter “R” into the third text field
9. Click the “Set Info” button

**Expected Results:**
1. The Create Transition button is highlighted
2. State 0 is outlined in yellow
3. A “Transition Details?” popup window opens with text fields for read, write and movement
4. The first text field now contains “_”
5. No text appears in the second text field, “5” is not in the alphabet definition for this machine
6. The second text field now contains “2”
7. No text appears in the third text field, “1” is not a valid direction
8. The third text field now contains “R”
9. A transition appears from state 0 to state 2 with the details “_/2/R”

**Test:**
Add a transition - from a state to itself

**Rationale:**
Users need to be able to add transitions to build new Turing Machines.

**Setup:**
Follow the steps in the "Load Machine" test case.

**Test Procedure:**
1. Click on the Create Transition button in the toolbar
2. Click on state 2
3. Select the first text field and enter “_”
4. Enter “2” into the second text field
5. Enter “R” into the third text field
6. Click the “Set Info” button

**Expected Results:**
1. The Create Transition button is highlighted
2. A “Transition Details?” popup window opens with text fields for read, write and movement
3. The first text field now contains “_”
4. The second text field now contains “2”
5. The third text field now contains “R”
6. A transition appears from state 1 to itself with the details “_/2/R”

**Test:**
Set the start state

**Rationale:**
The user should be able to define where the turing machine starts.

**Setup:**
Follow the steps in the "Load Machine" test case.

**Test Procedure:**
1. Click on the Set Start State button on the toolbar
2. Click on state 2
3. Click on state 1
4. Press the Run button on the toolbar

**Expected Results:**
1. The Set Start State button is highlighted
2. State 2 is coloured red to show it is the start state. State 0 is coloured white.
3. State 1 is coloured red and state 2 is reverted to white.
4. State 1 has the yellow outline marking it as the current state

**Test:**
Set an accepting state

**Rationale:**
The user must be able to set states as accepting states for the machine to halt correctly.

**Setup:**
Follow the steps in the "Load Machine" test case

**Test Procedure:**
1. Click on the Set Accepting State button on the toolbar
2. Click on state 2
3. Click on state 0
4. Click on state 2

**Expected Results:**
1. The Set Start Accepting button is highlighted
2. State 2 has an extra circle inside it to show it is an accepting state
3. State 0 has an extra circle inside it to show it is an accepting state
4. State 2 is no longer an accepting state and no longer has an extra circle

**Test:**
Run to completion

**Rationale:**
The turing machine simulator needs to be able to run turing machines to have a practical application.

**Setup:**
Follow the steps in the "Load Machine" test case

**Test Procedure:**
1. Click the Run button on the toolbar
2. Click the Run to Finish button on the Run toolbar
3. Click the Build button on the toolbar
4. Click the Set Final State button on the Build toolbar
5. Click on State 2
6. Click the Run button on the toolbar
7. Click the Run to Finish button on the toolbar

**Expected Results:**
1. The machine switches to Run Mode and displays the Run toolbar instead of the Build toolbar
2. A popup appears that says the machine halted with answer “no” in 9 steps
3. The machine switches to Build mode
4. The Set Final State button is highlighted
5. State 2 is set as a final state and has an inner circle
6. The machine switches to Run mode
7. A popup appears that says the machine halted with answer “yes” in 9 steps

# Stretch user stories

**Test:**
Save machine

**Rationale:**
For the turing machine simulator to be of long term use it will have to be able to save and load turing machines.

**Setup:**
Follow the steps in the “Create new machine - from application launch” Test Case but name the machine “Test Save”

**Test Procedure:**
1. Click on the Create State button on the toolbar
2. Click on the canvas to create a state
3. Click on another part of the canvas to create a second state
4. Click the Create Transition button on the toolbar
5. Click on state 0 and drag to state 1
6. Fill in the text fields with 0/1/R
7. Click the “Set Info” button
8. Click the Save button in the toolbar
9. Close the Application
10. To test that the Save worked, follow the Load Test case but replace the filename “Test Machine” with “Test Save”

**Expected Results:**
1. The Create State button is highlighted
2. A state is added where the user clicked
3. A second state is added where the user clicked
4. The Create Transition button is highlighted
5. A “Transition Details?” popup window opens with text fields for read, write and movement
6. The text fields are filled in with the user’s input
7. The popup closes and the transition is added
8. The machine is saved

**Test:**
Load Machine

**Rationale:**
Users will need to be able to load saved turing machines.

**Setup:**
NA

**Test Procedure:**
1. Launch the application
2. Click Load Machine
3. Click “Test Machine”

**Expected Results:**
1. Display a menu screen with the options: “New Machine”, “Load Machine”, “Import Machine” and “Delete Machine”
2. Open a list of saved turing machines, with their filename and size
3. The saved Turing Machine “Test Machine” is loaded and the Build screen is opened

**Test:**
Step through execution

**Rationale:**
Often it is very useful to be able to step through a Turing Machine to see how it functions.

**Setup:**
Follow the steps in the "Load Machine" test case

**Test Procedure:**
1. Click the Run button in the toolbar
2. Click the Step button in the toolbar
3. Repeat step 2 until the machine halts

**Expected Results:**
1. The simulator switches to the Run mode and toolbar
2. The simulator highlights the current state by surrounding it with a yellow circle, highlights the transition in yellow as it changes state, moves the window to show the path of the transition, updates the values in the tape at the bottom, highlights the current tape cell in yellow and scrolls the tape so the current cell is in the centre.
3. When the user clicks the Step button while the Turing Machine is on state 2 it will halt and a popup will appear to inform the user that the simulation halted with the answer ‘No’