

# Task 4 - Specification

## High Level Project Overview

Our project is an ASCII art trading platform/wallet, similar to [OpenSea](#) for NFT trading/collecting. In general, users will be able to create new accounts and login, view their art collections and buy and trade art pieces on a simulated market which fluctuates based on supply and demand of specific items.

## How It Will Work

- Running the project will prompt the user to sign into their account using their username and password in the user interface
- Once logged in, users will be presented with a menu of options, some of which (with consequent sub menus) are detailed below:
  - View Wallet
    - View Entire Collection
    - View Art Piece
    - View Cash on Hand
    - View Wallet Value
  - View/Update Profile
    - Edit Username
    - Edit Password
    - View Net Worth
    - Edit Wallets
  - Enter Art Market
    - View Art For Sale
    - View Art for Trade
    - Mint a new piece of ASCII Art to the platform

## CRC Cards

class	
General functionality	Interacts with...

## Entity Level:

User	
Knows wallets Knows username Knows password Knows net worth	Wallet UserManager

Wallet	
Knows owner Knows its stored art Knows how much currency is stored in it Knows its worth Knows whether its public or private	User Art WalletRegistrar TradingUtil

Art	
Knows its own data Knows its own price Knows its name Knows what wallet its stored in	Wallet ArtManager

## Use Case Level:

ArtManager	
Finds a piece of ascii art and access its information Relays information between ascii and other classes Adds a new piece of ascii art to the system	Art ArtLibrary Appraiser UniquenessChecker ArtFacade MarketSim StorageManager

Appraiser	
Appraises a piece of art based on an algorithm, assigns a cash value	ArtManager

ArtLibrary	
Knows all the art pieces in the system	Art manager

UniquenessChecker	
Checks if a piece of art already exists in the system	ArtManager

ArtFacade	
A facade for the controller to access the art manager functions	Controller ArtManager

MarketSim	
Adjusts the value of art pieces based on market environment (if certain symbols appear too often, it will lower the value of the art when it's appraised. etc)	ArtManager Market

WalletRegistrar	
Keeps track of all the wallets Access and modify individual wallets Relay wallet information to other classes	TradingUtil Market

Trading Util	
Makes the trade of an artwork between two wallets	TradingUtil

Market	
Knows all the tradable arts Knows the value of these arts Gives preview of the arts Re-evaluates art Escrows and other higher level trading actions	WalletRegistrar MarketSim MarketFacade StorageManager

MarketFacade	
Acts as a facade for the controller to access Market functions	Market Controller

UserManager	
Keeps track of all the users Access the user's fields and relay information to other classes Adds new users to the system	User LoginUtil RegisUtil UserFacade Storage Manager

LoginUtil	
Ensures that the password of the user matches, to allow for a log in	UserManager

RegisUtil	
Registers a new user and adds them to the system	UserManager

User Facade	
Acts as a facade for the controller to access the user manager's functions	UserManager Controller

Storage Manager	
Store and retrieve stored data	ArtManager UserManager

	Market
--	--------

### Controller Level:

Controller	
Decides which use cases is to be applied Passes information between the UI to the use case level	ArtFacade UserFacade MarketFacade Presenter

Presenter	
Presents the information to the user Collect user input to pass to controller	Controller