

<b>CS410 Project-3</b>		
<b>Doc # Turing-SDD</b>	<b>Version: 0.1</b>	<b>Page 1 / 10</b>

## REVISION HISTORY

<b>Date</b>	<b>Version</b>	<b>Description</b>	<b>Author</b>
24/12/2021	0.1	Design of the Sections	Muhammed Esad Simitcioglu

**TABLE OF CONTENTS**

Table of Contents

**REVISION HISTORY..... 1**

**1 Introduction .....3**

**2 Software design description.....4**

**2.1 TuringStateProps .....6**

**2.1.1 Component Interface.....6**

**2.1.2 Component Design Description.....6**

**2.2 TuringState .....7**

**2.2.1 Component Interface.....7**

**2.2.2 Component Design Description.....7**

**2.3 Tape .....8**

**2.3.1 Component Interface.....8**

**2.3.2 Component Design Description.....8**

**2.4 TuringPath .....9**

**2.4.1 Component Interface.....9**

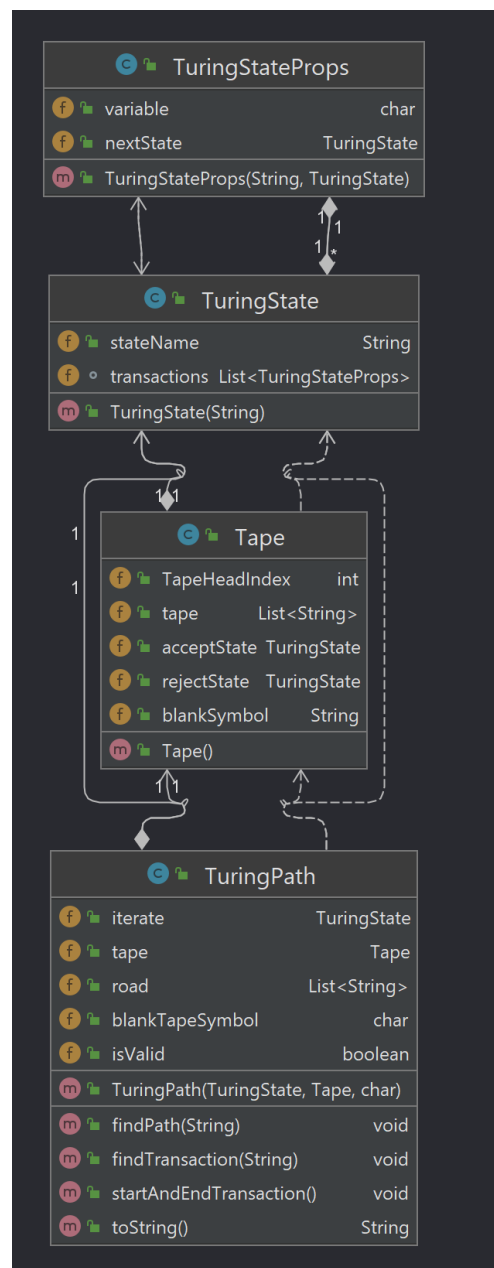
**2.4.2 Component Design Description.....9**

**3 COTS Identification ..... 10**

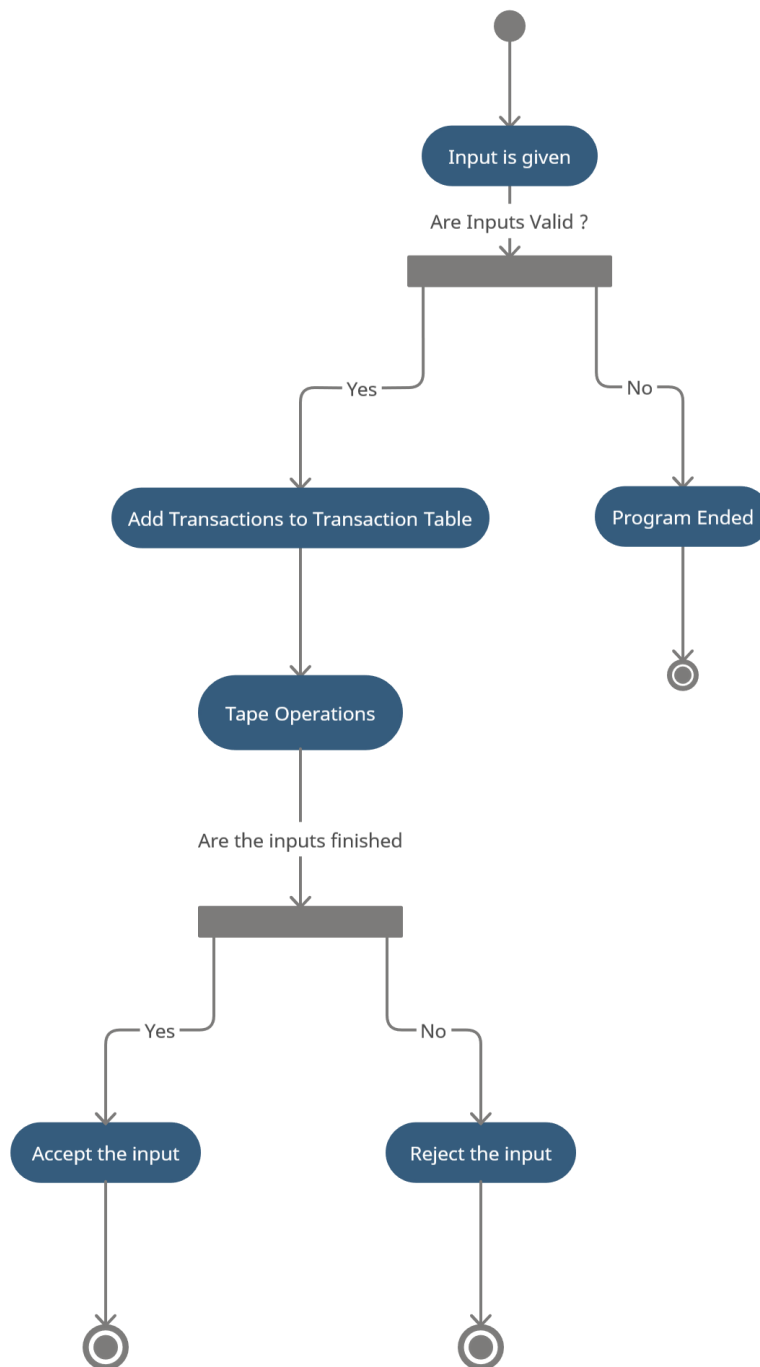
## 1 Introduction

This document describes the design of a Turing Machine. should be working exactly as a Turing Machine would do. Given a string the simulated Turing Machine should be able to tell if the string is accepted, rejected, or looped. Your program should output the information about the given string being accepted, rejected, or looped as well as which states it visited. Software Architecture overview

Class diagram is given below.



The Activity Diagram is given below:



## 2 Software design description

This program will not be desktop application and no need for database. So, in that case there won't be a UI or database implementation. The input and output format of the program should be like in the below.

CS410 Project-3		
Doc # Turing-SDD	Version: 0.1	Page 5 / 10

### Input:

1 (number of variables in input alphabet)  
 2 (number of variables in tape alphabet)  
 7 (number of states)  
 q1 q2 q3 q4 q5 qA qR (states)  
 q1 (start state)  
 qA (accept state)  
 qR (reject state)  
 b (blank symbol)  
 0 X b (the tape alphabet)  
 0 (the input alphabet)  
 q1 0 b R q2 (q1 state'inden tape de 0 okuyarak q2 state'ine gidiyor, tape e b yaziyor, tape de saga gidiyor.)  
 q1 b b R qR (q1 state inden tape de b okuyarak qR state ine gidiyor. tape e b yaziyor, tape de saga gidiyor.)  
 q1 X X R qR  
 q2 0 X R q3  
 q2 X X R q2  
 q2 b b R qA  
 q3 X X R q3  
 q3 0 0 R q4  
 q3 b b L q5  
 q4 X X R q4  
 q4 0 X R q3  
 q4 b b R qR  
 q5 0 0 L q5  
 q5 X X L q5  
 q5 b b R q2  
 00 (string to be detected)  
 000 (string to be detected)

### Output:

q1 q2 q3 q5 q5 q2 q2 qA (route taken)  
 Accepted  
 q1 q2 q3 q4 qR (route taken)  
 Rejected

The program will construct a List of "TuringState" object to store all the given states. After constructing and connecting "TuringStates", the program will create transaction table for each "TuringState". Each "TuringState" will know where to go when the appropriate variable arrives. Tape operations will done after each iteration if there is any operation needed (e.g. read or write) The name of each "TuringState" will be printed to reveal the route the program took. Finally, the program will print "Accepted" if the last "TuringState" is one of the target states. Otherwise, the program will print "Rejected".

CS410 Project-3		
Doc # Turing-SDD	Version: 0.1	Page 6 / 10

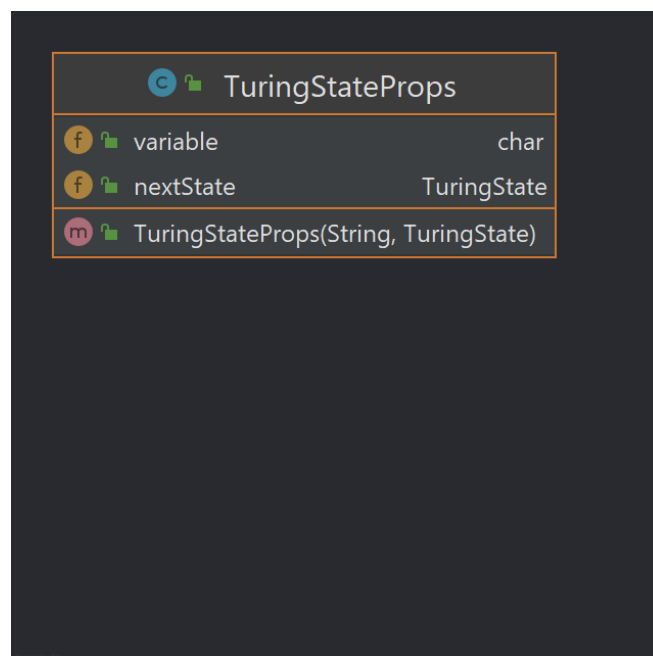
## 2.1 TuringStateProps

### 2.1.1 Component Interface

Methods of State which are available from other components are:

public TuringStateProps(String, TuringState) {} – Creates the State properties as specific variable and its next state.

### 2.1.2 Component Design Description



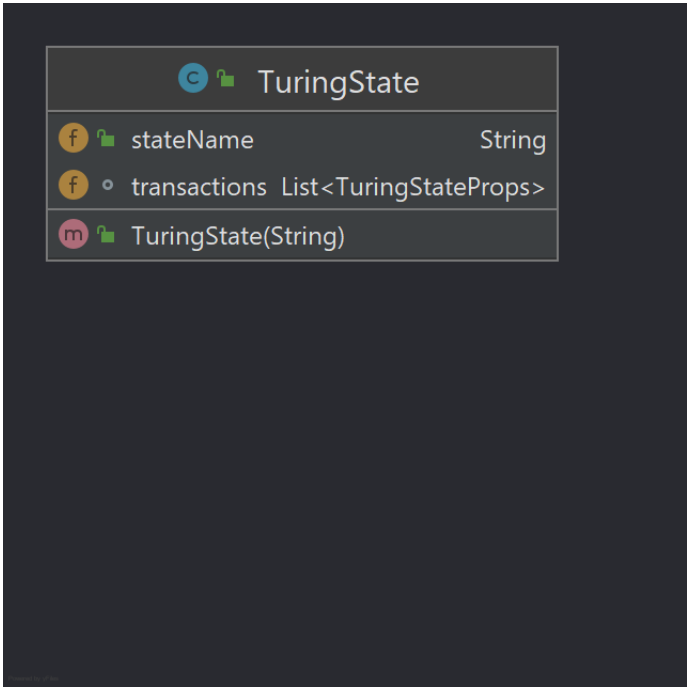
2.2 TuringState

2.2.1 Component Interface

Methods of TuringState which are available from other components are:

```
public TuringState (String) {}
```

2.2.2 Component Design Description



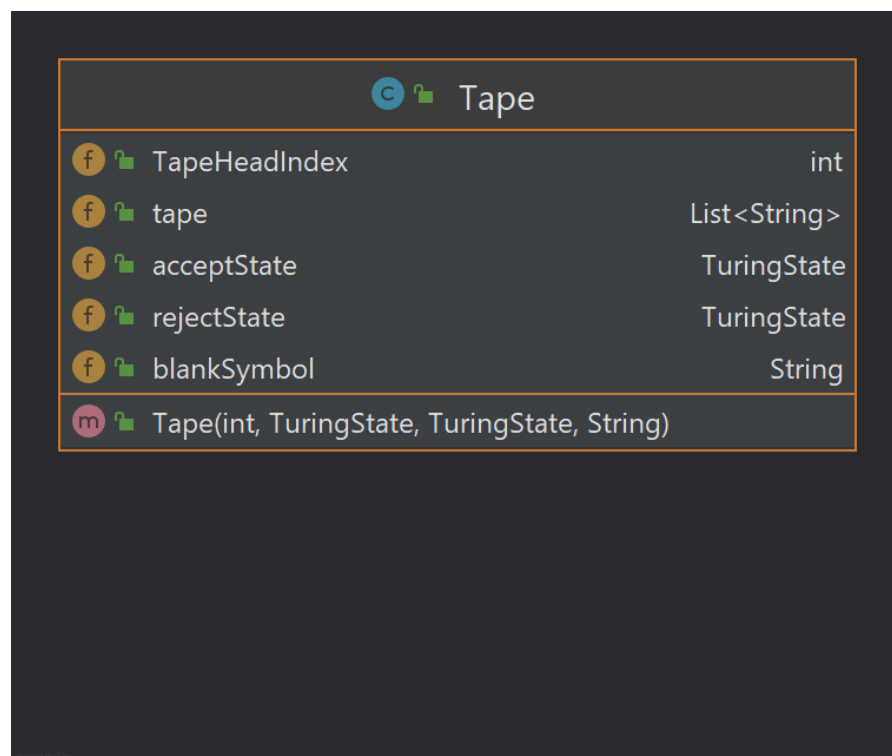
## 2.3 Tape

### 2.3.1 Component Interface

Methods of State which are available from other components are:

Public Tape(int, TuringState, TuringState, String) – {}

### 2.3.2 Component Design Description





## 2.4 TuringPath

### 2.4.1 Component Interface

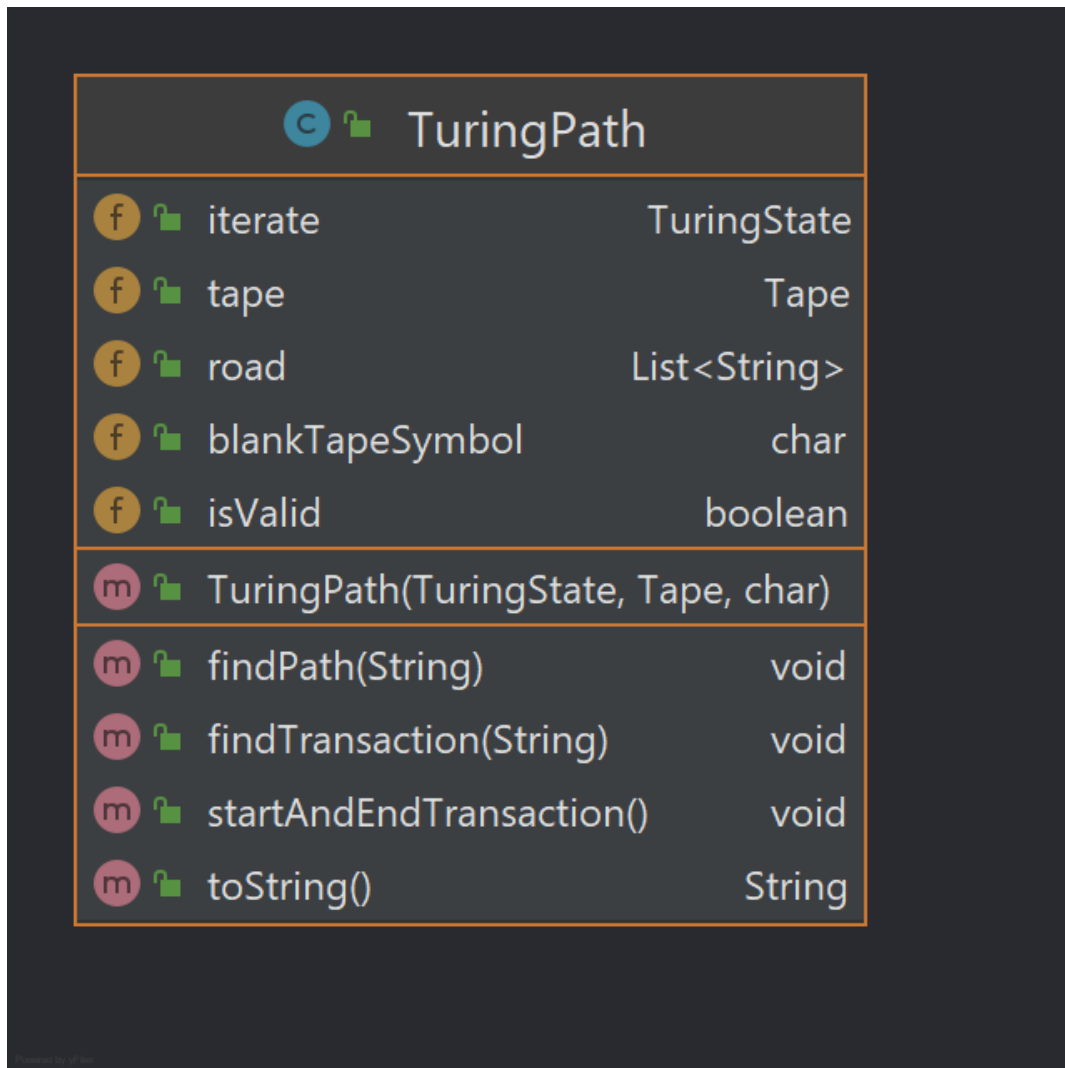
Methods of State which are available from other components are:

```

Public TuringPath(TuringState, Tape,char) - {}
Public findPath(String) - {} - define a path from beginning to its end TuringStates.
Public findTransaction(String) - {} - iterates the TuringStates except start and last state
Public startAndEndTransaction(String) - {} - iterates the TuringStates only start and last state
Public toString() - {}

```

### 2.4.2 Component Design Description



CS410 Project-3		
Doc # Turing-SDD	Version: 0.1	Page 10 / 10

### 3 COTS Identification

COTS (Commercial off the shell) libraries used :

Not Applicable