

**NANYANG TECHNOLOGICAL UNIVERSITY**



**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**SCSE18-0141**

**Visual Formulation of Smart Contracts on Blockchains**

**Sean Tan Jun Yu**

d      Bac      C      Sc      c

## Abstract

T c c a a b d a d -d  
d a c ac d E b c c a .O , S a  
C ac B d , a a d c c b c c a , c  
a c .G a ca b c a d d a a a d c d a d  
c, c a c a c ac .T S a C ac  
B d a a a d a c ac c d b c c a  
c c ab .W a a a c a d a a a a  
c ac b a d c a a a c ac .T  
d c a a d ,c a ac d a d a  
c c .

## Acknowledgements

T a A/P S a Sa a B c c a ca d da c a d  
.  
T a d a d a c a a d c a ,  
a d a c d a d da .

# Contents Page

<b>1. Introduction</b>	<b>6</b>
1.1 Background	6
1.2 Objectives and Scope	7
1.3 Expected Results	7
<b>2. Development and Implementation</b>	<b>9</b>
2.1 Setting up the Development Environment	9
<b>2.1.1 React and Electron</b>	<b>9</b>
<b>2.1.2 Additional packages</b>	<b>9</b>
<b>2.1.3 Remix IDE</b>	<b>10</b>
<b>2.1.4 Hardware</b>	<b>10</b>
<b>2.1.5 Dependencies</b>	<b>10</b>
2.2 Project Setup	11
2.3 Implementation	11
<b>2.3.1 Design</b>	<b>11</b>
2.3.1.1 Component Structure	12
2.3.1.2 Global State Management	12
2.3.1.2.1 Redux	13
2.3.1.2.2 Context	14
2.3.1.2.3 Custom Hooks	15
2.3.1.3 Backend Integration	16
2.3.1.3.1 Frontend Logic	17
2.3.1.3.2 Backend API	17
2.3.1.3.3 Authentication	18
<b>2.3.2 Backend Logic</b>	<b>22</b>
2.3.2.1 Database Setup	22
2.3.2.2 Business Logic	23
2.3.2.2.1 CRUD Operations	24
2.3.2.2.2 Authentication and Authorization	24
2.3.2.2.3 Security Measures	24
<b>3. Evaluation of Results</b>	<b>26</b>
3.1 Metrics and Analysis	26
3.2 Results	26
<b>4. Conclusions</b>	<b>30</b>
<b>5. Recommendations</b>	<b>31</b>
<b>6. End Section</b>	<b>32</b>
6.1 Appendix	32



## List of Figures

F	Pa
F 1.1: P c Sc d	11
F 2.1: Sc C c Pa	12
F 2.2: Sc G ba S a Tab	13
F 2.3: Sc a	14
F 2.4: Sc a	15
F 2.5: I a S a Tab a	16
F 2.6: G ba S a Tab C c Pa a C d F 2.4	16
F 2.7: E a P c a B d Tab	17
Tab 2.1: M da a a a c a d a d	18-21
F 2.8: E a d c d d c c	23
Tab 3.1: C a a c c c ac	26
F 3.1: C c F c	27
F 3.2: B d F c	28

# 1. Introduction

## 1.1 Background

Ab c c a a - c d b d da aba a - - -a  
c a a a ac a c ab a a d ac a d  
a ac a d ac .E c c c a a 2008, a  
a a a b d a a ac d c a.  
D a a c a dc c a a, d da a  
a a c ca a ac .  
B c c a a a db a b d Sa  
Na a 2008.T b c c a b c ad d a  
c c d b d b d c a .[1]  
W d d b c c a , a a  
a dd c b c c a c d ac c a .  
a ca , a c , a a ,c ac a a ad .  
N a 15% a ca a c b c c a c .I  
2013,V a B , a a a c b B c c d ba ,b ca  
a d a a a d d a a ab b c c a .  
M a c B c c ,B b d c d  
b cb c c a ca dE .T a d c b a  
E ca c d a c a a c ac , c c .  
B ca b c c a c a c , a a d  
a ,a d c a d .A a ,b c c a  
c a c d a da d a d  
-d a .I d b c c a ,  
a a ca ca dS a C ac B d .  
A a c ac c c d a a d a a  
c ac a ca d a ac a a .T S a C ac B d  
a ca a E a a c ac ,a d

S d a a .W b a a ca a d c a  
d b c c a , b -d .

## 1.2 Ob c a d Sc

T b c c c a a a ca ac a a  
a b d a d -d c a a d d a c ac .  
A b c c a a d a c ac a c c , b  
ab d a c ac c d a c a d  
a a c .B ab ac c d a a c a  
b , b a c a b d b c c a ,  
c a c a a a d c c a d a  
c .  
T c c c d a d a c ac  
b a a c ac .B ca a ,  
c a a .I , ca b d d  
a a a a d b c c a .

## 1.3 E S

A , a a a ca ac a ca  
ca d -d .C a d c a T  
d c a a d d a c ac .H , a  
a da c d d a c ab c a d .  
F , ad c ,d a d  
a c ac c d , d a ca a c d  
.T a a c a M aMa c a ac  
a c ac a d a a a ,b d c c  
b c a ca .T a a S a C ac B d  
d c a Tab a B Da a V a a a d SAS da a c c a d  
b a a c .H , c E a d b c c a a a

, a c  
ab a d c a a a d  
.

a b c c a  
acc b - c ca



## 2. Development and Implementation

## 2.1 S a A d T U d

### 2.1.1 R a c a d E c

I d b d S a C ac B d , E c a d R ac a			
c .			
E c a c a a a d c a b ac			
a , a d ab d c a ac HTML a d			
Ja aSc . W a E c a ca c d Sac ,			
G H b a d Mc V a S d C d . [2]			
R ac a a - d a d db Fac b a			
b ac V a DOM. W a R ac			
a : Fac b , l a a , N , W a a , Sa c , Ub , T N Y			
T , CNN, D b , Da M , IMDB, V , a d R dd . [3]			
E c R ac B a a d a ac a c. I			
E c , R ac , R d , R ac R , W b ac a d R ac H L ad			
a da ca d (HMR). [4]			

### 2.1.2 Add a ac a

I add	, cc	, Ga ac	CLI a d	b3 ac a	a	d. S c,
S d c	, d c	\$ d	a c	ac c d	ac	
adab c d	. Ga ac	CLI	d	a	c b c c a	b a
ab c		a	ca	, a d W b3	d	ac
b c c a	d d	a c	ac a d ca	c	. Ga ac	CLI
	:// ca	:8545 b	d a	.		
Ma a-UI,	a R ac	b a	a	G	a a d	, a d
c a a		a ca	ac	.		

T S R ac Da a ba a a d d da a dd  
da a c a d db Ga ca U l ac .T ba  
a ad cd a a ca a a ad  
d a a da a ,a a b d ab  
c a .

### 2.1.3 R IDE

R IDE a c a a d a c ac b .  
W R IDE a ca a ba a a a c ac  
d c a a c a d c c a a a .

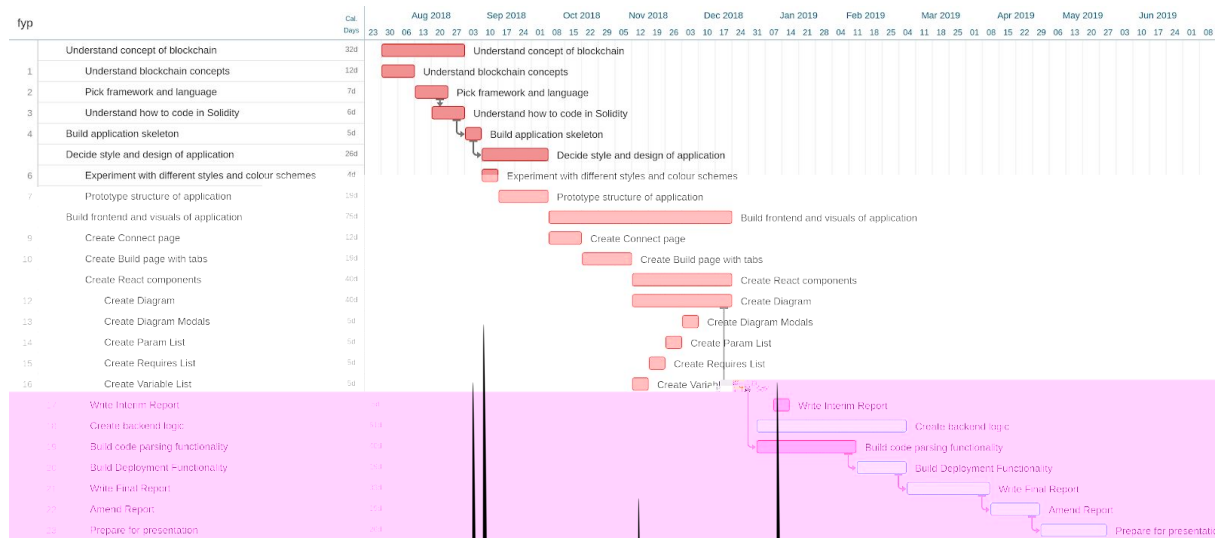
### 2.1.4 Ha d a

T a ca a b a a c 64 b L Ub 18.04,  
8GB RAM a da l (R) C (TM) 7-3770 CPU @ 3.40GH c .l  
a a b d a W d ac 8GB RAM.

### 2.1.5 D d c

A d 1 d d c ac a . c .

## 2.2 Project Schedule



F 1.1: Project Schedule

The project schedule is a Gantt chart showing the duration of various tasks. The tasks are listed on the left, and the timeline is shown on the right. The tasks are color-coded: red for tasks 1-16 and pink for tasks 17-22. The timeline starts in August 2018 and ends in June 2019. The tasks are: Understand concept of blockchain, Understand blockchain concepts, Pick framework and language, Understand how to code in Solidity, Build application skeleton, Decide style and design of application, Experiment with different styles and colour schemes, Prototype structure of application, Build frontend and visuals of application, Create Connect page, Create Build page with tabs, Create React components, Create Diagram, Create Diagram Modals, Create Param List, Create Requires List, Create Variable List, Write Interim Report, Create backend logic, Build code parsing functionality, Build Deployment Functionality, Write Final Report, Amend Report, and Prepare for presentation.

## 2.3 Interview

The interview was conducted with the project manager, who provided a detailed overview of the project schedule and the tasks involved.

### 2.3.1 Data

The data collected from the interview includes the project schedule, the tasks involved, and the duration of each task. The data is presented in a Gantt chart format, which allows for a visual representation of the project timeline and the dependencies between tasks.

a a a d a da .W d a a d b c  
c a c b a c a d ca c  
a c a d c a a d c .[5]  
T a c ac b d a 3 a c a d a :  
c c a , G ba S a Tab a d B d Tab .T B d a a  
c a G ba S a Tab a d B d ab a a bac b c  
c c a ,a a a b d b , c a  
d a c ac c d a d d .

### 2.3.1.1 C c Pa

T a c ac b d a d a a c c c .U  
add a d b c c a d c c .F a ,  
c c Ga ac CLI, add d b ca a d d b  
8545,ba d d a Ga ac CLI .C c a a d b c c a  
b c d a , B d a , c c a G ba S a  
Tab a d B d Tab .

Smart Contract Builder

Connect to blockchain

Protocol

HTTP ▾

:

// Blockchain Address

:

Blockchain Port

CONNECT

F 2.1: Sc C c Pa

### 2.3.1.2 G ba S a Tab

T G ba S a Tab d a a c ac a a d a  
c , a , a d c c a ab d c a a .

Smart Contract Builder

GLOBAL STATE

INITIAL STATE

+

Events

Event Name

ADD +

Entities

Entity Name

ADD +

BACK

LOAD

SAVE

GENERATE CODE

DEPLOY

F2.2: ScG ba S a Tab

2.3.1.2.1 E B

S d	a ab ac	EVM	c a .
A ca ca	b c b a d		RPC ac
a E	c .		
E a	ab b c ac .W	ca , ca	
a b d	a ac - a	ca da a c	
b c c a .T	a a ca d add	c ac , a	
c a d	b c c a , a d a	a a a b c acc b . [6]	
I S a C ac B d ,	add b	a a d	
c c add b .T	a a ab b a d	ca add a d	
c a a .T	ca b d	B d Tab .	
A ab a	b c d S c	2.3.1.3.3.	

Smart Contract Builder

GLOBAL STATE

INITIAL STATE

+

Events

Deposit

Variable Name

from

Variable Type

Address

Variable Name

id

Variable Type

String

Variable Name

value

Variable Type

Integer

+

Event Name

ADD +

Entities

Entity Name

ADD +

BACK

LOAD

SAVE

GENERATE CODE

DEPLOY

F2.3: Sc a

2.3.1.2.2 E B

I S a C ac B d , c c a

S d a a C.E ca a

c a c a a d ca c d .S a

G ba S a Tab a ca b d a

Tab .

a a c

a a b c

, a d ca d

d a a B d

Smart Contract Builder

GLOBAL STATE

INITIAL STATE

+

Events

Event Name

ADD +

Entities

Car

Variable Name

price

Variable Type

Integer

Variable Name

brand

Variable Type

String

Variable Name

seller

Variable Type

Address

+

Entity Name

ADD +

BACK

LOAD

SAVE

GENERATE CODE

DEPLOY

F 2.4: Sc a

2.3.1.2.3 C c Pa a B

T c c a a a b a d I a S a Tab, a d d  
b d G ba S a Tab. T a a c ac b  
a d c c a a a a .l c c d a a  
a a , c c a a d b d. F a ,  
c c b I a S a Tab F 2.4, C c Pa a  
b a a a F 2.5:

## Function Inputs

Variable Name

Initial Value

Variable Type

Integer

▼

Variable Name

Seller Name

Variable Type

String

▼

+

F 2.5: I a S a Tab a

Smart Contract Builder

GLOBAL STATE

INITIAL STATE

+

Events

Event Name

ADD +

Entities

Entity Name

ADD +

Constructor Parameters

Value of Initial Value

Value of Seller Name

BACK

LOAD

SAVE

GENERATE CODE

DEPLOY

F 2.6: G ba S a Tab C c Pa a C d F 2.4

### 2.3.1.3 B d Tab

T B d Tab c a c ac . Eac c  
c d B d Tab, a d c a a B d Tab.



T I a S a Tab c c c , add a c  
 ca b add d b c c a d a -d ca a .  
 T a 3 b a B d Tab: c b , c c a a d  
 ac a .

Smart Contract Builder

GLOBAL STATE

INITIAL STATE

PURCHASE

+

Function Inputs

Variable Name

Variable Type

Integer

+

Checking Phase

Variable 1

Comparator

is

Variable 2

Failure Message

+

Action Phase

Nodes

Assignment Node

Event Node

Transfer Node

Return Node

Conditional Node

Start

F 2.7: E a P c a B d Tab

2.3.1.3.1 F c I B

T c b a  
 c a a . C c  
 add a c a a .

4

c c a a ac  
 b add a

2.3.1.3.2 C c P a B

T c c a a d b a c . I  
 c d c d a , a ac a d b c c a b d

b c a .T a a b d a d b a  
c ac c d .

#### 2.3.1.3.3 Ac P a B

T ac a ac a c a b a a d c d , a d  
a da ada add da a .T c a d a  
a add da a a add a add  
da a .A da b d a d a a d c  
da a d c c d b , d b add d da a .  
T c c d ca d c .  
T ab da ac d :

N d	M da		
A N d	<div>New Assignment Node</div> <div>Variable NameAssigned Value</div> <div>CANCEL X DONE ✓</div>		

E N d

## New Event Node

Event to emit

Deposit

### Event Parameters

Value of from

Value of id

Value of value

CANCEL 

DONE 

E N d

## New Entity Node

Entity Name

chosen car

Entity Type

Car

### Event Parameters

Value of price

1000

Value of brand

"Mercedes"

Value of seller

message sender

CANCEL 

DONE 

T a N d

## New Transfer Node

Transfer to

Value

CANCEL 

DONE 

R N d	<h3>New Return Node</h3> <p>Return Variable</p> <hr/> <div> <span>CANCEL ✕</span> <span>DONE ✓</span> </div>
C d a N d	<h3>New Conditional Node</h3> <div> <div>Variable 1</div> <div>Variable 2</div> </div> <div> <div>Comparator</div> <div>is</div> </div> <div> <span>CANCEL ✕</span> <span>DONE ✓</span> </div>

Tab 2.1: M d a a a a c a d a d

N a a d d , a d

a a d d G ba S a

Tab. I Tab 2.1, a d F 2.3.

A a ab d a . T a d c a

a ab a d a ab d c a , a bac d c

a d ( a c 2.4). O a d

ad d ca a a ab a a d G ba S a Tab,

b d c d d a a d ab ac c c a ab

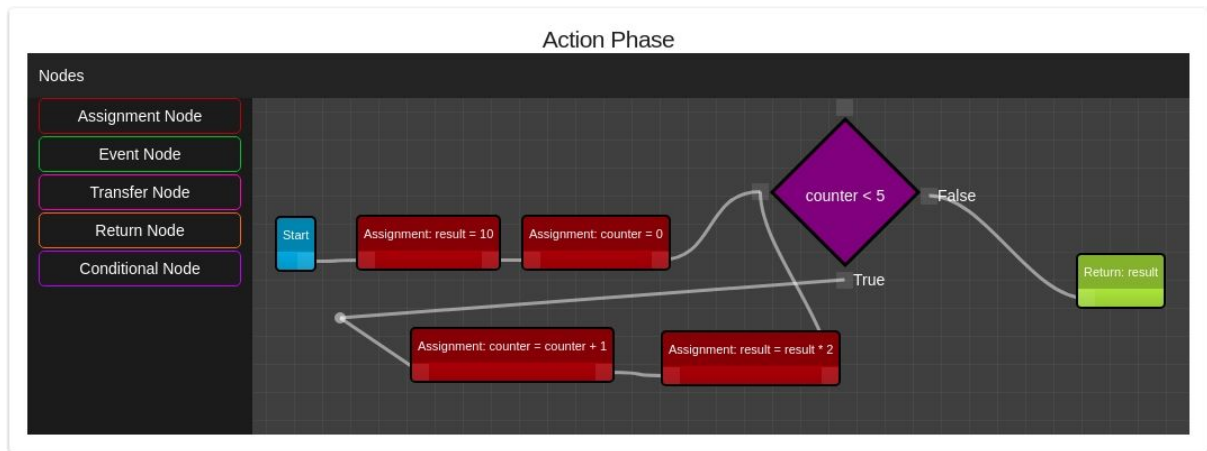
a c a b .

## 2.3.2 Bac d L c

T a 2 a c b bac d c: B dPa  
ca a d B dO c .T a a a a d ad  
c a ,a a a d c

### 2.3.2.1 B dPa

T B dPa b a da a a d a d d  
ac c .A a c B dPa ca a ac d ac a c  
Ac P a da a .A U da d a ac d ac da a ,a d  
B dPa a d ca d a c a db 2 d ,.  
T B dPa ac c d a da d , c  
d d c d ca a .T B dPa a da a 2 .  
T a ab , a d add a a d  
a a ab ab .l c c A ,N E a d Ta d  
a d d a ab a ab .l  
c d , B dPa a ac d c a d  
da a ,a d ac b .A ac d , a a a ab  
d a d a Va ab da d a c d  
a d a N d d.l a ab ab  
a ac a a ab a ca a d ,a  
c c b .l d a d , a da  
a ab c d ca a .  
W c ac d a d , c a 2 c c d b  
a da c d , a ac a a a d  
c d .l a ,a F 28, d c a c c  
da a a da da a c ac c d ad a  
a .A , d c a c a da a ,  
c d a d a a a a .



F 2.8: E a d c d d c c

T a a d a a d a a , b  
 a d a b a a d d a ab  
 a ab d a ad d . l a a d a ab  
 a d a c d a d a a , c d  
 c d b . A a d 2  
 ad 1, a ab a a a  
 a ab a d a c d .  
 S c a 2 a a a ab a d a c d  
 a d d ac a d ac c , a a a  
 c  $O(2) = O()$  b d da a . T  
 a b d da a a a c c d  
 a d .

### 2.3.2.2 B dO

T B dO R ac c b a c d a d  
 d b c c a . l d c c B d b  
 a b B d a . l a d a a d c d a d  
 l a S a T a b a d B d T a b , a d a a ac ca c c S d  
 a c ac . l b3 a d a c ac . l a  
 a d, b d . l d cc , a ac a

a c ac b .T a ac a ca b d c c  
a a ac b c c a .

#### 2.3.2.2.1 C d G a

Eac d c ca b b d , c c d ac  
c a b .A d 2 a c a ca  
d c ac a d c .  
B d c a a c ac , a ab ac  
a b a c c .T a a a  
a a a a c ac .  
W a c d b c c d , b a d a d d  
S d a d\_c ac d c .T a c a d c b d  
a S c 2.3.2.2.3.

#### 2.3.2.2.2 C ac C a a d D

B ca c b a a bac d a c a d  
d d , cc a b d bac d, a a  
a c .B cR d a d cMa ca , a  
ab d a a ad c d c d  
ac adab a ca b a ac (ABI).T C ac A ca B a  
I ac (ABI) a da d a ac c ac E  
c , b d b c c a a d c ac - c ac ac .  
[7] A c a , b3 b d d c ac b c c a .

#### 2.3.2.2.3 Sa a d L ad M c a

T S a C ac B d ab a c a d ad  
a a a .T ac d E c add , adF a d F  
c d ( ).T a d a d\_da a  
d c .T add c ad a a d\_da ad c a d



a .W	a b	c c d,a	a a
a a	.A	a a ,	a
B d c , a	a d c	a l a S a	Tab a d
B d Tab , b	d a d d d	a JSON	
a ca d c .W	ad b	c c d,a	a a
c c JSON	ad. T JSON	b ad a d a	d, a d
a B d c	b d	c	JSON .

3. Evaluation of Results

3.1 M d a a

T a a c a d c c a ca ,  
a a c d d a d c a c ac .  
Ga a a a a c a a a a  
c c a a .E a a a a E ,b  
a a ac , a a c ac, a ICO a a a .  
Ga a d cac a a a d b ad  
d c a a . [8]  
W ba da a aS d a c ac ca S d  
d c a [9] a d d d S a C ac B d ac  
a c a .W a b a a d a d a c ac  
R IDE, c a , c a a  
S d c ac a b .R d a a  
b , c d a ac a .F b c ac , d a  
Ja a c VM a d.

3.2 R

T a a :

	O _a c .  D c a	S a C ac B d	P a c D c
D c	400000 a	422800 a	-5.7%
C c c	440866 a	463667 a	-5.17%
B d c	63208 a	63543 a	-0.53%
W d a c			-
A c d d c			-

Tab 3.1: C a a c c c ac

T c d \_a c . a d c d a d a c ac  
b d ca b d A dc 3 a d 4 c . F 3.1 a d 3.2  
B d Tab c c a d b d c c .

Function Inputs

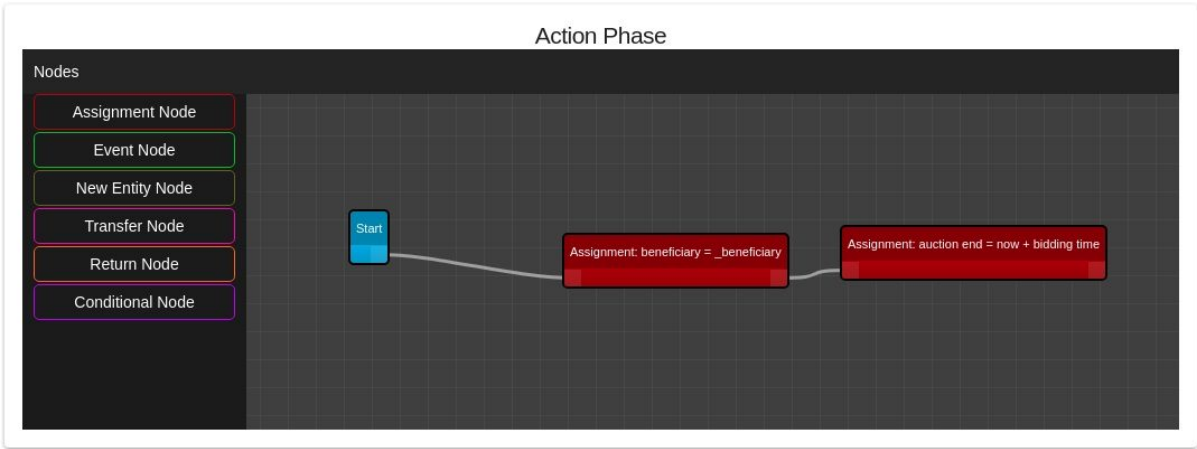
Variable Name	Variable Type
<input type="text" value="_beneficiary"/>	<input type="text" value="Address"/>
Variable Name	Variable Type
<input type="text" value="bidding time"/>	<input type="text" value="Integer"/>

+

Checking Phase

Variable 1	Comparator	Variable 2	Failure Message
<input type="text"/>	<input type="text" value="is"/>	<input type="text"/>	<input type="text"/>

+



F 3.1: C c F c

Checking Phase

Variable 1

now

Comparator

less than or e...

Variable 2

auction end

Failure Message

Auction already eni

Variable 1

message value

Comparator

greater than

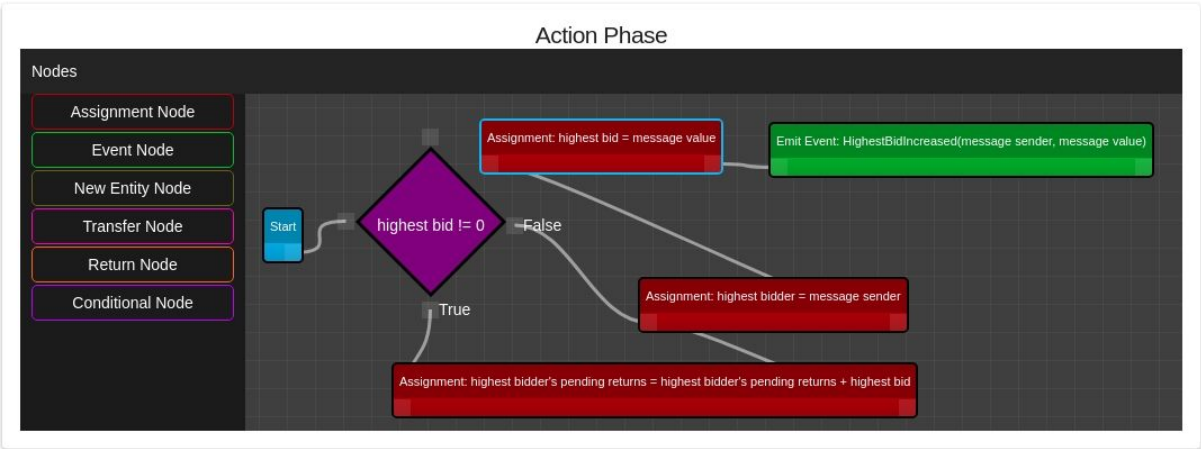
Variable 2

highest bid

Failure Message

There already is a l

+



F

3.2: B d F c

U

a , R IDE a c a a

ab b a a a

d a a d a c d d c , a d a a b

c c ac a d a d c ac .

D

c a ba d c d da a b c c a . T a 4

c a a ac c :

ba c a a ac (21000 a )

c a c ac d (32000 a )

c b da a c d a a ac .

c - b da a c d a a ac .

E

c c a ba d c c a a a c a

c d a a a ac .

F

ab 3.1, ca a d c 5.7% a d

c

ac , c c c c 5.17%

a d c ac , a d c c b d c 0.53% . T

a c d d a ca c c ac , b

b ca a a a , c ac . F a a c ac , 5%

d c b a ca . H , c c d  
a ca a d, d a acc ab a d d c  
ca . T a a d c d a a ca b d  
a c c a d a a .

## 4. Conclusions

I c, a a c a a , - d a ca a a d  
b d a d -d c a a c ac a dd  
a .T a c a c a a ac a a c a a d ,  
bac d c d a b c d a  
a a d , c a a ab a d  
da a .  
F , ca a a c , S a C ac  
B d ab a ad c a c ac , b  
d c a d ac a a - c d , ca c  
c c a dd c ac , a Tab 3.1.T a  
a a a c ac d c a , a  
a a c d a a c a c ac .M  
a b d a d a a c ac  
da a b ad c ca d a .F ,  
S a C ac B d a c a d a  
a d c d a a d c .  
N , S a C ac B d c d ac a  
a c , b ad c a d b c c a .  
D a S a C ac B d a a a  
c ac , a d a a .U a d  
b c c a b d a c ca - ca S a C ac  
B d a a a a c ac a .

## 5. Recommendations

T c ab , c a ab ac  
c d a a a c ca b d b d c a  
a a a a a c a ca d c b  
a a c ac dd ad c.T da a ca b  
a d d c , c a a a .  
F , ac d b c c a a d E ,  
a d a d c a , a ca a b c da d  
S d d c da .C a d d  
a ca da S d a da d .I a b b a S d  
b c b a E a d c d V , a  
a c ac a a .[10] F d b d a  
a ca d .C a b a c ac  
a a V ca b d c d a a b a d  
.I add ,c a b dB c c a  
H d Fab c ca a b d c d a a a c a a d c ca  
b ac d.

## 6. End Section

### 6.1 A      d c

```
"devDependencies": {
  "babel-core": "^6.26.3",
  "babel-eslint": "^8.2.6",
  "babel-jest": "^23.4.2",
  "babel-loader": "^7.1.5",
  "babel-plugin-add-module-exports": "^0.2.1",
  "babel-plugin-dev-expression": "^0.2.1",
  "babel-plugin-flow-runtime": "^0.17.0",
  "babel-plugin-transform-class-properties": "^6.24.1",
  "babel-plugin-transform-es2015-classes": "^6.24.1",
  "babel-preset-env": "^1.7.0",
  "babel-preset-react": "^6.24.1",
  "babel-preset-react-optimize": "^1.0.1",
  "babel-preset-stage-0": "^6.24.1",
  "babel-register": "^6.26.0",
  "chalk": "^2.4.1",
  "concurrently": "^3.6.1",
  "cross-env": "^5.2.0",
  "cross-spawn": "^6.0.5",
  "css-loader": "^1.0.0",
  "detect-port": "^1.2.3",
  "electron": "^2.0.6",
  "electron-builder": "^20.26.0",
  "electron-devtools-installer": "^2.2.4",
  "electron-rebuild": "^1.8.2",
  "enzyme": "^3.3.0",
  "enzyme-adapter-react-16": "^1.1.1",
```



```
"enzyme-to-json": "^3.3.4",
"eslint": "^5.2.0",
"eslint-config-airbnb": "^17.0.0",
"eslint-config-prettier": "^2.9.0",
"eslint-formatter-pretty": "^1.3.0",
"eslint-import-resolver-webpack": "^0.10.1",
"eslint-plugin-compat": "^2.5.1",
"eslint-plugin-flowtype": "^2.50.0",
"eslint-plugin-import": "^2.13.0",
"eslint-plugin-jest": "^21.18.0",
"eslint-plugin-jsx-a11y": "6.1.1",
"eslint-plugin-promise": "^3.8.0",
"eslint-plugin-react": "^7.10.0",
"express": "^4.16.3",
"fbjs-scripts": "^0.8.3",
"file-loader": "^1.1.11",
"flow-bin": "^0.77.0",
"flow-runtime": "^0.17.0",
"flow-typed": "^2.5.1",
"husky": "^0.14.3",
"identity-obj-proxy": "^3.0.0",
"jest": "^23.4.2",
"lint-staged": "^7.2.0",
"mini-css-extract-plugin": "^0.4.1",
"minimist": "^1.2.0",
"node-sass": "^4.9.2",
"npm-logical-tree": "^1.2.1",
"optimize-css-assets-webpack-plugin": "^5.0.0",
"prettier": "^1.14.0",
"react-test-renderer": "^16.4.1",
"redux-logger": "^3.0.6",
```

```
"rimraf": "^2.6.2",
"sass-loader": "^7.0.3",
"sinon": "^6.1.4",
"spectron": "^3.8.0",
"storm-react-diagrams": "^5.2.1",
"style-loader": "^0.21.0",
"stylelint": "^9.4.0",
"stylelint-config-standard": "^18.2.0",
"uglifyjs-webpack-plugin": "1.2.7",
"url-loader": "^1.0.1",
"webpack": "^4.16.3",
"webpack-bundle-analyzer": "^2.13.1",
"webpack-cli": "^3.1.0",
"webpack-dev-server": "^3.1.5",
"webpack-merge": "^4.1.3",
"yarn": "^1.9.2"
},
"dependencies": {
  "@fortawesome/fontawesome-free": "^5.2.0",
  "@material-ui/core": "^3.0.1",
  "@material-ui/icons": "^3.0.1",
  "devtron": "^1.4.0",
  "electron-debug": "^2.0.0",
  "history": "^4.7.2",
  "react": "^16.4.1",
  "react-dom": "^16.4.1",
  "react-hot-loader": "^4.3.4",
  "react-redux": "^5.0.7",
  "react-router": "^4.3.1",
  "react-router-dom": "^4.3.1",
  "react-router-redux": "^5.0.0-alpha.6",
```

```
"redux": "^4.0.0",
"redux-thunk": "^2.3.0",
"source-map-support": "^0.5.6",
"typeface-roboto": "0.0.54",
"web3": "^1.0.0-beta.35"
},
"devEngines": {
  "node": ">=7.x",
  "npm": ">=4.x",
  "yarn": ">=0.21.3"
}
```

A d 1: ac a . d d c

```
pragma solidity ^0.5.4;

contract {contract name} {
  /// variable declarations
  {variable_type} public {variable_name}
  uint public value;
  address public seller;
  address public buyer;

  constructor() public payable {
    {constructor code}
  }

  event {event_name} ({parameter_type} {parameter_name})
  event Aborted();
  event PurchaseConfirmed();
  event ItemReceived();
}
```

```

    /// returns does not appear if there is no return statement
function {function_name}({function_params}) public returns (int) {
    {require_statements}
    {function_code}
}

function abort() public
{
    emit Aborted();
    seller.transfer(address(this).balance);
}

function confirmPurchase() public payable
{
    emit PurchaseConfirmed();
    buyer = msg.sender;
}

function confirmReceived() public
{
    emit ItemReceived();
    buyer.transfer(value);
    seller.transfer(address(this).balance);
}
}

```

A d 2: S c a S d a c ac

```

pragma solidity >=0.4.22 <0.6.0;

contract SimpleAuction {
    // Parameters of the auction. Times are either
    // absolute unix timestamps (seconds since 1970-01-01)
    // or time periods in seconds.

```

```

address payable public beneficiary;
uint public auctionEndTime;

// Current state of the auction.
address public highestBidder;
uint public highestBid;

// Allowed withdrawals of previous bids
mapping(address => uint) pendingReturns;

// Set to true at the end, disallows any change.
// By default initialized to `false`.
bool ended;

// Events that will be emitted on changes.
event HighestBidIncreased(address bidder, uint amount);
event AuctionEnded(address winner, uint amount);

// The following is a so-called natspec comment,
// recognizable by the three slashes.
// It will be shown when the user is asked to
// confirm a transaction.

/// Create a simple auction with `_biddingTime`
/// seconds bidding time on behalf of the
/// beneficiary address `_beneficiary`.
constructor(
    uint _biddingTime,
    address payable _beneficiary
) public {
    beneficiary = _beneficiary;
    auctionEndTime = now + _biddingTime;
}

/// Bid on the auction with the value sent
/// together with this transaction.
/// The value will only be refunded if the
/// auction is not won.
function bid() public payable {
    // No arguments are necessary, all
    // information is already part of

```

```

// the transaction. The keyword payable
// is required for the function to
// be able to receive Ether.

// Revert the call if the bidding
// period is over.
require(
    now <= auctionEndTime,
    "Auction already ended."
);

// If the bid is not higher, send the
// money back.
require(
    msg.value > highestBid,
    "There already is a higher bid."
);

if (highestBid != 0) {
    // Sending back the money by simply using
    // highestBidder.send(highestBid) is a security risk
    // because it could execute an untrusted contract.
    // It is always safer to let the recipients
    // withdraw their money themselves.
    pendingReturns[highestBidder] += highestBid;
}
highestBidder = msg.sender;
highestBid = msg.value;
emit HighestBidIncreased(msg.sender, msg.value);
}

/// Withdraw a bid that was overbid.
function withdraw() public returns (bool) {
    uint amount = pendingReturns[msg.sender];
    if (amount > 0) {
        // It is important to set this to zero because the
recipient
        // can call this function again as part of the
receiving call
        // before `send` returns.
        pendingReturns[msg.sender] = 0;
    }
}

```

```

        if (!msg.sender.send(amount)) {
            // No need to call throw here, just reset the
amount owing
            pendingReturns[msg.sender] = amount;
            return false;
        }
    }
    return true;
}

/// End the auction and send the highest bid
/// to the beneficiary.
function auctionEnd() public {
    // It is a good guideline to structure functions that
interact
    // with other contracts (i.e. they call functions or send
Ether)
    // into three phases:
    // 1. checking conditions
    // 2. performing actions (potentially changing conditions)
    // 3. interacting with other contracts
    // If these phases are mixed up, the other contract could
call
    // back into the current contract and modify the state or
cause
    // effects (ether payout) to be performed multiple times.
    // If functions called internally include interaction with
external
    // contracts, they also have to be considered interaction
with
    // external contracts.

    // 1. Conditions
    require(now >= auctionEndTime, "Auction not yet ended.");
    require(!ended, "auctionEnd has already been called.");

    // 2. Effects
    ended = true;
    emit AuctionEnded(highestBidder, highestBid);
}

```

```

        // 3. Interaction
        beneficiary.transfer(highestBid);
    }
}

```

A d 3: \_a c . S d d c a

```

pragma solidity ^0.5.4;
contract Code {
    bool public ended;
    uint public amount;
    mapping(address => uint) pending_returns;
    address payable public highest_bidder;
    uint public highest_bid;
    address payable public beneficiary;
    uint public auction_end;
    event HighestBidIncreased (address payable bidder, uint amount);
    event AuctionEnded (address payable winner, uint amount);
    constructor(address payable _beneficiary, uint bidding_time)
    public payable {
        beneficiary = _beneficiary;
        auction_end = now + bidding_time;
    }
    function bid() public payable {
        require(now <= auction_end, "Auction already ended.");
        require(msg.value > highest_bid, "There already is a higher
        bid.");
        if (highest_bid != 0) {
            pending_returns[highest_bidder] = pending_returns[highest_bidder]
            + highest_bid;
        }
        highest_bidder = msg.sender;
        highest_bid = msg.value;
        emit HighestBidIncreased(msg.sender, msg.value);
    }
    function withdraw() public payable {
        amount = pending_returns[msg.sender];
        if (amount > 0) {
            pending_returns[msg.sender] = 0;
            msg.sender.transfer(amount);
        }
    }
}

```



```

}
function auctionEnd() public payable {
    require(now >= auction_end, "Auction not yet ended.");
    require(ended == false, "auctionEnd has already been called.");
    ended = true;
    emit AuctionEnded(highest_bidder, highest_bid);
    beneficiary.transfer(highest_bid);
}
}

```

A d 4: C d a d b S a C ac B d

## 6.2 R c

- [1] F b .c . (2019). A V B H O B c c a T c E  
S d R ad. [ ] A a ab a :  
:// . b .c / /b a d a /2018/02/16/a- -b - - -b c c  
a - c - - d- ad/#79c719 97bc4 [Acc d 18 F b. 2019].
- [2] H -T G . (2019). W a A E c A , a d W Ha T B c S  
C ? . [ ] A a ab a :  
:// . .c /330493/ a-a - c -a -a d- - a - -b c  
- -c / [Acc d 18 F b. 2019].
- [3] M d . (2019). Ad a a D M d W b a R ac . .  
[ ] A a ab a :  
:// d .c /@ a a a d/ad a a - -d - d - b-a  
- - ac - -8504c571db71 [Acc d 18 F b. 2019].
- [4] G H b. (2019). c - ac-b a / c - ac-b a . [ ]  
A a ab a : :// b.c / c - ac-b a / c - ac-b a  
[Acc d 18 F b. 2019].
- [5] Ca , J. (2019). W b d c : c a  
c b d . [ ] T N W b. A a ab a :  
:// b.c /dd/2015/04/07/ - -c a - - - -c -  
- b-d / [Acc d 18 F b. 2019]. 7
- [6] S d . ad d c . . (2019). C ac S d 0.5.3 d c a .

[ ] A a ab a: :// d . ad d c . / / 0.5.3/c ac .

[Acc d 18 F b. 2019].

7

[7] S d . ad d c . . (2019). C ac ABI S c ca S d 0.5.3

d c a . [ ] A a ab a:

:// d . ad d c . / / 0.5.3/ab- c. [Acc d 18 F b. 2019].

[8] B c .c . (2019). [ ] A a ab a:

://b c .c / d / - a - -b - - d / [Acc d 18 F b.

2019].

7

[9] S d . ad d c . . (2019). S d b E a S d 0.5.5

d c a . [ ] A a ab a:

:// d . ad d c . / / a / d -b - a . # - -a c

[Acc d 1 Ma . 2019].

[10] B c .c . (2019). [ ] A a ab a:

://b c .c / d / / [Acc d 18 F b. 2019].