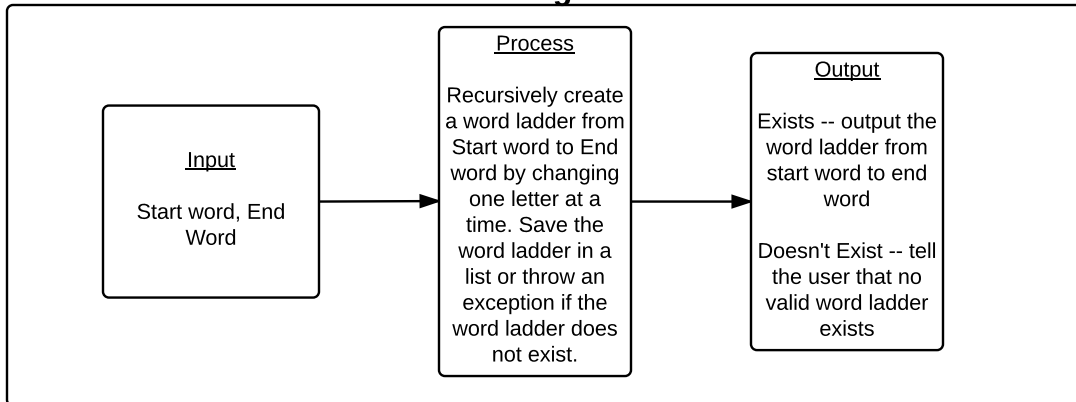
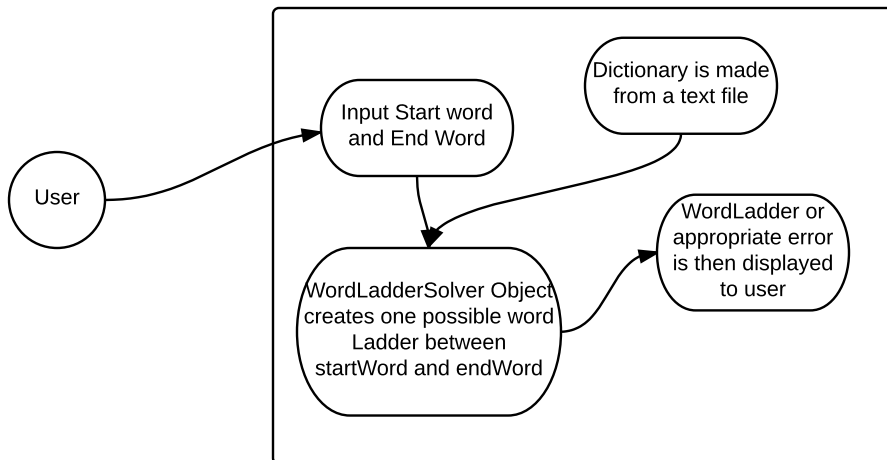


Assignment 4 Design Document

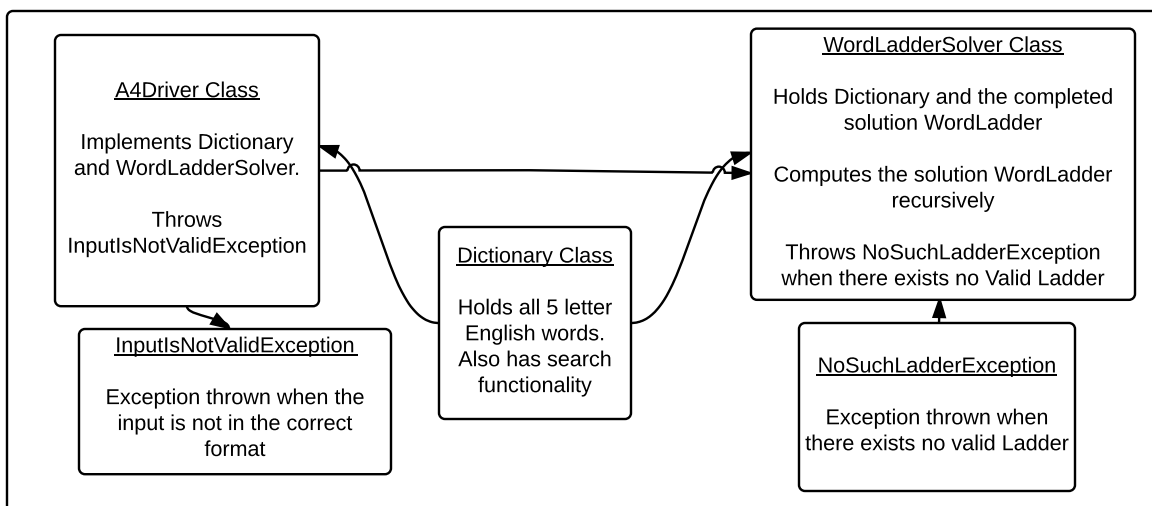
IPO Diagram



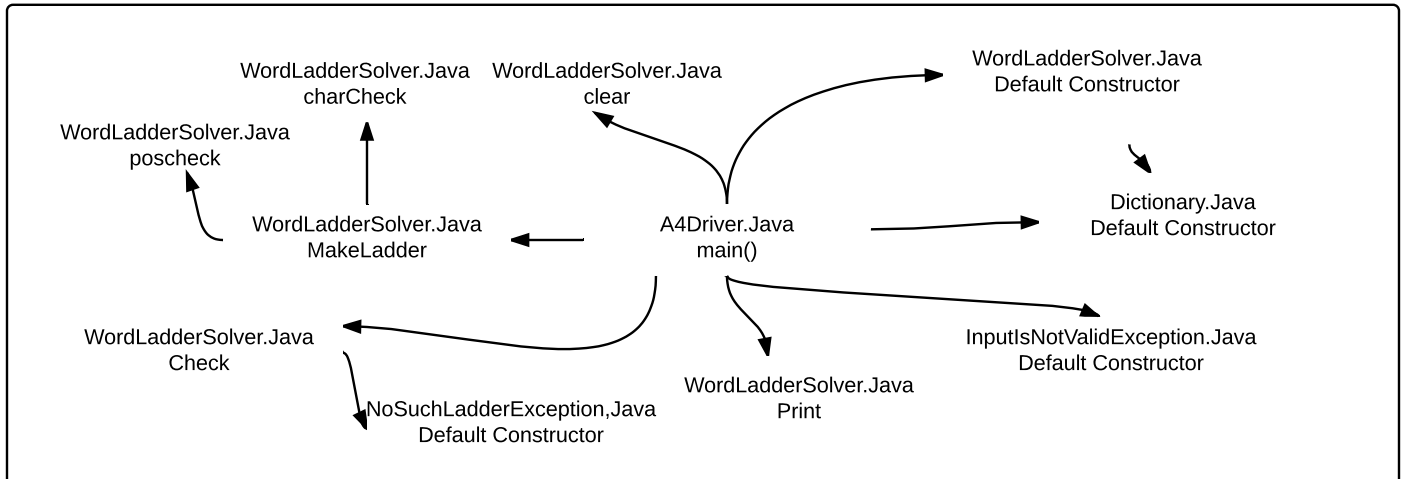
Use-Case Diagram



UML Class Diagram



Functional Block Diagram



Main Method Algorithm

- 1) Check to see if there are files input
- 2) Parse first file input as Dictionary
- 3) Start reading user input from second file
- 4) If input line is valid then try making a Word Ladder
- 5) Output either WordLadder or appropriate error indicating no WordLadder could be made
- 6) Clear the word ladder and continue on to next user input
- 7) If there are no more input lines, end program

Design Rationale

We decided to make the Dictionary its own separate class in order to emphasize its importance to the program. The dictionary is the backbone of the program as it is what we check every single word against. We also implemented the recursion inside of the `WordLadderSolver` object itself in order to gain access to the `Solutionlist` that is found inside of the `WordLadderSolver` object while performing the recursion. Lastly we decided to create two different types of exception classes in order to clearly specify what type of exceptions were being thrown at any point in the program, avoiding any possible confusion due to mislabeled exceptions.