

Toffee Mint Handover Manual

IS480 AY17/18 Semester 2

**Contact Person :**

Sean Koh - sean.koh.2015@smu.edu.sg

**Last Modified :** 27 April 2018

Contents :

[**1. Introduction**](#_pxrzqpdgafq4) **3**

[1.1. Project Scope :](#_706dq03k1hho) 3

[1.2. Project Architecture :](#_sgurhzn9mbe5) 4

[**2. Getting Started.**](#_u0cpah5d0v6) **5**

[2.1. Development Environment :](#_r9f5crnvz5ef) 5

[2.2. Code Repository :](#_4upbtgeop9z9) 5

[**3. Truffle Middleware Services.**](#_ua8je78dlbz2) **6**

[3.1. List of Middleware Services :](#_osijjnkr0h5y) 6

[**4. Client-facing Components.**](#_w6nmzd3ovu8k) **8**

[4.1. Trade Finance Life Cycle :](#_qjbg06bc95w5) 8

[4.2. Game Module :](#_xmmzr4uzggv7) 9

[**5. Known Bugs.**](#_kavgd4u2f9nn) **10**

## 

## 

## 1. Introduction

**What is Project ToffeeMint?**

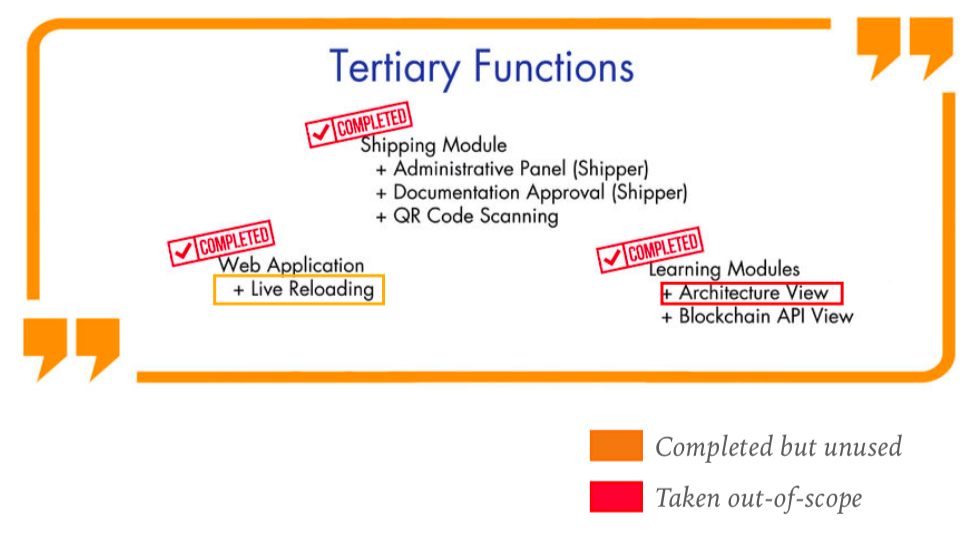
Project ToffeeMint was an IS480 Project aimed at building a blockchain-backed trade finance banking application for SMU tBank. SMU tBank is a teaching bank used for in-class exercises, to better explain inner workings of financial banking systems taught in class.

This new blockchain-backed trade finance system uses the Ethereum Blockchain Architecture, and runs on a private network. Smart contracts have been built to house and execute trade finance Letters of Credits, keep track of their statuses, and store the URLs of the relevant trade documents, that are related to each contract.

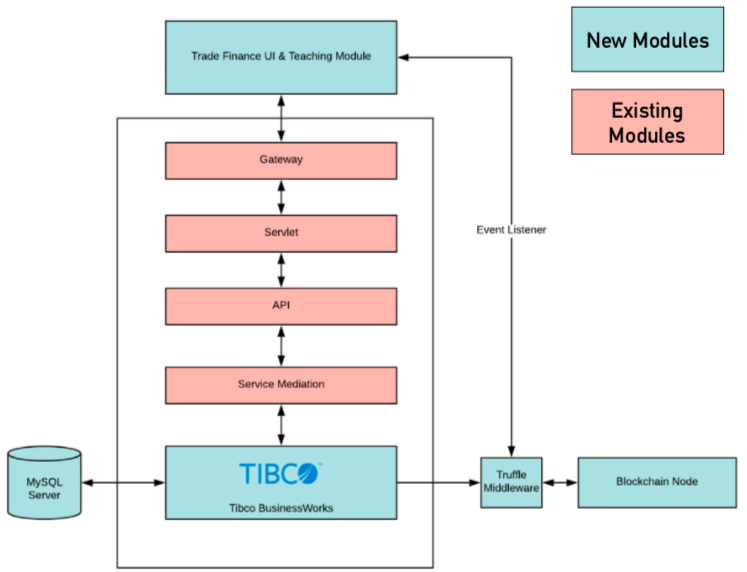
For teaching purposes, a learning module has also been built to facilitate in-class exercises. The web application features an in-built quiz module, which gives students timed pop-quizzes as they perform the tasks assigned to them. A leaderboard feature has also been built to allow students to compete against each other; based on how well they performed on their timed-quizzes.

### 1.1. Project Scope :





### 1.2. Project Architecture :



Project ToffeeMint was built upon a Service-oriented Architecture (SOA), which relies on an enterprise service bus, which facilitates access to tBank’s back-end services.

**Trade Finance UI & Teaching Module**

The Trade Finance User Interface was built with pure javascript. This component is the client-facing end of our project, and facilitates the trade finance lifecycle, and administers the learning module quizzes.

**TIBCO BW**

TIBCO Business Works serves as an enterprise service bus, facilitating the flow of information between the back-end services of tBank.

**Truffle Middleware**

A node.js+express application that compiles and migrates the blockchain smart contracts, while also serving as a middleware, allowing TIBCO BW to access the smart contracts stored on the blockchain.

**Blockchain Node**

A stand-alone ethereum node, with a gas limit of -l 10,000,000.

It currently runs on Ganache-CLI.

## 2. Getting Started.

In this section, we will outline everything that you will to get your development environment ready to run our blockchain-backed trade finance environment.

### 2.1. Development Environment :

You will need to install the following software to get our Ethereum blockchain running on your development machine (install in order):

1. Node.js
2. NPM
3. Git Bash
4. Ethereum Core
5. Ganache-CLI

Once you have installed the softwares listed above, continue on to the next section.

### 2.2. Code Repository :

You can clone our entire repository at : <https://github.com/wonkishtofu/ToffeeMint>

Once you have had your development environment setup (as detailed in Section 2.1.), and setup your NPM dependencies, navigate to App > 0.TruffleApp on your command line.

After having done so, install our node dependencies with the command :

npm install

This might take awhile, but it would install all the package dependencies needed for us to run the application.

After the dependencies have been installed, you will find the following shell scripts in the main folder (0.TruffleApp):

1. blockchain.sh (run first)
2. script.sh (run next)

Run the blockchain.sh file first, to get our ganache instance of a blockchain node up and running, before running script.sh, which will compile our smart contracts, migrate them on to our private chain, and start the blockchain middleware (On windows, you can run both on command prompt).

## 3. Truffle Middleware Services.

Now that you have got the blockchain and truffle middleware up and running on your computer, now you will be able to access the smart contract functions that are up and running on your testnet.

In this section, we have listed all the services which have been developed and are used by our blockchain-backed trade finance system.

**Important Things to Note on Middleware Services**

* These services are called upon by the middleware (TIBCO BW), through a ***get*** method.
* To query for a contract that exists on the blockchain, query using the full reference number (including the six ‘0’s that precede the contract number).
* You can expect JSON responses from all functions listed here.
* The querying the event listeners with a reference number will return you the entire history of that specific contract that has been put onto the chain. The JSON array response will be arranged in sequence of most recent iteration of the contract, to earliest iteration of the contract (highest block >lowest block).

### 3.1. List of Middleware Services :

LocalHost Service URL : localhost:9001

(Use this URL if you are testing on a local environment)

Staging Environment Service URL : <http://smu.tbankonline.com:9001>

(Use this URL if you are testing the response on the staging server. This server exists only within the school (SMU) network, so clearpass is needed to access this service )

**\*\*Letter of Credit\*\***  
  
//Create Contract  
http://smu.tbankonline.com:9001/lc/createContract?refNum=\_\_\_\_\_\_&contract=\_\_\_\_\_\_  
  
//Get Contract   
http://smu.tbankonline.com:9001/lc/getContract?refNum=\_\_\_\_\_\_

//Modify Contract  
http://smu.tbankonline.com:9001/lc/modifyContract?refNum=\_\_\_&contract=\_\_\_

**\*\*Contract Status\*\***//Set Status  
http://smu.tbankonline.com:9001/lc/setStatus?refNum=\_\_\_&status=\_\_\_  
  
//Get Status  
http://smu.tbankonline.com:9001/lc/getStatus?refNum=\_\_\_

**\*\*Contract Amendments Request\*\***//request Amendment  
http://smu.tbankonline.com:9001/lc/amendLC?refNum=\_\_\_&amendments=\_\_\_  
  
//get Amendment Request  
http://smu.tbankonline.com:9001/lc/getAmendments?refNum=\_\_\_

**\*\*BOL and other Trade Documents\*\***   
//Set BOL  
http://smu.tbankonline.com:9001/lc/setBOL?refNum=\_\_\_&BOL=\_\_\_  
  
//Get BOL  
http://smu.tbankonline.com:9001/lc/getBOL?refNum=\_\_\_

---------------------------------------------------------------------------------------------------------------------------

**\*\*Events** **Listeners**\*\*  
- Think of this as like a search history function   
- Define search parameters where ‘\_\_\_’ is specified

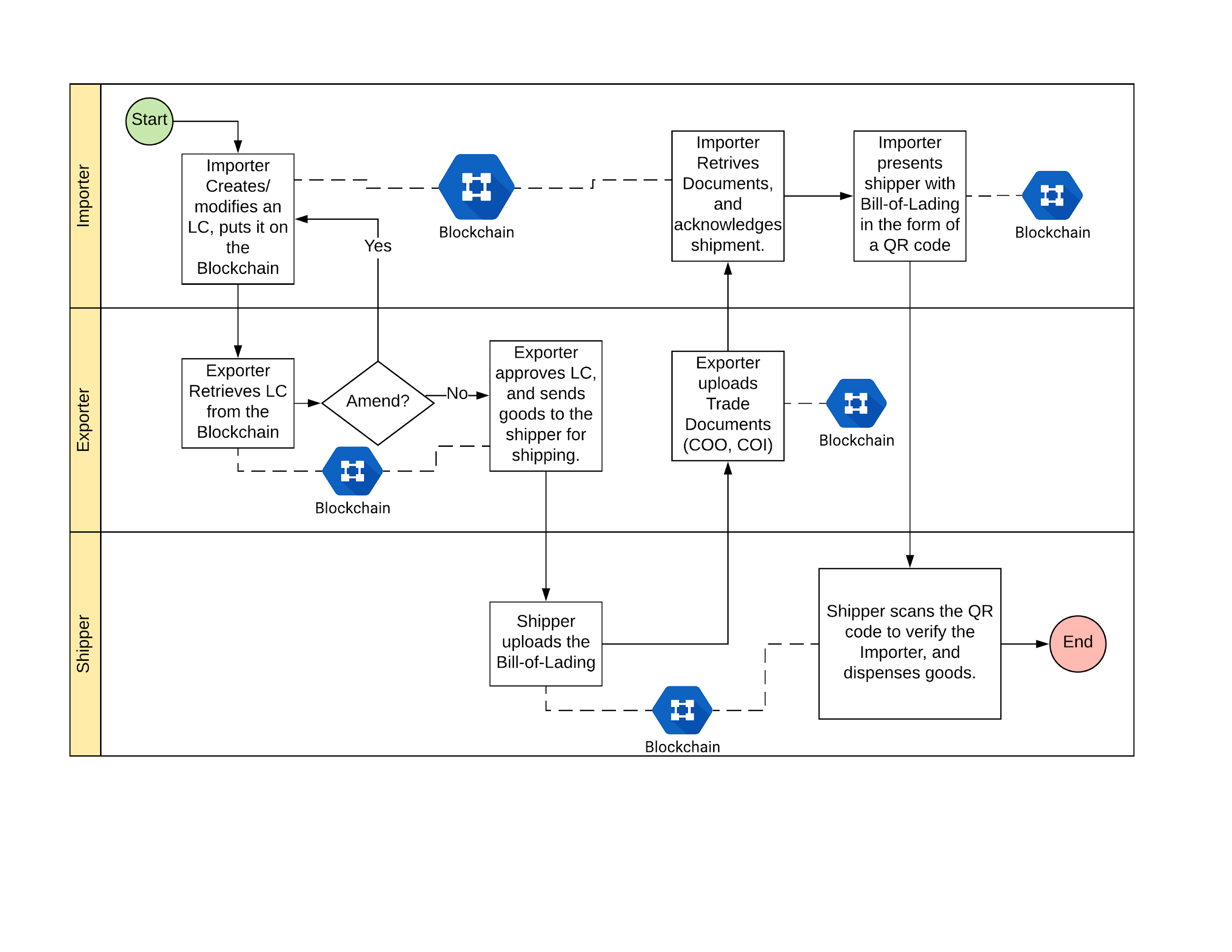
//Get events of when an LC is created, where ‘\_\_\_’  
http://smu.tbankonline.com:9001/events/LCCreated?refNum=\_\_\_\_\_

//Get events of transaction hash when an LC is created, where ‘\_\_\_’  
http://smu.tbankonline.com:9001/events/LCCreatedHash?refNum=\_\_\_\_\_  
  
//Get events of when an LC is modified, where ‘\_\_\_’  
http://smu.tbankonline.com:9001/events/LCModified?refNum=\_\_\_  
  
//Get Status events where ‘\_\_\_’  
http://smu.tbankonline.com:9001/events/status?refNum=\_\_\_&status=\_\_\_  
  
//Get events of when a BOE/BOL is modified, where ‘\_\_\_’  
http://smu.tbankonline.com:9001/events/documentsModified?refNum=\_\_\_  
  
//Get events of when an amendment is requested, where ‘\_\_\_’  
http://smu.tbankonline.com:9001/events/amendments?refNum=\_\_\_

## 4. Client-facing Components.

The Trade Finance User Interface was built in pure javascript, but as it uses an older version of a bootstrap-based theme, Jquery libraries were imported to keep some UI elements working.   
  
Note that the Trade Finance User Interface application is a netbeans project, that has to be opened in netbeans version 8.1 and above.

### 4.1. Trade Finance Life Cycle :

****

To simulate a the lifecycle of a letter of credit, we will let students take one of 3 roles :

1. Importers
2. Exporters
3. Shippers

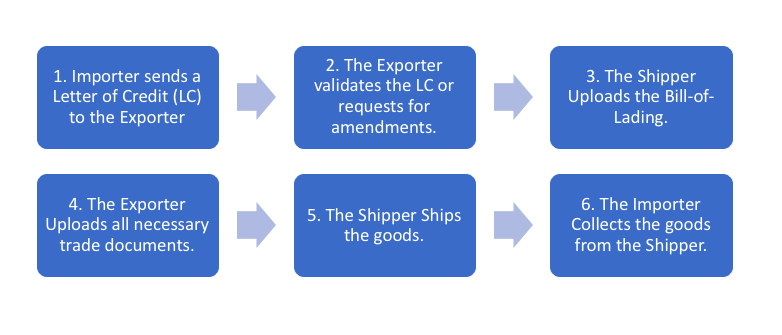
Students will be put in groups of threes, and they will be tasked to complete the lifecycle together, as a team.

To access the application, students will log into : <https://tbankonline.com/SMUtBank_TradeFinance/>

with their assigned username and pin, and complete the tasks assigned to them; via task sheet which can be found here : <https://seank1.typeform.com/to/dLIEjY>

A list of login usernames and password can be found in the students.csv in the documentation folder.

### 4.2. Game Module :



Life Cycle (LC) Steps

For each step of the trade finance life cycle, students will be prompted with a timed pop-quiz, which uses an existing quiz service.

To set the quiz questions, you must first create a question on the tBank administrative panel, and later, modify the following files that correspond to the quiz steps listed above. The quiz html files can be found under the main ‘web’ file of the netbeans project.

**Quiz Files :**

LC Step 1. : gameModelPop\_Create.html

LC Step 2. : gameModelPop\_Amend.html

LC Step 3. : gameModelPop\_Modify.html

LC Step 4. : gameModelPop\_Upload.html

LC Step 5. : gameModelPop\_Collect.html

Pre-test : gameModelPre.html

Post-test : gameModelPost.html

## 5. Known Bugs.

The following are known bugs that we were not able to resolve, the bugs are all minor, and UI related.

There are/is (1) known bug(s) .

1. **The Trade Finance WebApp is broken in Safari.**

Description :

The latest version of Safari (as of 27th April 2018), does not work with the web application. The core of the problem lies with the window not detecting screen size, and hence, not displaying the contents on the page.