

Configuration Manual

Funds Routing System: an accountable fund allocation mechanism

MSc in Fintech **Research Project**

Rodolfo José Monsberger Student ID: x18175457

School of Computing National College of Ireland

Supervisor: Victor del Rosal

National College of Ireland

MSc Project Submission Sheet



Year: 2020

School of Computing

Rodolfo José Monsberger Student Name:

Student ID: X18175457

Programme: MSc. in Fintech

Module: MSc. Research Project

Supervisor: Victor del Rosal **Submission Due** 17th August 2020 Date:

Project Title: Funds Routing System: an accountable fund allocation mechanism

Word Count: 482 Page Count: 10

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:

17th August 2020

Date:

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple	
copies)	
Attach a Moodle submission receipt of the online project	
submission, to each project (including multiple copies).	
You must ensure that you retain a HARD COPY of the project, both	
for your own reference and in case a project is lost or mislaid. It is not	
sufficient to keep a copy on computer.	

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office	Use	Only

Office Use Unly	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Rodolfo José Monsberger x18175457

1 Introduction

The FRS prototype has been developed in solidity language of Remix IDE. This first version focuses on the back-end solution in order to test the basic functions of the design before proceeding in a second stage with a more comprehensive development which will include the front-end functionalities.

2 Remix IDE structure

Remix is a Solidity IDE used to write, compile and debug Solidity code. It has three environments for executing the transactions:

- JavaScript VM: a sandbox implemented with JavaScript to emulate a real blockchain.
- Injected Web3: a web3 able to inject Mist and Metamask
- Web3 Provider: a remote node with geth, parity or any Ethereum client. Can be used to connect to the real network

FRS prototype is designed to work on the sandbox version of JavaScript VM.

Remix IDE can be accessed from a web browser without any special installation. Open the Remix IDE with following link:

https://remix.ethereum.org/

Various panels compose Remix IDE. The "main panel" is where the code is written. Icon Panel contains most plug-in functions. The "Side Panel" is where compiling and running of the code takes place. The "terminal panel" is where results of interaction are shown.



Source: https://remix-ide.readthedocs.io/en/latest

3 Instructions for code uploading

Expand the side panel to work more comfortably.

	aformación sobre la re 🗙 🛛 🐢 Remix - Ethereum IDE 🔿	Remix - Ethereum IDE 🗙	E Remix-IDE Layout — Re 🗙	😫 Traductor de DeepL - 🗆 🗙 📔 🥔	Ethereum Unit Convert	: 🖉 Ethereum Uni	t Converte 🗙 📔 🕂	-	×
\leftarrow	→ C	&evmVersion=null&version=soljsc	n-v0.5.17+commit.d19bba13.j	js		☆ 🔘	🖭 🕝 🕈 📘	C Section 1	🗯 🖪 🗄
A A	licaciones Ġ Google 📑 Iniciar sesión en la 🌰	NCI Library 🌇 NCIRL moodle 🗵	Lingue K wordreference	🔚 leo 🛭 😝 Traductor de DeepL 🛔	Longman Dictionar	🗱 Macmillan		>>	Otros marcadores
۲	FILE EXPLORERS	Q Q Home ×							1 tabs =
ළු	1_Storage.sol 2_Owner.sol		4						
S	3_Ballot.sol • tests			Learn more Use previous	is version				
	FK5.30I								
ý	Expand the si	Environments	W Lyper	Featured Pl 8 PPELNE	lugins MYTHX	SV Sourcify	資 DEBUGGER	MORE	:
		File	1	Resources					
		New File	1	Documentation					
		Connect to Localhost		Medium Posts					
		110007-00014	i i	Tutorials					
		Gist GitHub Swarm	Ipfs https Resolver	-engine					
		🗧 🛇 0 🗌 listen on network	Q Search with transact	ion hash or address					
		 Use exports/.register(key 	<pre>, obj)/.remove(key)/.clear()</pre>	to register and reuse object ac	cross script execution	is.			^
¢		>							
-		E 🔒 📀 오	🖷 🦷 📦				? ∧ a.	• ¢0) ♥ (0	10:21

Create a "new file" by clicking on the button with the "+" icon in the top left corner of the side panel.

← -	→ C 🔒 remix.ethereum.org/#op	timize=false&evmVersion=null&version=soljson-v0.6.6+commit.6c089d02.js		☆		R2 🕝	9 🛛		* B
🔛 Ap	olicaciones 💪 Google 👫 Iniciar sesión	en Ia 🜰 NCI Library 🌇 NCIRL moodle 🌃 Lingue 🕷 wordreference 🔣 Ieo	😂 Traductor de DeepL 🛛 🇯 Longman Dictionar	🚧 Macmillan				» 📙	Otros marcado
	FILE EXPLORERS	Q Q ♣ Home ×							114
که کا د ک	Listora 2,Ownersol 3,Ballotsol + tests FRS.sol	Create new file. Name it "FRS"	Use previous version						
\sim									
\$		Environments	Featured Plugins						
¢.		Soudity Vyper	B Xr PIPELINE MYTHX	S V SOURCIFY		* DEBUGGE	R	MORI	
		File New File Open Files	Resources Documentation Gitter.channel						
		Connect to Localhost	Medium Posts						
		\bullet 0 \Box listen on network \bullet Search with transaction has	h or address						
-		 ethers.is imacross resix (run resix.help() for more info) Executing common command to interact with the Remix interfactscript. Use exports/.register(key, obj)/.remove(key)/.clear() to reg 	e (see list of commands above). Note that the gister and reuse object across script execution	se commands can ns.	also b	e includ	ed and r	un from a	JavaScript

A new dialog pops up. Change the file name for "FRS.sol", then press OK.

FILE EXPLORERS	ଷ୍ ଷ୍ 🗣 Home ×		
•browser O O D	Create new file	×	
2_Owner.sol 3_Ballot.sol	File Name (e.g. Untitled.sol)		
 tests FRS.sol 	Intilled.so		
		OK Cancel	
	/		
	Environments	Featured Plugins	
	Soudity Vyper	S X SV PIPELINE MYTRK SOURCIFY	R DEBUGGER MORE
/	/		
Home X			
Create new file	×		
File Name (e.g Untitled.sol)			
FRS.sol	9		
	Cik Cancel		

Open the word document called "FRS code", copy its content and paste it in the mail panel. Secondly, go to the Icon Panel and click "Solidity compiler".

	FILE EXPLORERS	Q Q Home FRS.sol X
4	browser O O E 1_Storage.sol 2_Owner.sol 3_Builds.col	<pre>prages solidity 40.5.16; 3 contract FRS { /** F1 address 1/ /** F1 address</pre>
(% *>	+ tests	<pre>address public *r_wookcasy 9 /** folkens per Loom in WET */ 10 uint public constant WEI_Tokens = 1000 * (10**18); 11 /** Borrower address for tokens credited */ 13 mapping(address >> uint) public AddressTokens(redited;</pre>
\$£	2. Solidity Co	mpiler account of tokens debited */ -> uint) public AddressTokensDebited; coan file */ -> uint) public borrowerLoan;
		<pre>20 21 22 /** Structure of a loan */ 23 * struct Loan { 24 uint loanAmountFlATMeij // Amount of FIAT currency in WEI user took loan 25 uint CRMeij // Conversion rate in WEI 26 address loanAmeciver; // Address of Loan receiver-> borrower 27 string loanAmountFlATMei // Address of Loan 28 string loanAmountFlATMei // Address of Loan 29 string loanAmountFlATMei // Address of Loan 29 } 29 } 20 } 21 /** Nodifier to restrict access only to FI */ 30 modifier onlyF1 { 31 modifier onlyF1 { 32 modifier onlyF1 { 33 modifier onlyF1 { 34 modifier onlyF1 { 35 modifier onlyF1 { 36 modifier onlyF1 { 37 modifier onlyF1 { 38 modifier onlyF1 { 39 modifier onlyF1 { 39 modifier onlyF1 { 30 modifier onlyF1 { 31 modifier onlyF1 { 32 modifier onlyF1 { 33 modifier onlyF1 { 34 modifier onlyF1 { 34 modifier onlyF1 { 35 modifier onlyF1 { 35 modifier onlyF1 { 36 modifier onlyF1 { 37 modifier onlyF1 { 38 modifier onlyF1 { 39 modifier onlyF1 { 30 modifier onlyF1 { 30 modifier onlyF1 { 31 modifier onlyF1 { 32 modifier onlyF1 { 33 modifier onlyF1 { 34 modifier onlyF1 { 35 m</pre>

Proceed to compile contract.



4 Instruction for deploying and running the code

JavaScript VM provides several fake accounts with 100 ether each, which can be used to test the contract. First address will be selected for the Financial Institution and the second one for the borrower. Copy the first address to clipboard.



Paste address and deploy contract.

📑 Info	ormación sobre la reunión - Z 🗙 📔 🐢 Remix - Eth	ereum IDE	×	Remix - Ethereum IDE	× 🗉	Remix-IDE Layout	— Remix, Ether 🗙	+				-	- 0	\times
$\leftarrow \rightarrow$	C remix.ethereum.org/#optimize=	false&evm\	Version=null&ver	sion=soljson-v0.5.17	7+commit.d19bba1	3.js			\$	💿 🕱 💿	9 🛙		(* B	:
Aplic Aplic	caciones 💪 Google <table-cell-rows> Iniciar sesión en la</table-cell-rows>	NCI L	ibrary 👘 NCIRL	moodle 🗶 Lingue	wR wordreference	📙 leo 🛭 😝 Ti	raductor de DeepL	🛔 Longman Dictionar	🗱 Macmillan			»	Otros marcado	res
€2 50	DEPLOY & RUN TRANSACTIONS ENVIRONMENT JavaScript VM ACCOUNT O CAS LIMIT 300000 VALUE 0 VALUE 0 CONTRACT FRS - browser/FRS.sol	 i i i i i x x x 		ne FRS.sol X solidity ~6.5-16; cct FRS (* FI address */ idress public f_L000 * Tokens per loan i nt public constant ** Dornover address ** Bornover: loan i puping(address ~> ui ** Bornover: loan is Bornover: loan is Bornover: loan is Bornover: loan string loanPurpoi string loanPurpoi string loanPurpoi string loanPurpoi string loanPurpoi	DRESS; in WEI */ WEI_Tokens = 1000 for tokens creditr. int) public Address t of tokens debited int) public Address t of tokens debited int) public borrowe oan */ ATMEL: se; // And wer; i and tran rict ad	* (10**15); id */ ircoans; ircoans; te addr press ' nsaction	ess into 'Deploy 1) "Deploy fi " to execut	eld" te				20	
	OR At Address Load contract from Address Transactions recorded	~	33 * mc 34 35 36 } 37 38 39 * /*	difier onlyFI { require(msg.sende _;	er FI_ADDRESS);									÷
	Deployed Contracts		* 0 0	listen on network	Q Search wit	h transaction ha	ish or address							
٥	Currently you have no contract instances interact with.	to	• Use exp	upt strupt. Horts/.register(key,	, obj)/.remove(key)	/.clear() to re	egister and reu	e object across script	executions.					^

Deploy contract and check menu option made up of 2 orange buttons called "non-payable functions" which are used to input parameters and create a transaction. There are 5 blue buttons called "constant functions" which will only return values when clicked.

DEPLOY & RUN TRANSACTIONS DEPLOY & RUN TRANSACTION At Address tast contract from Address Transactions recorded Deployed Contracts	Home Program solid Program solid Contract FRS Contract FRS Program solid Contracts" arrow and you will see the different bottoms to execute the contract Contract Program solid Program solid Contracts" arrow and you will see the different bottoms to execute the contract
Institution of the second	2. Click "NewLoan" arrow to open it and you will see three fields to input parameters: loanReceiver, loanAmountFIATWei, and loanPurpose
AddressTokensC. address v AddressTokensC. address v bortowerLoam address v	21 22 /** Struct 23 struct Lo 25 vint 26 vint 27 strin 28 strin 29) NewLoan ◆
FLADDRESS Witt, Tokens	au Jan IoanReceiver 32 /** Mod.fr IoanReceiver 34 requirer IoanAmountFIATWee 35 36 } IoanPurpose 39 IoanPurpose string
Low level interactions i CALLDATA Transact	→ 0 1 ★ 0 0 listen 6

Scroll up to the top of the "Side Panel" up to the "account" section.



Scroll down to the "NewLoan" section in the "Side Panel" and paste the borrower's address in the box "LoanReceiver".

	DEPLOY & RUN TRANSACTIONS	Q Q Home FRS.sol X	2 tabs =
Cn	At Address Load contract from Address	1 pragma solidity 40.5.16; 4 contract FRS (ĺ
	Transactions recorded <1 🗸 🗸 🗸	6 /** FI address */ 7 address public FI_ADORESS; 8 Dacto horrowor's address into	
~	Deployed Contracts	9 /** Tokens per loan in WEI */ 10 ulst miblic constant WEI Tokens = 1000 *	
۲	✓ FRS AT 0X37611872 (MEMORY)	/** Borrower address for tokens - resisted mapping dadress -> viority-teril.charesst	
ý	InputProformaln string invoice, uint256 amountFlatSpentInW V	14 15 /** Address to mount of tokens debited */ 16 mappingtrearess -> uint) public AddressTokensDebited;	
	NewLoan	<pre>18 /** Borrower's loan file */ 18 mapping(address -> Loan) public borrowerLoan; 20</pre>	
	IoanAmountFIATWei: uint236		
	IoanPurpose: string transact	25 uint CMesi; // Conversion rate in WEI 26 address LoamReceiver // Address of Loan receiver-> borrower 27 string LoanPurpose; // Loan purpose explanation 28 string invoice; // Invoice for the Loan 29 20 3	

Go to following address and type the loan amount of $\notin 3,200$ (Let's assume $\notin 1 = 1$ Ether) to be converted into WEI.

https://etherchain.org/tools/unitConverter

erchain.org/tools/unitConverter ogle 🚦 Iniciar sesión en Ia 🌰 NCI Library	🕐 🏠 NCIRL moodle 🛛 🗶 Lingue	wR wordreference 🛛 🔀 Ieo 🛭 ᡇ Traduc	tor de DeepL 🏾 🗯 Longman Dictionar	🖈 📭 🛍 😋 🤇
🖉 etherchain.org 🛛 🔗 Blockchain 👻 💄	Accounts 👻 🗠 Statistics 👻 🖬	Tools 👻 🛠 Mining Pools 🕑 👻		Block Number / Tx Hash
♀ INFO: Ethereum uses a lot of different	units like Wei, GWei, Shannon	and Ether. You can use this tool to co	nvert between them.	
Ethereum unit converter				
320000000000000000000000000				Wei
320000000000000000				KWei
32000000000000	\backslash			MWei
320000000000		1 Introduce 2200		GWei (Shannon)
320000000		1. Introduce 5200		Szabo
3200000				Finney
3200		2. Copy the equivale	ent amount in WFI	Ether
3.2				KEther
0.0032				MEther
				CEther

Go back to Remix IDE to the "NewLoan" section. Paste the amount in WEI in loanAmounFIATWei, write the loan purpose and execute the transaction.

📑 In	formación sobre la re 🗙 📔 🏟 Remix - Ethereum IDE 🛛 🗙	Remix - Eth	hereum IDE 🗙 📳 Remix-IDE Layout — Ri 🗙	🔯 Traductor de DeepL - 🛛 🗙 🛛 🥔 Ethereum Unit Convert: 🗙 🛛 🤣 Ethereum Unit Convert: 🗙 📔 + 👘 🗖	3 ×
\leftarrow	C eremix.ethereum.org/#optimize=false8	mVersion=nu	lll&version=soljson-v0.5.17+commit.d19bba1	13.js 🏠 🖾 🖾 🛣 🖈	R E
Apl	icaciones 💪 Google 🚦 Iniciar sesión en la 💩	CI Library 👘	NCIRL moodle 🗾 Lingue 💘 wordreference	🌃 leo 🛭 Traductor de DeepL 🇯 Longman Dictionar 🔗 Macmillan 🛛 🔅 📙 Otros m	narcadores
۲	DEPLOY & RUN TRANSACTIONS		Home FRS.sol ×		2 tabs #
ළු	At Address Load contract from Address Transactions recorded	3 4 - 5 6 7	contract FRS { /** FI address */ address public FI ADDRESS;	1. Paste the 320000000000000000000 WEI	
~~ ♦	Deployed Contracts V FRS AT 0X37611872 (MEMORY)	8 9 10 11	/** Tokens per loan in WEI / uint public constant WEI_Tokens = 1000	* (10**18);	- 1
ý:	InputProformaln string invoice, uint256 amountFiatSpentleW	12 13 14 15 16 17 18	<pre>/** Borrower agrees for tokens credit mapping(address -> uint) public Address /** indress to amount of tokens debite apping(address -> uint) public Addres /** Borrower's loan file.</pre>	2. Write the loan purpose in the corresponding field	
	NewLoan IoanReceiver: 0x61dD28D8AD13C3D737fd593c66D7C9K	19 20 21	mapping(address -> toan) public borrow		
	IoanAmountFIATWei: 32000000000000000000000000000000000000	22 23 * 25 26	<pre>struct Loan { unt LoanAmountFIATWei; // Amount unt CRWei; // Conversion and International address LoanAmountFIATWei; // Address o </pre>	3. Execute transaction	
	transact	27 28 29 NewLoan - tran	string loanPurpose; // Loan purpos string invoice; // Invoice for the nsact (not payable)		
	AddressTokensC address	32 33 - 34 35	<pre>/** Modifier to restrict access only t modifier onlyFI { require(msg.sender FI_ADDRESS); ;</pre>	o FI */	
	AddressTokensD address	36 37 38 39 =	}		÷
	borrowerLoan address		0 Contra	the benerication hash as address.	

Go to the "InputProformaInvoice" Section and proceed to input the purchase data.



Check the transaction performed by inputing the the borrower's address in the corresponding field.



References

Remix documentation

https://remix-ide.readthedocs.io/en/latest/.

Blockchain for developer - Solidity

https://blockgeeks.com/guides/solidity/

Solidity tutorials

https://www.tutorialspoint.com/solidity/