Use Case 1	**Edit variable**
Primary Actor:	End-User
Preconditions:	Add method which does have a variable parameter.
Postconditions:	Variable is defined.

Main Success Scenario:

1. User selects variable
2. User changes the parameter
3. Check type
4. if it is a compatible tyep allow input and save parameter

Extensions:

2.a Invalid type data:

1. Incorrect input
2. User returns to step 1 or exits

Non-Functional Requirements Needed:	• Learning Experience. • Usability.

Use Case 2	**Dragging and dropping functionality for methods**
Primary Actor:	End-User
Preconditions:	There is a method for which it is possible to be selected.
Postconditions:	Method is now ready to be use.

Main Success Scenario:

1. User creates new variable and selects it
2. They can now drag and drop the variable in the boundaries provided

Extensions:

2.a Invalid drop location:

1. The user attempts to drag the variable outside of acceptable boundaries, the variable will no be "locked" inside of boundary.

Non-Functional Requirements Needed:	• Learning Experience. • Usability.

Use Case 3	**Instantiating a Conditional Statement**
Primary Actor:	End-User
Postconditions:	A conditional is instantiated

Main Success Scenario:

1. A user drags and drops a conditional statement
2. they change the parameter (e.g. if this)
3. the inside of the conditional is then draged and dropped

Non-Functional Requirements Needed:	• Learning Experience. • Usability.

Use Case 4	**Instantiating a Boolean Operator**
Primary Actor:	End-User
Postconditions:	A boolean operator is instantiated

Main Success Scenario:

1. A user drags and drops an operation (e.g. and, or, not)
2. the user sets the two variables or expressions

Non-Functional Requirements Needed:	• Learning Experience. • Usability.

Use Case 5	**Connecting actions**
Primary Actor:	End-User

Preconditions:	There are two or more actions on the development board.
Postconditions:	The methods are connected.

Main Success Scenario:

1. The method is draged and droped by the user above or below the action they want to connect to
2. The user releases the method

Non-Functional Requirements Needed:	• Learning Experience. • Usability.

Use Case 6	**Save a Program**
Primary Actor:	End-User
Preconditions:	Add method which does have a variable parameter.
Postconditions:	Variable is defined.

Main Success Scenario:

1. The user selects save this adds the current development board to a list that the user can access

Extensions:

2.a Unnamed program:

1. The user will name the program

Non-Functional Requirements Needed:	• Usability.

Use Case 7	**Delete a method**
Primary Actor:	End-User
Preconditions:	There are methods on the development board
Postconditions:	Selected methods are deleted

Main Success Scenario:

1. The user selects a method or group of methods
2. The user selects to delete the selected items

Extensions:
2.a Invalid type data:

1. Incorrect input
2. User returns to step 1 or exits

Non-Functional Requirements Needed:

- Learning Experience.
- Usability.

Use Case 8	**Run the Program (play)**
Primary Actor:	End-User
Preconditions:	Methods have been added to the development board.
Postconditions:	The program has been run and the state is maintained.

Main Success Scenario:
1. user selects the play button
2. the program is compiled and if there are no errors the program is run.

Extensions:
2.a Compile time errors:

1. Does not run the program
2. Highlights error for user

Non-Functional Requirements Needed:

- Learning Experience.
- Usability.

Use Case 9	**Pause the Program**
Primary Actor:	End-User
Preconditions:	The program is running.

Postconditions:	The state of the program when it was paused is maintained.

Main Success Scenario:

1. the user pauses the program
2. the program stops and maintains the current state

Non-Functional Requirements Needed:	• Learning Experience. • Usability.

Use Case 10	**Instantiating a Loop**
Primary Actor:	End-User
Preconditions:	Add method which does have a variable parameter.
Postconditions:	A loop is instantiated

Main Success Scenario:

1. The user drags and drop a loop to the development board
2. The user then inputs the conditionals for the loop and its exit conditions.

Extensions:

2.a Invalid input:

1. User returns to step 1 or exits

Non-Functional Requirements Needed:	• Learning Experience. • Usability.