Design Manual

Authors

Andrew Guerin C00243040

Supervisor

Dr Hisain Elshaafi

Submission date

17/12/21

INSTITUTE of TECHNOLOGY CARLOW

Institiúid Teicneolaíochta Cheatharlach

Abstract

The purpose of the Blockchain marketplace project is to develop a web application to allows users to sell and buy products from other users within blockchain based environment aided by smart contracts. Transactions are recorded on the blockchain for security and transparency. The application allows the user to view previous sales and buying history, give feedback to a buyer and seller, as well as creating listings for items to be sold.

The project is designed with a Peer 2 Peer business approach where each user on the app are nodes within the network. For each sale that is in process escrow may be applied by the seller where the currency for the transaction is held in an escrow account before finally being to the respective seller.

Contents

1. Introduction

The following document will detail the interface design and architecture of the Blockchain marketplace application project. This document follows the research manual and functional specification document are created to give a more informed understanding of the design requirements for the application. The design manual uses the information learned from previous documents to achieve informed decisions for potential screen designs although they may change drastically when application development takes place.

The project aims to establish a web application that offers users the ability to buy and sell goods on a digital marketplace supported by a blockchain to secure and record transactions made. Given that the price of Ethereum (ETH) is volatile at the given moment a user's account balance may be worth more or less on a given day where the market effects the price. Given that a seller may set the price of a product listing they instantiate how much ETH the product is worth while the seller simply matches the price. Given that funds may be sent through escrow and the time delay of funds being released may vary, the actual amount of ETH received may vary in terms of its Dollar or Euro value in respect to the original price presented at the time of sale.

Application Architecture

The Blockchain Marketplace application can be denoted as having one main logical component.

The user component provides the functionality for the users to do various tasks such as creating an item listing to be publicly viewable, buying a product, viewing order or sale history. More options include creating an account, viewing feedback giving by other users or to rate seller based on their conduct during transactions of products.

Front End, Back End, Database Architecture

The Blockchain Marketplace application is to be supported by a back-end server database being Firebase.

Firebase is a useful piece of software most commonly detailed as a backend as a Software (BaaS). The goal of Firebase is to supply backend tools that can be used to develop an application with a streamlined fashion as there is an abundance of tools and documentation to aid in assistance when using Firebase. In the case of the proposed project Firebase will be used as a realtime database.

A Realtime database is known as a NoSQL cloud-based database where data is updated regularly in real time. The information that is gathered is stored within JSON for easy readability also allowing for the same database to be shared across both IOS devices as well as Android. Firebase simplifies the development process by allowing the back end to be built in a shorter timeframe. The allows time to handle separate issues that are trickier and of more functional importance. While MySQL is good for complex data, Firebase handles large sets of data more efficiently. Firebase can quickly sync data via android and is more dynamic and can facilitate unstructured data while MySQL has a more present schema
would be harder to perform more complex queries.

Technologies

Node.js

Node.js is a JavaScript runtime designed specifically to build scalable network applications. It allows the use of JavaScript for backend development making it easy to match functions between the back and front-end of an application. Creating a blockchain with Node.js is possible with documentation readily available for use.

Blockchain

A blockchain is somewhat similar to an augmented linked list where each node within the chain points to the preceding node. Each block within the blockchain contains important transaction information such as the timestamp of when the block was added to the chain, the encrypted hash of the represented block that precedes the current block

Solidity

Solidity creates machine level code that is executed the Ethereum Virtual Machine (EVM). The EVM is a virtual computer on the blockchain that turns peoples or organizations ideas into code that runs applications on the blockchain. EVM smart contracts have some limitations that need to be addressed. One limitation is the limited access to useful library functions for parsing JSON structures or floating-point arithmetic.

Smart Contracts built using Solidity are a way to automate non-business and business processes between different people. This ensures that people making transactions on the blockchain do not have to worry about risks such as fraud or not being able to use the same currency.

Smart Contracts

GitHub

To backup the code of this project and to aid in the following of the agile methodology Github was chosen as the repository of the project. Github allows local git repositories to be pushed to remote repositories on Github. Regular commits will be pushed to this repository with each increment to the application to follow the agile guidelines.

Database Schema

UI Design Prototypes

The chosen colour scheme for the project is represent as follows

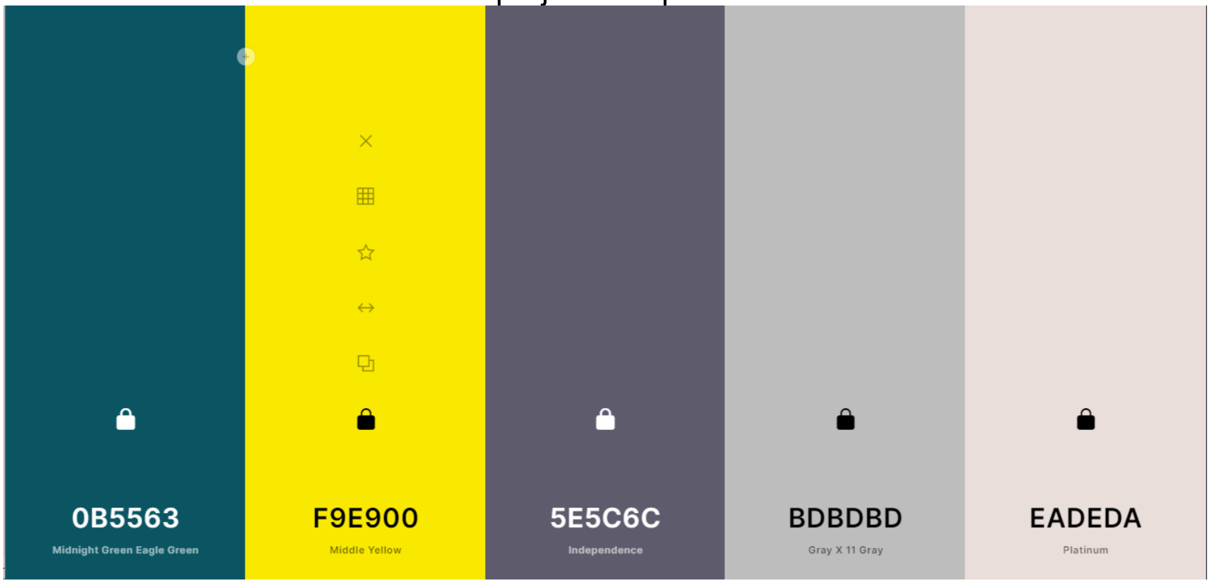


Figure 1. Colour Pallette

Screen Prototypes

This section represents the potential screens that may be implemented into the project's application.

Login Screen

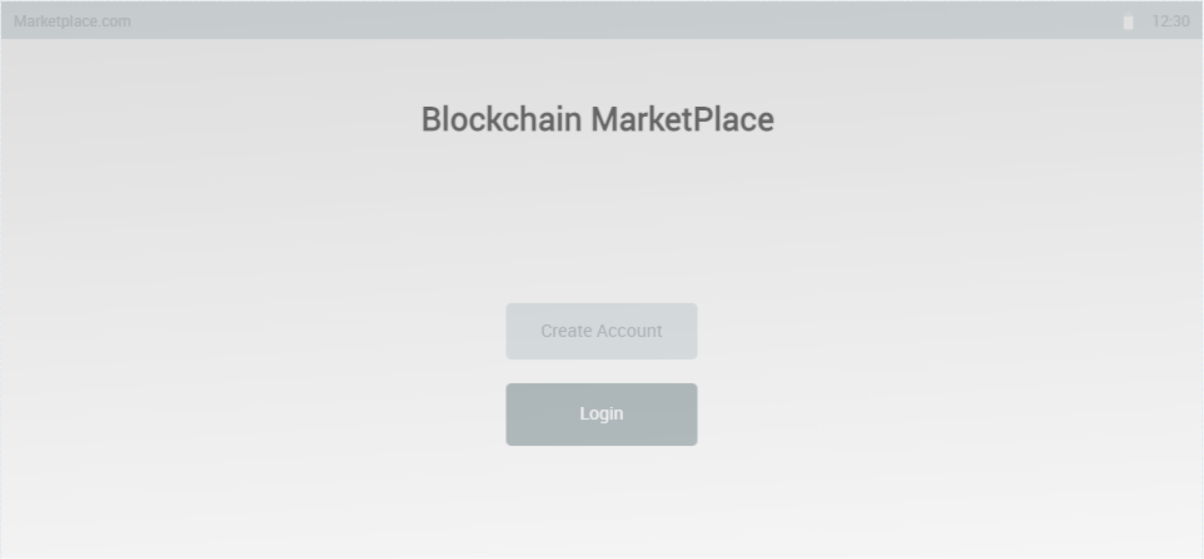
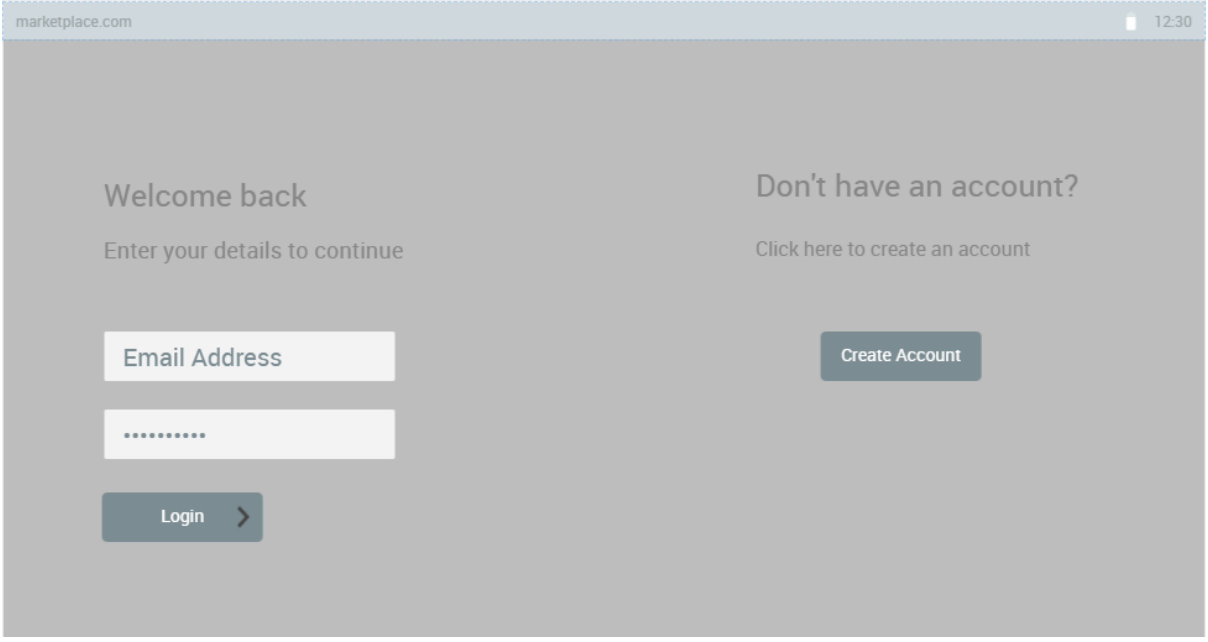


Figure 1. Welcome screen

The login screen will be the first screen that is viewed by a user when they open the application.

Sign In Screen



marketplace.com

12:30

Welcome back

Enter your details to continue

Email Address

••••••••••

Login

Don't have an account?

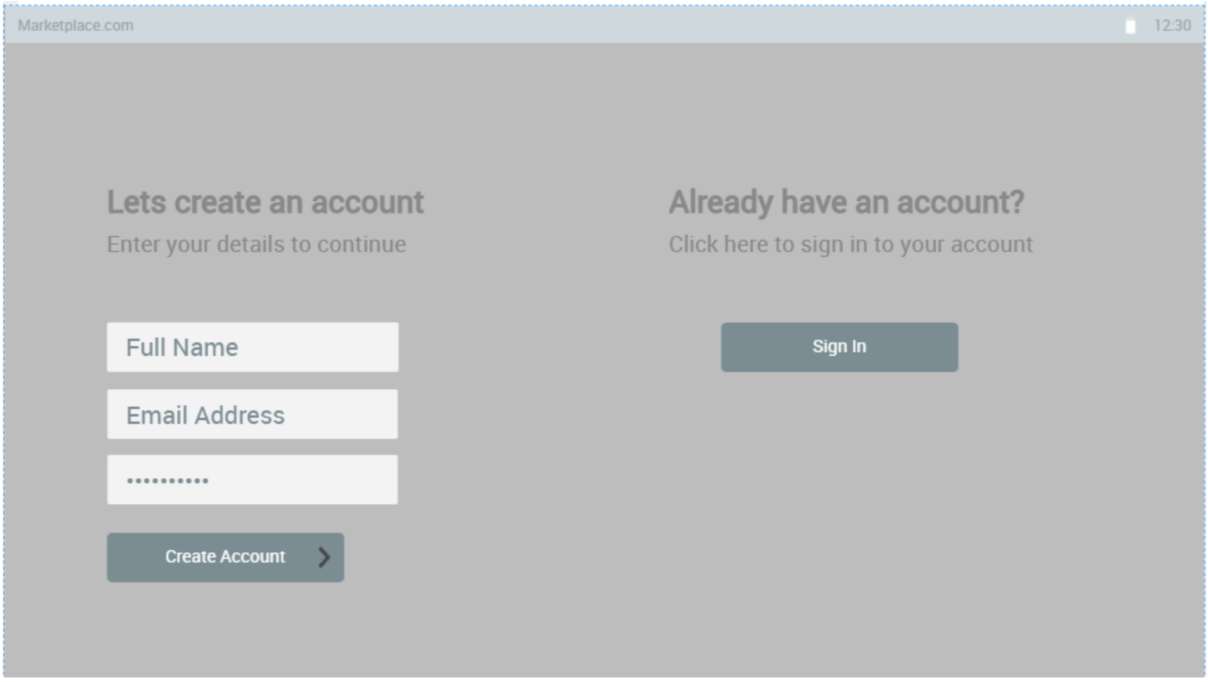
Click here to create an account

Create Account

Figure 2. Login Screen

The sign in screen is displayed to the user after they select the option to login in from the welcome screen. Within this screen the user must enter their valid email address and password before being successfully logged into their account. The application will compare the information given by the user to the information within the database for validity. If the details are incorrect and error message will appear stating that some of the information provided by the user does not match the details of the same email address. If the user does not already have an account, then they can choose the option to create an account where they are brought to the Sign Up screen.

Sign Up Screen



Lets create an account

Enter your details to continue

Already have an account?

Click here to sign in to your account

Figure 3. Sign Up Screen

The Sign Up screen is presented for a user that wishes to create a new account which will be added to the backend database for future reference.

The user will enter their details such as name, email address and a password to successfully create and account.

If the user already has a registered account then selecting the sign in button will direct the user to the login screen where they can enter their details to login.

Home Screen

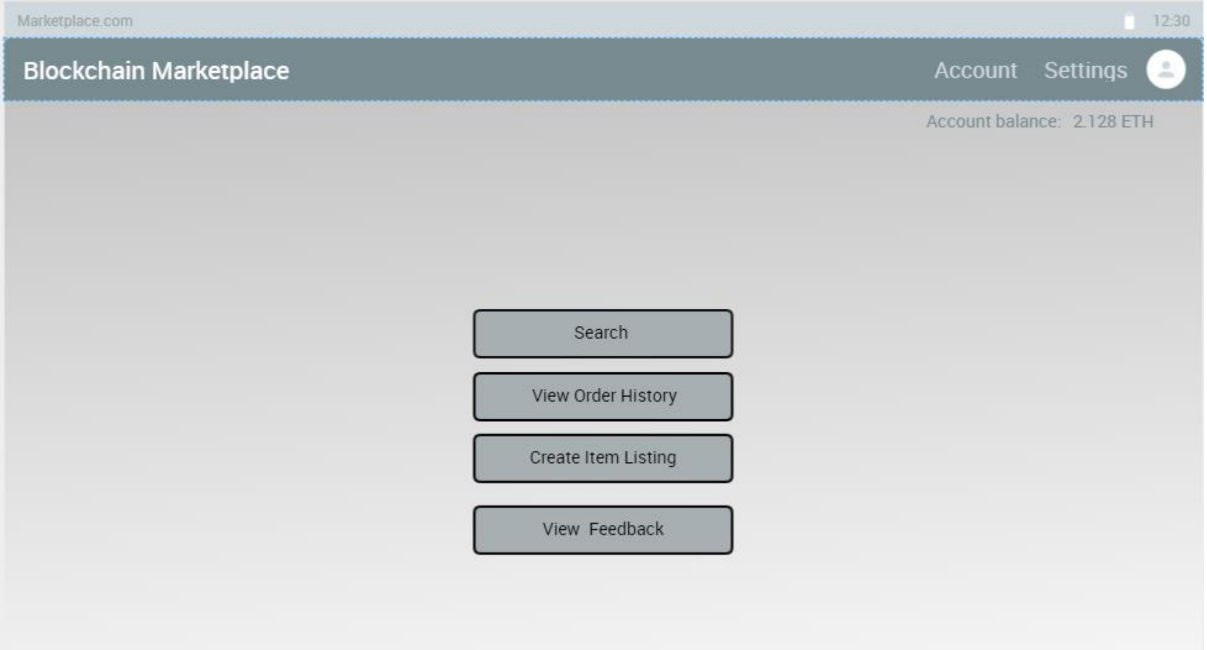


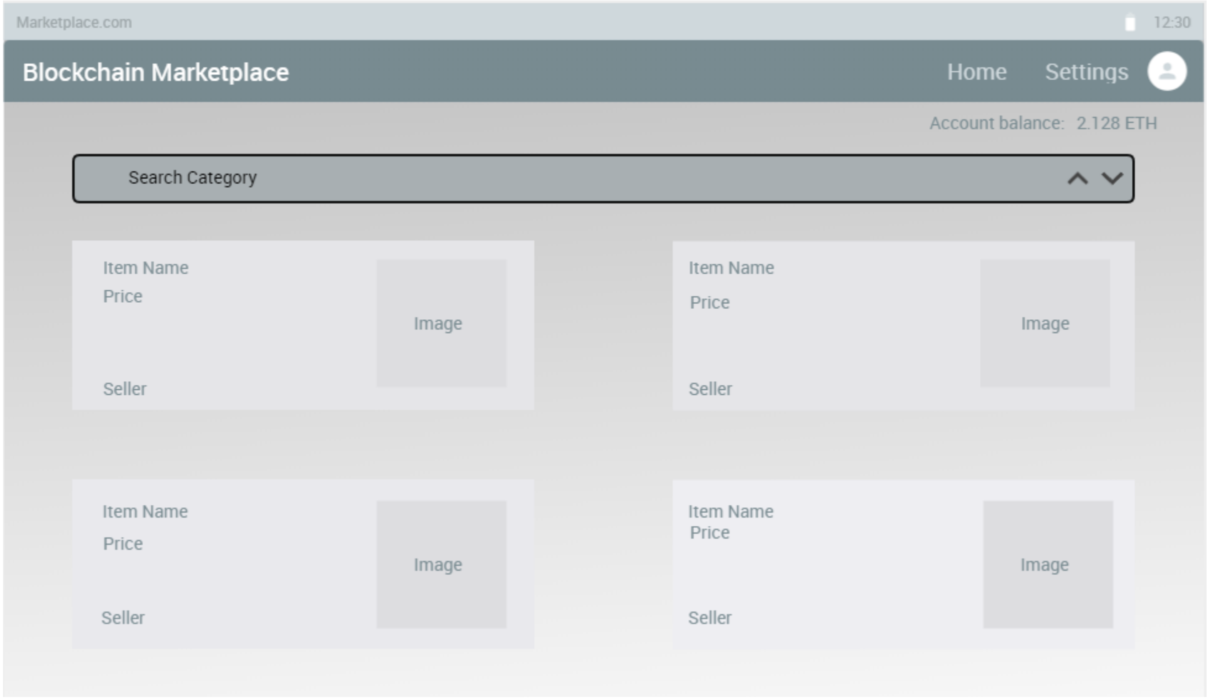
Figure 4. Home Screen

The home screen is the main screen the user will interact with. There are several options to choose from at the home screen being:

- **Search**
- **View Order History**
- **Create Item Listing**
- **View Feedback**

By selecting one of these options the user is directed to their respective screen for further interaction

Search Function

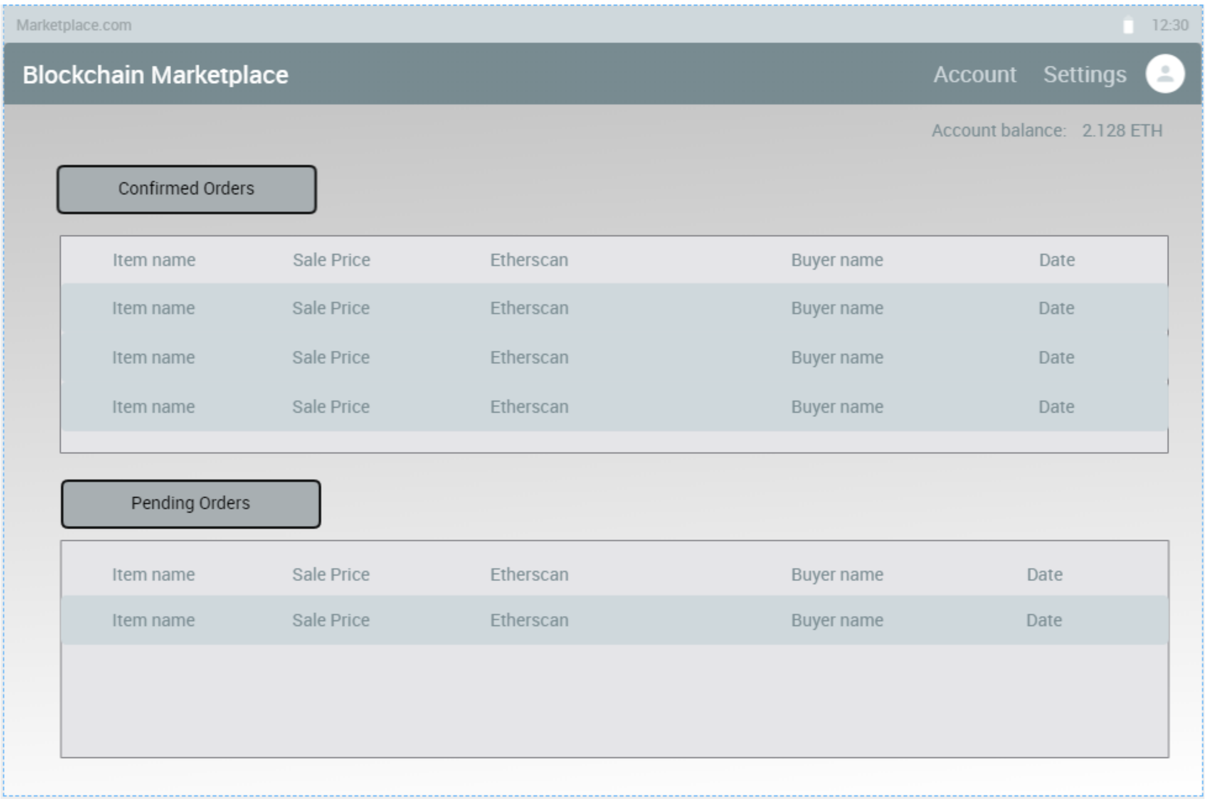


Account balance: 2.128 ETH

Fig 5. Search Function

From the search catalog screen the user can choose to search from a wide range of products by category or a specific search where results are displayed in a grid format. Further information about a particular product is displayed such as the item in question, seller name, price of the product as well as an image that represents the item listing.

View Order History



Account balance: 2.128 ETH

Item name	Sale Price	Etherscan	Buyer name	Date
Item name	Sale Price	Etherscan	Buyer name	Date
Item name	Sale Price	Etherscan	Buyer name	Date
Item name	Sale Price	Etherscan	Buyer name	Date

Item name	Sale Price	Etherscan	Buyer name	Date
Item name	Sale Price	Etherscan	Buyer name	Date

Figure 6. View Order History

The View Order History screen allows the user to view orders they have paid and received to date. The confirmed orders that are completed are displayed in their own section while pending orders that have not been fully completed are displayed in the Pending Orders section.

Some information that are displayed includes the name of the item, seller name, date, sale price as well as an etherscan link to show the transaction details of the purchase.