



VENDA: ELECTRONIC ASSET TRADING PLATFORM

CAB302 Semester 1, 2021

Jeremy Chang, Nicole Truong, Natalie Smith – Group 21

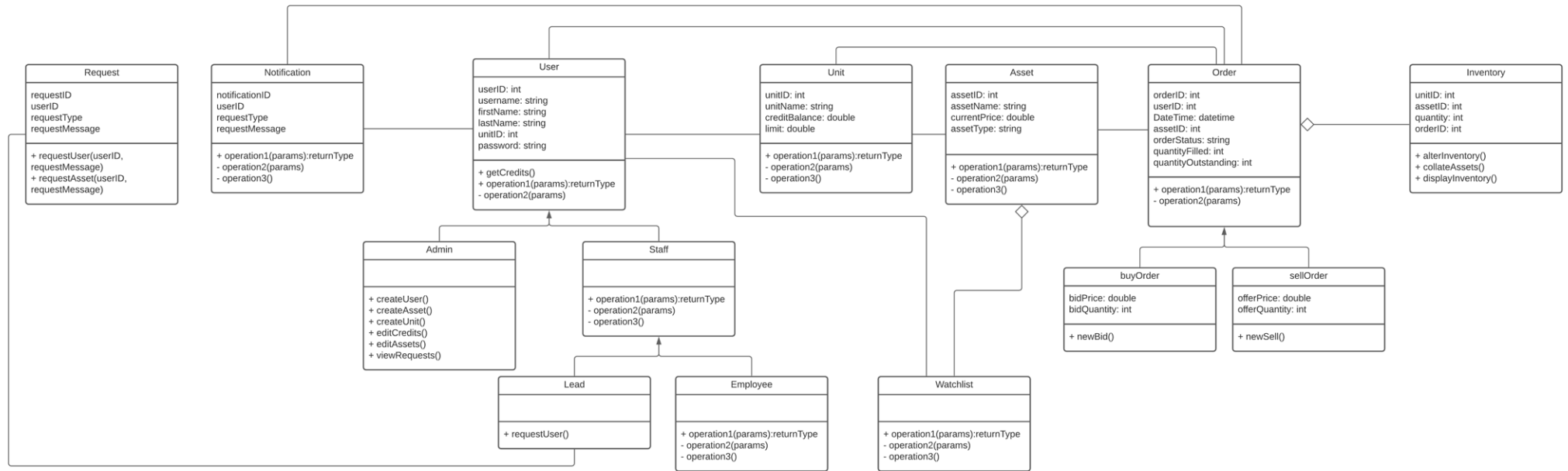
This document includes the Detailed Design documents for Milestone #1

Table of Contents

Detailed Design	0
UML Class Diagram	0
GUI Designs	0
Hierarchy of Pages	1
Wireframes.....	0
Database Schema	0
Tables	0
ORM Diagram	2

Detailed Design

UML Class Diagram



GUI Designs

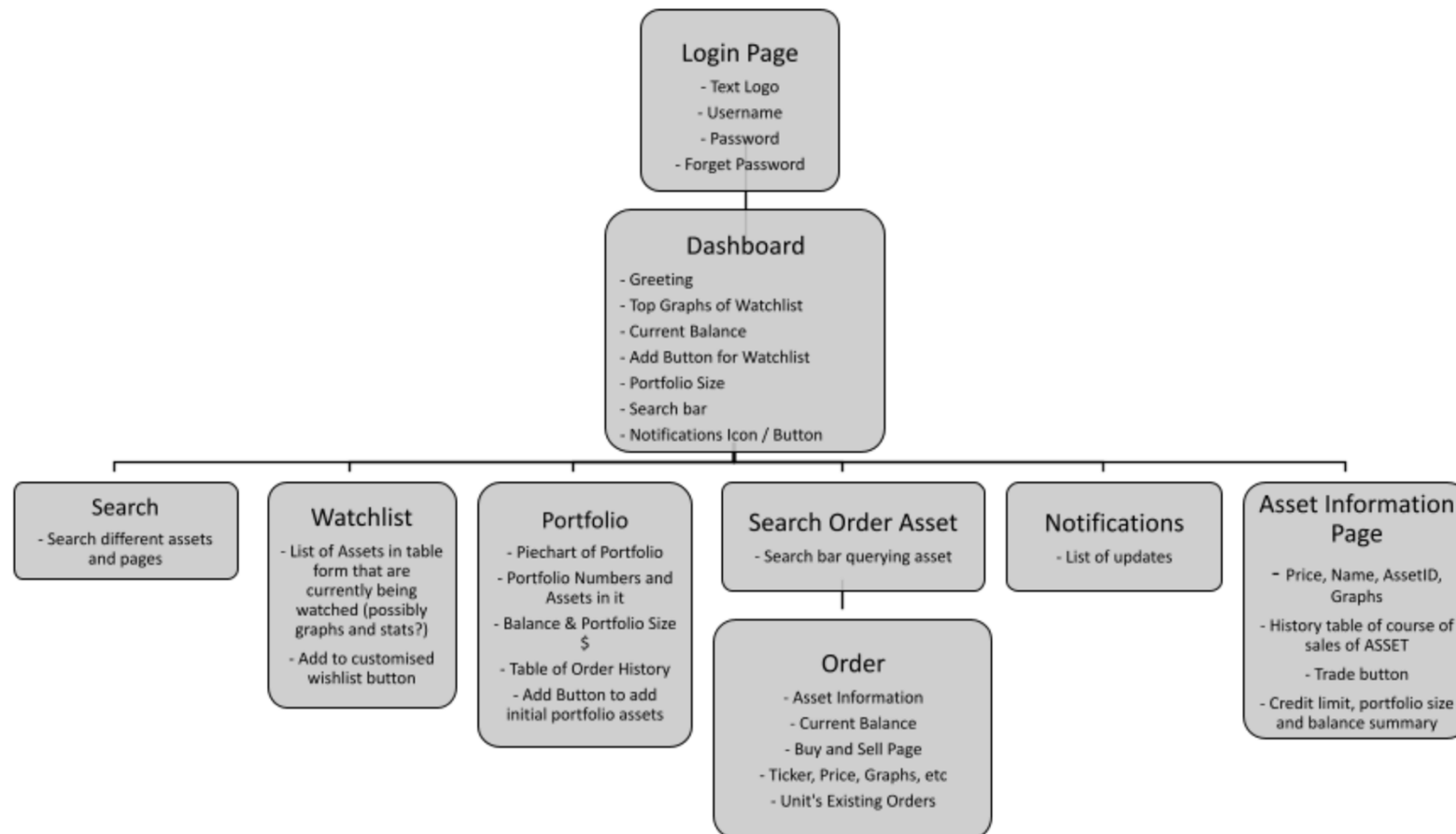
Style Guide

For the design of the website, we have chosen a simple, minimalist blue colour palette to evoke a responsible, loyal and trustworthy feel to the platform. As shown below, we have also decided on a Logo for the platform.



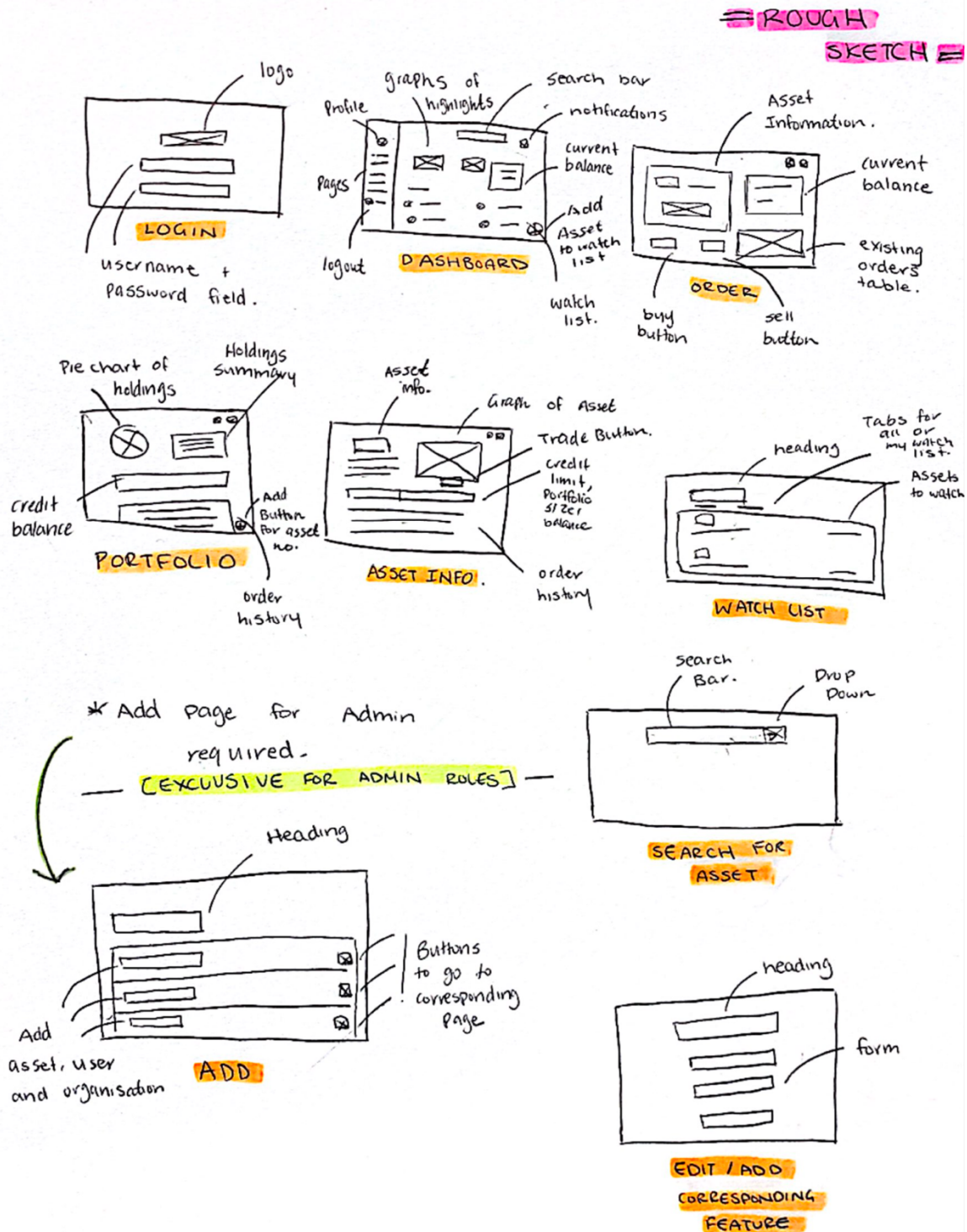
Hierarchy of Pages

The hierarchy of the pages was developed to establish the different elements required for each of the pages within the GUI. This would then be used in designing the wireframes.



Wireframes

Below are icons of the different pages we expect to be included in the electronic trading system. Consistent design elements were applied to ensure the user can easily and effectively navigate throughout the platform.



Database Schema

Tables

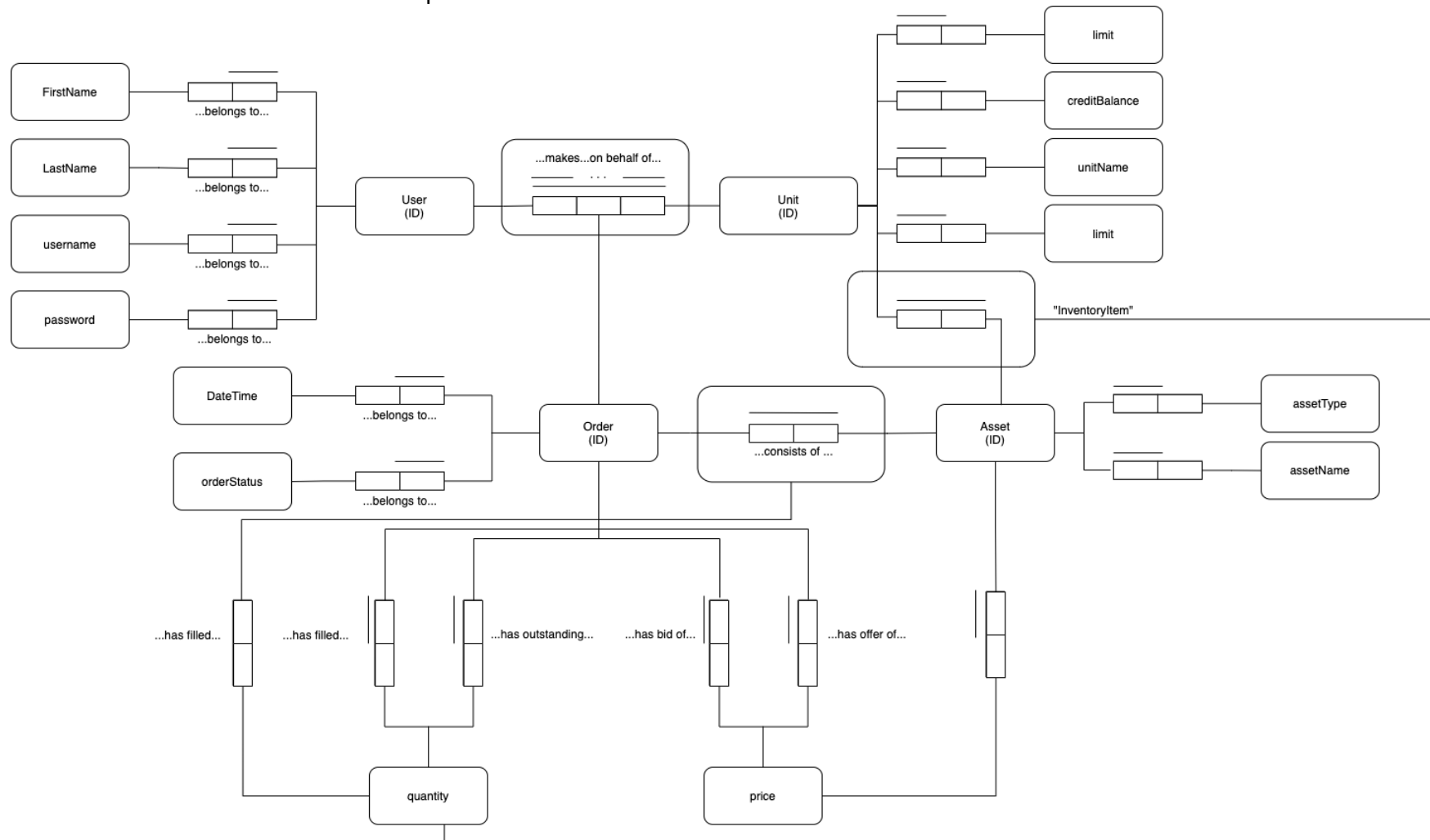
Below is a table of information stored within each of the tables within the database. It also shows the relevant constraints of each of the variables used.

Table	Description	Variable	Data Type	Constraints
Users	This table is used to store the users within the system. Each user is provided with a unique username and password. Along with this, it is required that the user be associated with a specific unitID, to ensure operations regarding orders and buying and selling are effective.	userID	Int	Primary Key, unique, mandatory
		username	string	AlternateKey, Mandatory, unique, mandatory
		firstName	String	
		lastName	String	
		unitID	Int	Foreign Key, mandatory
		Password	string	Mandatory, encrypted
Units	The purpose of the units table is to collate the information of each unit within the organisation, and provide relative information required for the system – this includes assigning the unitID with a unit name, credit balance and limit to the credits a unit can use.	unitID	Int	Primary Key, unique, mandatory
		unitName	string	Mandatory
		creditBalance	Double	Mandatory
		limit	Double	mandatory
Assets	Assets are used in various functions within the system.	assetID	Int	Primary Key, mandatory
		assetName	String	Mandatory
		currentPrice	Double	
		assetType	string	mandatory
Orders	Each order that a user submits will be added to the order table. In doing so, it would also assign an order ID to it. For each order, as they are processed before being fulfilled therefore requires the use of the variables order status, quantity filled and outstanding. This assumes an order could go through	orderID	Int	Primary Key, mandatory
		userID	Int	Foreign Key, mandatory
		dateTime	datetime	Mandatory
		assetID	int	Foreign Key, mandatory
		orderStatus	string	Mandatory
		orderType	Enum {BUY, SELL}	mandatory
		quantityFilled	Int	Mandatory
		quantityOutstanding	Int	mandatory
		orderID	Int	Primary Key, mandatory

Trade History	<i>Each completed and successful trade that is conducted would be stored in a trade history table. Each unit would have different past trade orders, and these should be outputted to the user based on their associated userID.</i>	userID	Int	Foreign Key, mandatory
		dateTime	datetime	Mandatory
		assetID	int	Foreign Key, mandatory
		orderStatus	string	Mandatory
		orderType	Enum {BUY, SELL}	mandatory
		quantityFilled	Int	Mandatory
		quantityOutstanding	Int	mandatory

ORM Diagram

A first iteration of the ORM was developed to visualise the connections between the tables.



Relational Schema

Users {userID, username, firstName, lastName, unitID, password}

Units {unitID, unitName, creditBalance, limit}

Assets {assetID, assetName, currentPrice, assetType}

Order {orderID, userID, dateTime, assetID, orderStatus, quantityFilled, quantityOutstanding}

