REVISION HISTORY

Date	Version	Description	Author
14.10.2021	0.1	Software Architecture overview is created	Uygar KAYA
15.10.2021	0.2	InputOutput Package is created	Uygar KAYA
17.10.2021	0.3	Algorithm Package is created	Uygar KAYA
18.10.2021	0.4	The whole diagram is created	Uygar KAYA
19.10.2021	0.5	COTS Identification is created	Uygar KAYA
07.11.2021	0.6	The Software Detailed Design Document is updated	Uygar KAYA

TABLE OF CONTENTS

1 Introduction

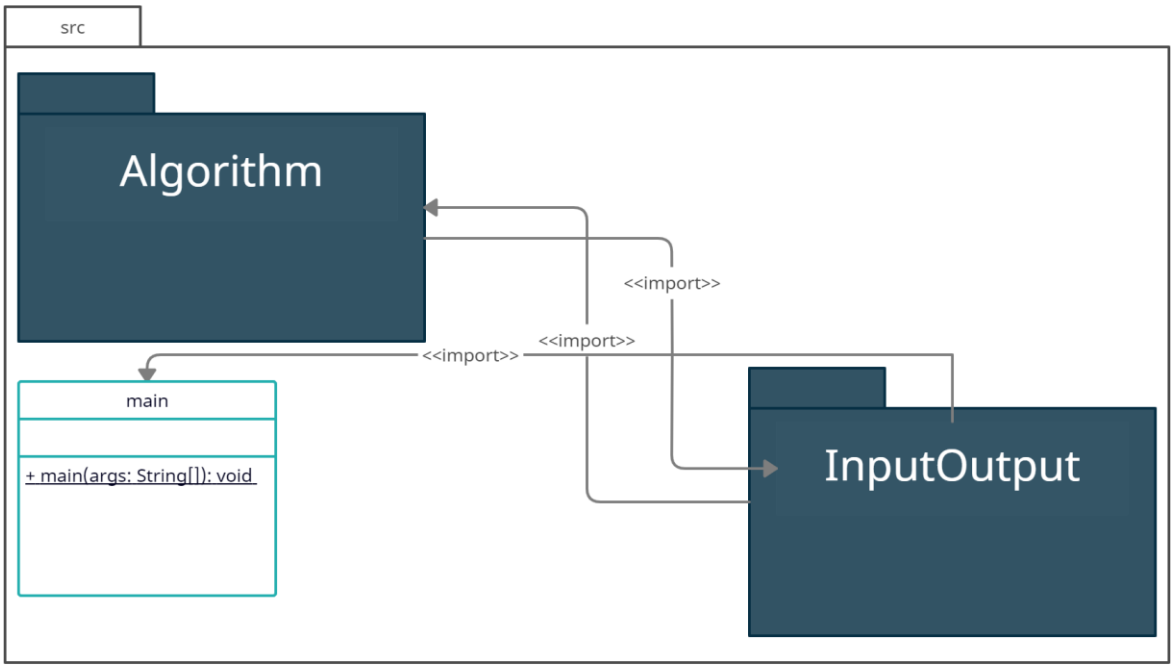
This document describes the design of the DFA (Deterministic Finite Automata) software system. In this project, I programmed the DFA Algorithm, which is given a string the simulated DFA should be able to tell if the string is accepted or rejected. In this project, I will implement the program with Java Programming Language.

1.1 References

1.1.1 Project References

#	Document Identifier	Document Title

2 Software Architecture overview



3 Software design description

3.1 InputOutput

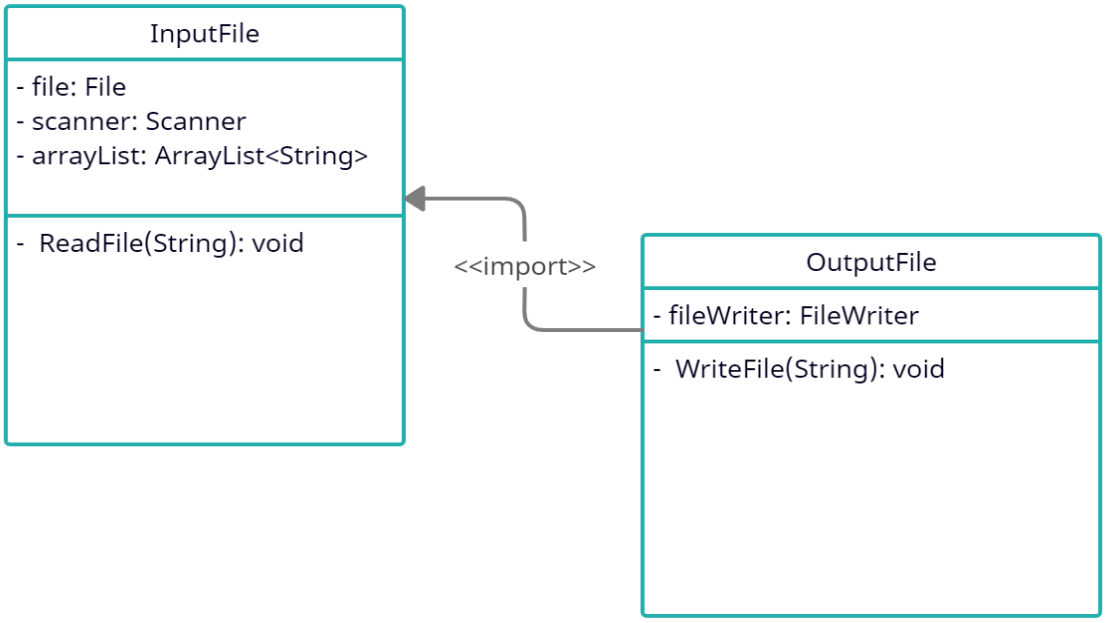
3.1.1 Component interfaces

In the InputOutput Package, there are two classes which are InputFile & OutputFile

Methods of InputFile which are available from other components are:
public void ReadFile(String filePath) – To read the input .txt file

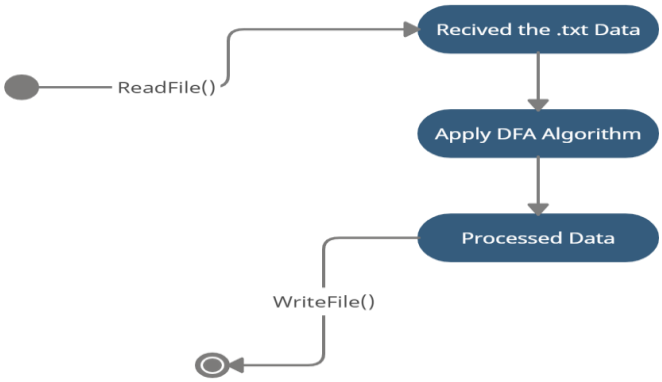
Methods of OutputFile which are available from other components are:
public void WriteFile(String routeValidation) – To write the output .txt file

3.1.2 Component design description



3.1.3 Workflows and algorithms

The Activity Diagram is given below:



3.2 Algorithm

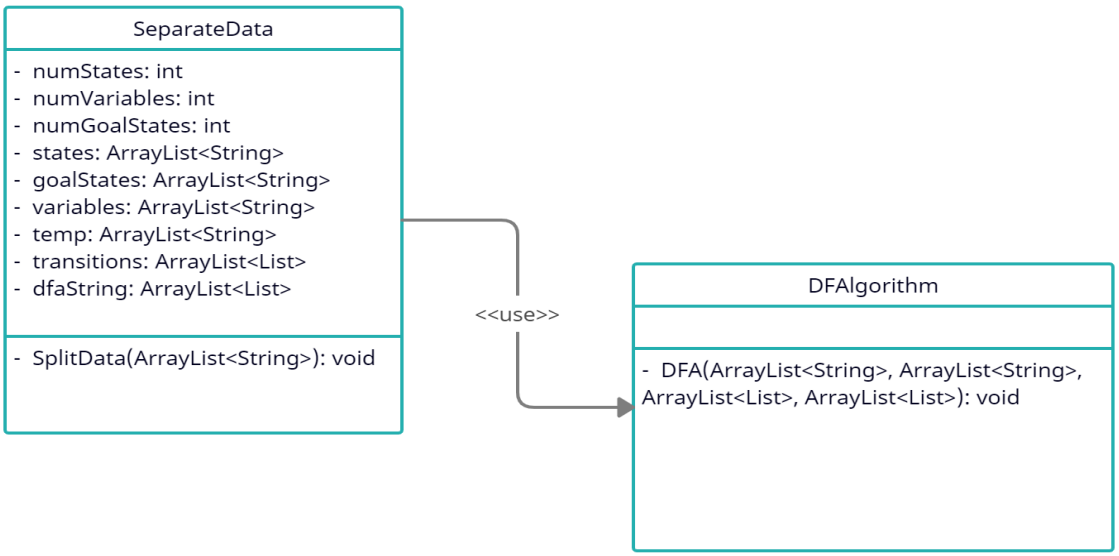
3.2.1 Component interfaces

In the Algorithm Package, there are two classes which are SeparateData & DFAlgorithm

Methods of SeparateData which are available from other components are:
public void SplitData(ArrayList<String> arrayList) – To separate the given ArrayList

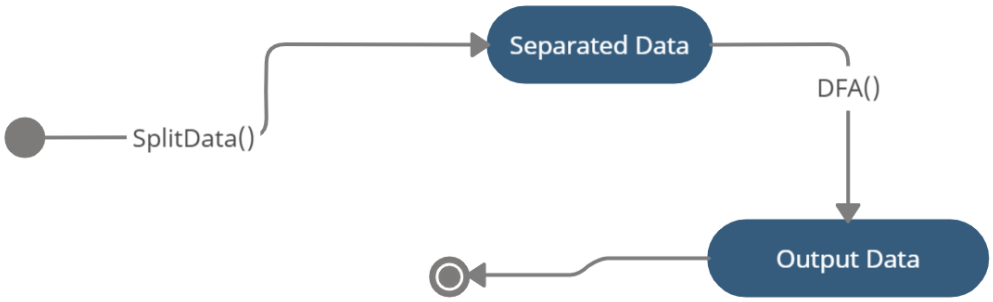
Methods of DFAlgorithm which are available from other components are:
public void DFA(ArrayList<String> states, ArrayList<String> goalState, ArrayList<List> transitions, ArrayList<List> dfaString) – To apply the DFA Algorithm

3.2.2 Component design description



3.2.3 Workflows and algorithms

The Activity Diagram is given below:



3.3 The Whole Diagrams

3.3.1 Component interfaces

Methods of main which are available from other components are:
public static void main(String[] args) – To run the code

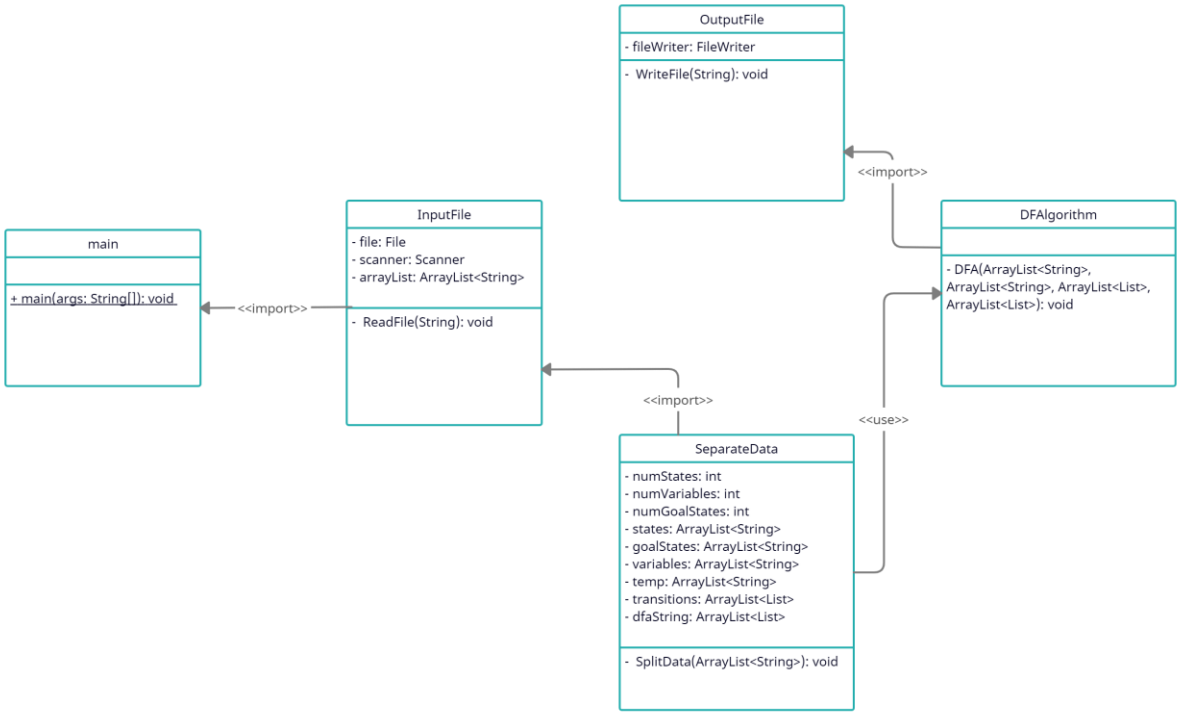
Methods of InputFile which are available from other components are:
public void ReadFile(String filePath) – To read the input .txt file

Methods of OutputFile which are available from other components are:
public void WriteFile(String routeValidation) – To write the output .txt file

Methods of SeparateData which are available from other components are:
public void SplitData(ArrayList<String> arrayList) – To separate the given ArrayList

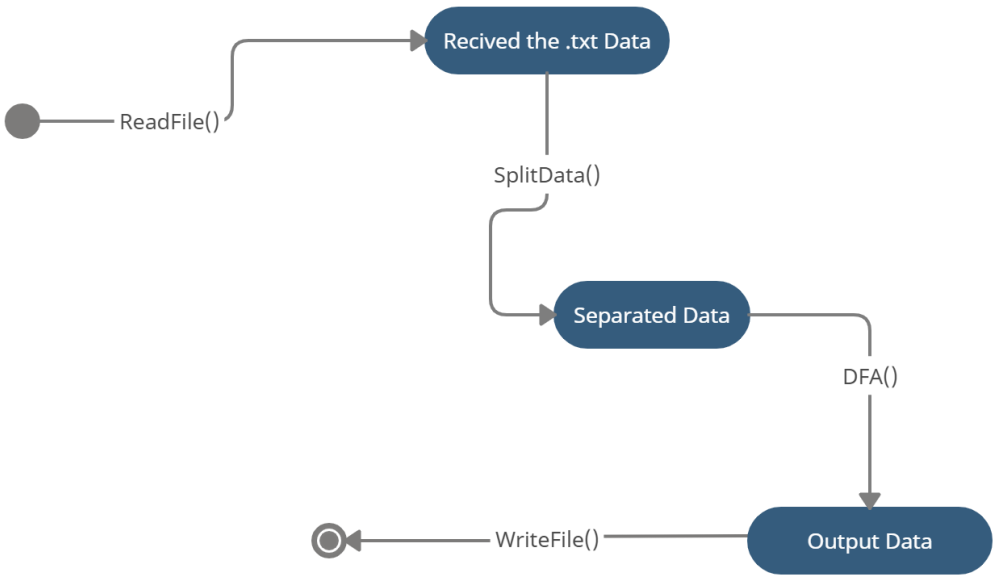
Methods of DFAlgorithm which are available from other components are:
public void DFA(ArrayList<String> states, ArrayList<String> goalState, ArrayList<List> transitions, ArrayList<List> dfaString) – To apply the DFA Algorithm

3.3.2 Component design description



3.3.3 Workflows and algorithms

The Activity Diagram is given below:



4 COTS Identification

COTS (commercial of the shelf) libraries used in DFA are the following:

- Java, JDK 8, https://www.oracle.com/java/technologies/downloads/