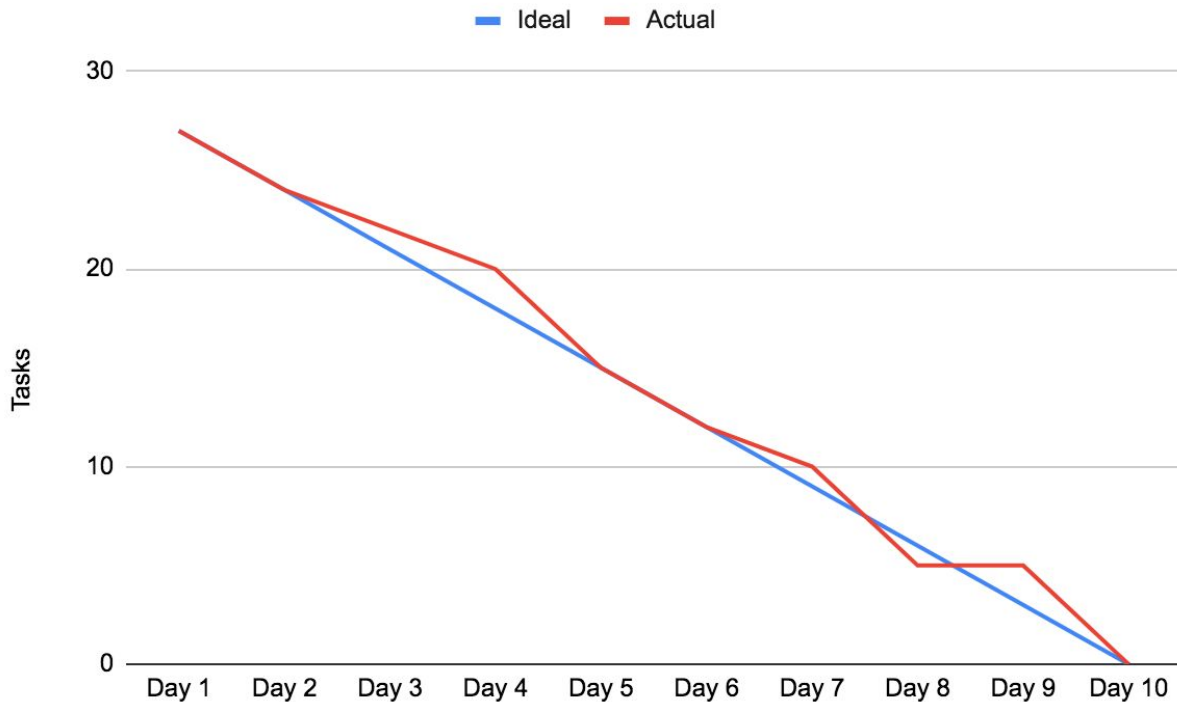


Team 5  
Team member:  
Yijian Hu  
Shih-Yu Chang  
Kai-Rui Hsu  
Hongqi Zhang

## 1. Burndown Chart



## 2. Sprint Backlog

### Items:

1. As a user, I want a start button box which contains an open parenthesis description and a right button dot.
2. As a user, I want an end button box which contains a close parenthesis description and a left button dot.
3. As a user, I want a left button box which contains an open-angle parenthesis description and three dots button, two of which are on the right side, and the other is on the left side.
4. As a user, I want a right button box that contains a close angle parenthesis description and three dots button, two of which are on the left side, and the other is on the right side.
5. As a user, I want a loop button box that contains '@' symbol description and four dots button, two of which are on the left and right side respectively.
6. As a user, I want a bar button box that contains '||' symbol description and two bars, one of which is on the left side, and the other is on the right side.
7. As a user, I want a dot button box that contains '-' symbol description in the middle of the two-dot buttons on each side.

- [illegible]

		2. StartButtonBox class implementation. 3. Testing.											
2	2	1. EndButtonBox class implementation 2. Testing.	0	1.5	1	0	0	0	0	0	0	0	Hongqi Zhang
3	2	1. LeftButtonBox class implementation. 2. Testing.	0	1.5	1	0	0	0	0	0	0	0	Hongqi Zhang
4	2	1. RightButtonBox class implementation. 2. Testing. 3. Add set, get method, and connection method to the class.	0	2	1	0	0	0	0	0	0	0	Hongqi Zhang
5	2	1. LoopButtonBox class implementation. 2. Testing.	0	0	1	0	0	0	0	0	0	0	Hongqi Zhang
6	2	1. BarButtonBox class implementation. 2. Testing.	0	0	1.5	0	0	0	0	0	0	0	Hongqi Zhang
7	2	1. DotButtonBox class implementation. 2. Testing.	0	0	1	0	0	0	0	0	0	0	Hongqi Zhang
8	4	1. Research about singleton, creational factory, and observer design pattern. 2. implement a ButtonBox Factory class. 3. implement Box class.	0	0	0	3	3	1.5	1.5	0	0	0	Hongqi Zhang
9	2	1. Connection class implementation 2. Modify Connections data structure. 3. Make the line directed 4. testing	1	0	0	0	0	0	0	1	0	0	ShihYu Chang, Hongqi Zhang

9,10 ,12, 13, 14, 15	4	1. ConnectionCon troller class implementation 2. Make the connection rules testing 3.	0	4	3	1	0	0	0	0	0	0	ShihYu Chang
11	2	1. Line class implementation testing 2.	2	0	0	0	0	0	0	1	0	0	ShihYu Chang
23	2	1. Connection moving implementation testing 2.	0	0	0	2	0	0	0	0	0	0	ShihYu Chang
16	1	1. Connection delete implementation testing 2.	0	0	0	1	0	0	0	0	0	0	ShihYu Chang
14	1	1. Modify the connection painting method. testing 2.	0	2	0	0	0	0	0	0	0	0	ShihYu Chang
17	4	1. Implement JOptionPane 2. Testing assign and read value from JOptionPane	0	0	0	0	0	0	1	1	1	0	KaiRui Hsu, Hongqi Zhang
18	1	1. Add a compile Jbutton in menubar class 2. Testing	0.5	0	0	0	0	0	0	0	0	0	KaiRui Hsu
19	1	1. Add a new space Jbutton in menubar class 2. Testing	0.5	0	0	0	0	0	0	0	0	0	KaiRui Hsu
20	1	1. Modify the listener in menubar class to create tabs	0	0	0	0	0	0	0	0	0.5	0	Yijian Hu
21	4	1. Learn things about JTabbedPane 2. Replace original RightPanel with a new JTabbedPane. 3. Modify the original RightPanel to make it work	2	2	0	0	0	0	4	4	0	1	Yijian Hu

		with multiple tabs 4. Create a new helper class to store relationships between different items appear on tabs.											
22	2	1. Modify the original FileManager to fetch and save data from Model class	0	0	0	0	0	0	0	0	3	1	Yijian Hu
23	2	1. Modify the original FileManager to read data from newly constructed xml file and load it to program	0	0	0	0	0	0	0	0	2	1	Yijian Hu, KaiRui Hsu
24	2	1. Learn how to use multi thread in java 2. Compilation class implementation 3. Testing	0	0	0.5	0.5	0	0	0	2	0	0	Kairui Hsu
25	1	1. Update compilation class 2. Testing	0	0	0	0	0	0	2	0	0	0	Kairui Hsu
26	4	1. Learn how to use set, union-find data structure in graph to check if there is loop inside a graph 2. Update compilation class 3. Create Graph class 4. Testing	0	0	0	0	0	0	0	2	2	1	Kairui Hsu
27	2	1. Learn how to use join() in java to wait all threads finished and pop out all results at a time 2. Update compilation class and menubar class 3. Testing	0	0	0	0	0.5	0	0	0	0	1	Kairui Hsu

## Sprint Review

- what has been done

In this sprint, we have finished all of our tasks. We researched how to implement and develop those tasks.

- what has not been done

Nothing

- what work that has been add

1. Button Box with different symbols and dot buttons.
2. Dot-buttons Connection between two or more Button Boxes.
3. Compiler that compiles the Button Boxes and their connections .
4. Tab Page which allows multiple working areas.
5. JOptionPane that can edit a Button Box's description.

## 4. Sprint Retrospective

- What went well in this Sprint

We designed the architecture for our software before we start coding. We use some design patterns in this sprint, and we start early for our project 4. We have good communication this sprint which means we share the information and work progress in time.

- What could be improved

The architecture could be better. The Model class could be more powerful if we move the data manipulation from ConnectionController to this class. Aslo, we are trying to add observer design pattern in the next sprint

- What will we commit to improve in the next Sprint

Design a better architecture.