**CPS Faster Payment- Data Input Automation**

**Alpha Mackie**

**Software Developer Level 4**

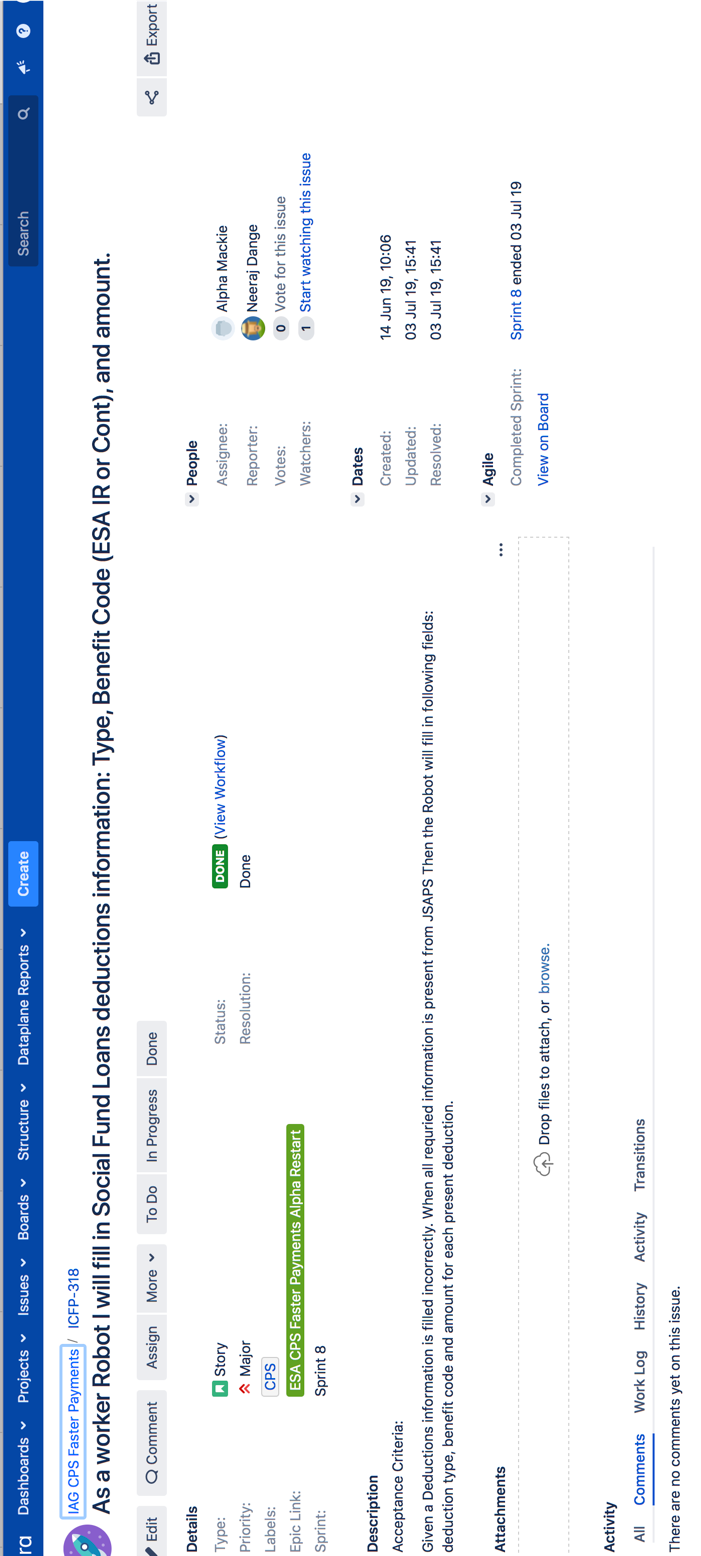
Introduction

In this piece of work, I was tasked with automating data entry into some input fields in the Customer Payment System (CPS) application. The software robot should fill in the deduction type, benefit code and amount in the respective fields as stated in the user story.

The technologies used for this project are:

* Jira and Confluence for recording and tracking the progress of the user stories.
* UIpath Studio for building the Robots.

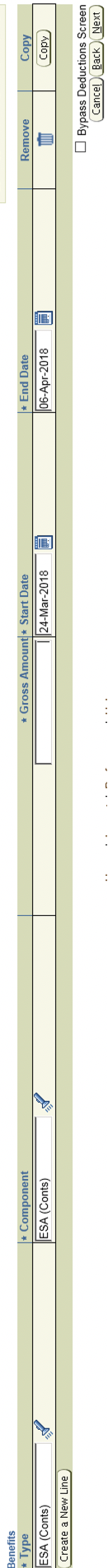
Please see the user story below:



*Figure 1: User story for completing input fields.*

First step

As usual I consulted with the lead developer during the first week of our sprint meeting, to get a better understanding of what was required from me. I also went through the flow diagram to make sure I build the right components to match the flow diagram and the user story**.** Please see below, figure 2a, an image of the part of the application to be automated and figure 2b, a snippet of the flow diagram.



*Figure 2a: The input fields with the required data inputted.*

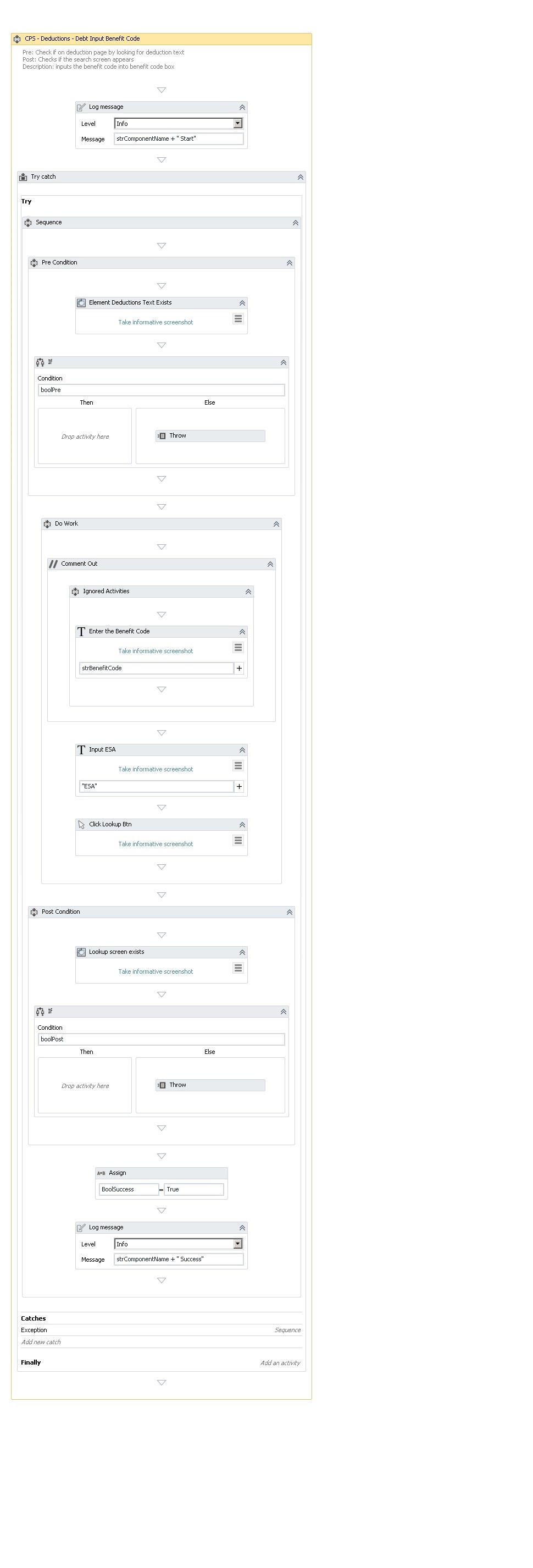
*A screenshot of a cell phone

Description automatically generated*

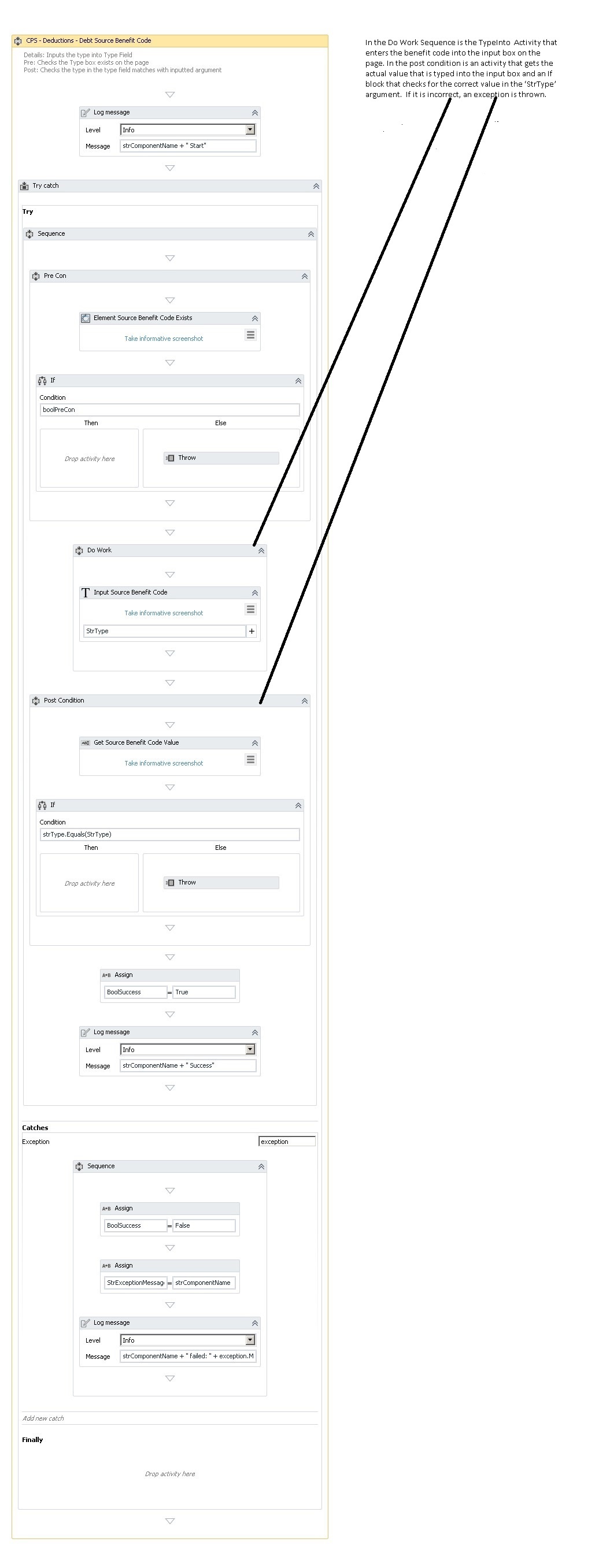
*Figure 2b: A snippet of the Flow diagram.*

Development process

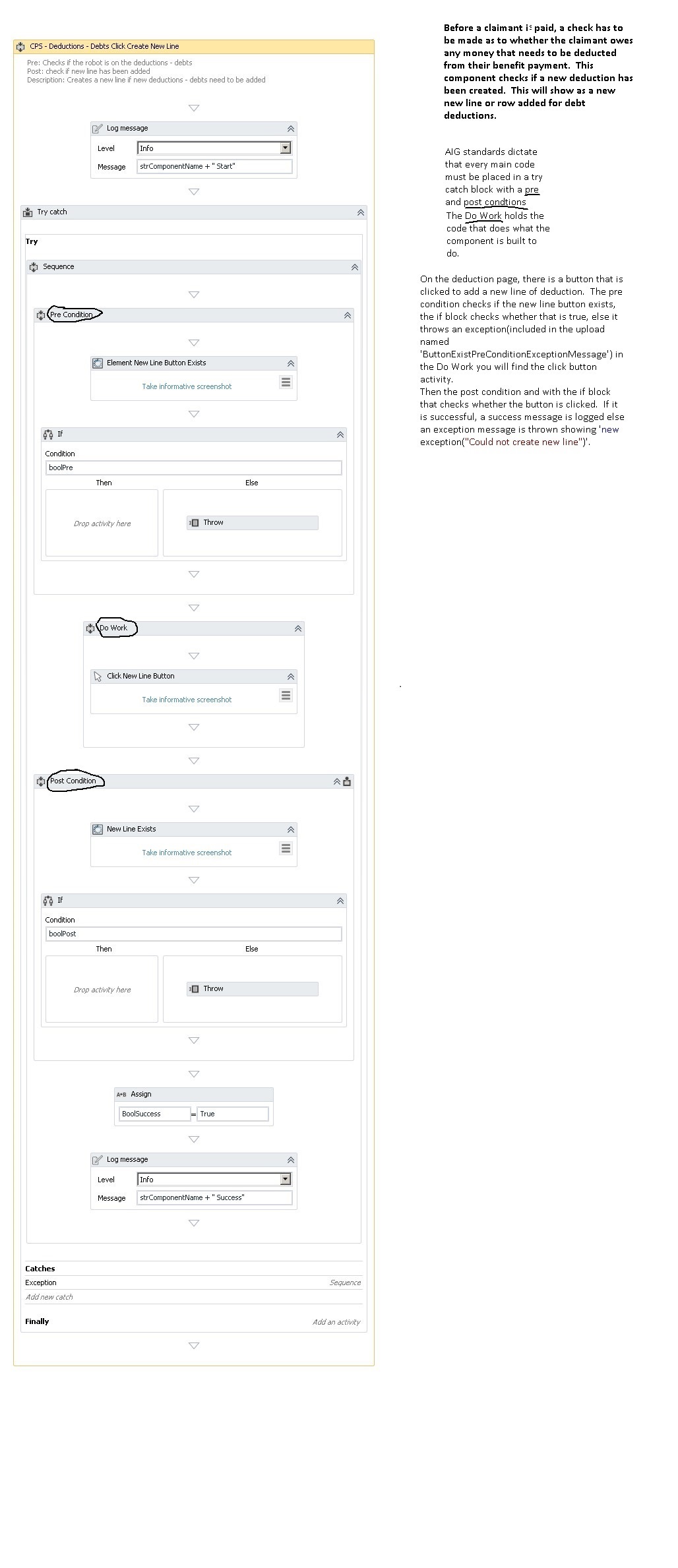
In this scenario I built fourcomponents, one for each data input field. Each component has a start and end log, a Pre-condition activity that checks whether a targeted input box in the application exists, ifit does, the process will continue or throws an error. The Do-Work section is where the actual data entry process takes place and then the Post-condition checks whether the process was successful, and also to check whether the first stage of the next process is in place. This is the build process for all the components as can be seen in the screenshots below. I have also included some annotations to clarify how the activities work, on some of the screenshots below.



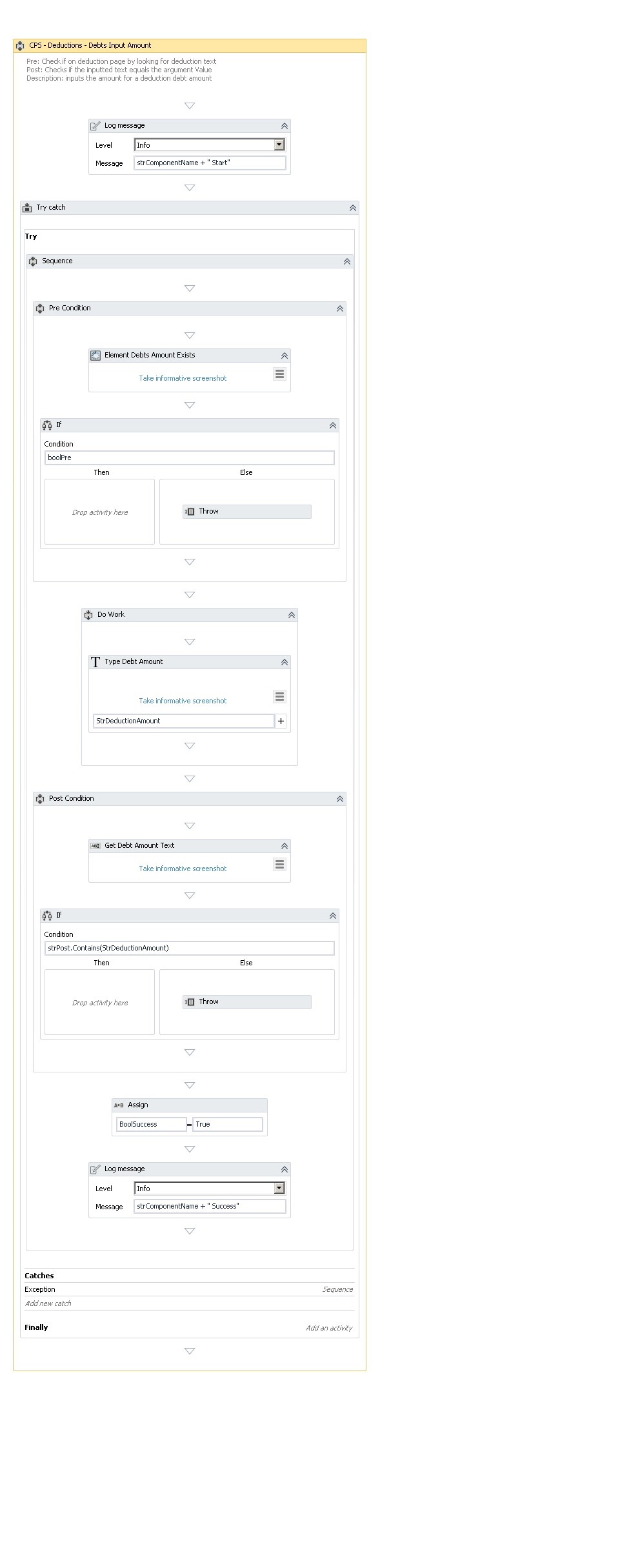
*Figure 3a: This component input the benefit type into the type input field.*



*Figure 3b: This component inputs the benefit code.*



*Figure 4: This component creates a new line of deduction.*



*Figure 5: This component inputs the deduction amount.*

Testing

The testing was done by the lead developer and the tests were successful in the development environment. The selectors worked well; dummy data was used for the automation, which was entered by the software robot in all the fields successfully.

Final steps

The senior developer is then provided with the opportunity to test and confirm that all the components work before they are integrated into the workflow. If so, it is approved on JIRA as confirmed by the senior developer and the Scrum master.

Conclusion

It was not really challenging to build the components this time as the flow diagram is easy to follow and with experience of building and testing previous components which have been reviewed by my lead developer. This has boosted my confidence greatly. I could have built a single component for the deductions user story but I thought it would be better to build separate components to handle each of the actions so that errors can be detected easily and if things break when components are integrated into the workflow, bugs can be tracked in less time(debugging). Also Object-Oriented methodology is applied as each component is treated as an object that does only one thing(single responsibility).