Software Architecture Document

Katacoin Live

Team 4

Table of Contents

1. Introduction …3
    - 1.1 Definitions, Acronyms and Abbreviations …3
    - 1.2 Overview …3
2. Architectural Goals and Constraints …3
    - 2.1 Client Side …3
    - 2.2 Server Side …3
    - 2.3 Persistence …3
    - 2.4 Reliability …3
    - 2.5 Development Tools …3
    - 2.6 Schedule …3
3. System …4
4. Wireframes …5
5. Use-Case …6
    - 5.1 Use-case diagram …6
    - 5.2 Use-case realizations …7
6. ERD …8
7. Process View …8
    - 7.1 Create Order …8
    - 7.2 Manage Orders …9
    - 7.3 Edit Order …9
    - 7.4 Delete Order ..10
    - 7.5 Search Trades ..10
8. Implementation View ..11
9. Size and Performance ..12
10. Quality ..12

1. Introduction

Katacoin Live will allow you to safely trade your Katacoin with other buyers and sellers. The system displays an orderbook, ticker feed and graph of Katacoin data. It will retain your orders and trades while you make new ones. It will automatically match orders for you.
This document elaborates the software architecture for Katacoin Live.

1.1 Definitions, Acronyms and Abbreviations

MVC – Model view control architecture
OOP – Object oriented programming
KTC – Katacoin

1.2 Overview

This document will present a detailed analysis of the architecture of Katacoin Live. Further sections cover the goals, constraints, use-cases and views.

2. Architectural Goals and Constraints

2.1 Client Side

Only one client may operate KTC Live at a time.

2.2 Server Side

The server is localhost:8080 on the presenter's computer.

2.3 Persistence

All data will be saved in the presenter's MySql database.

2.4 Reliability

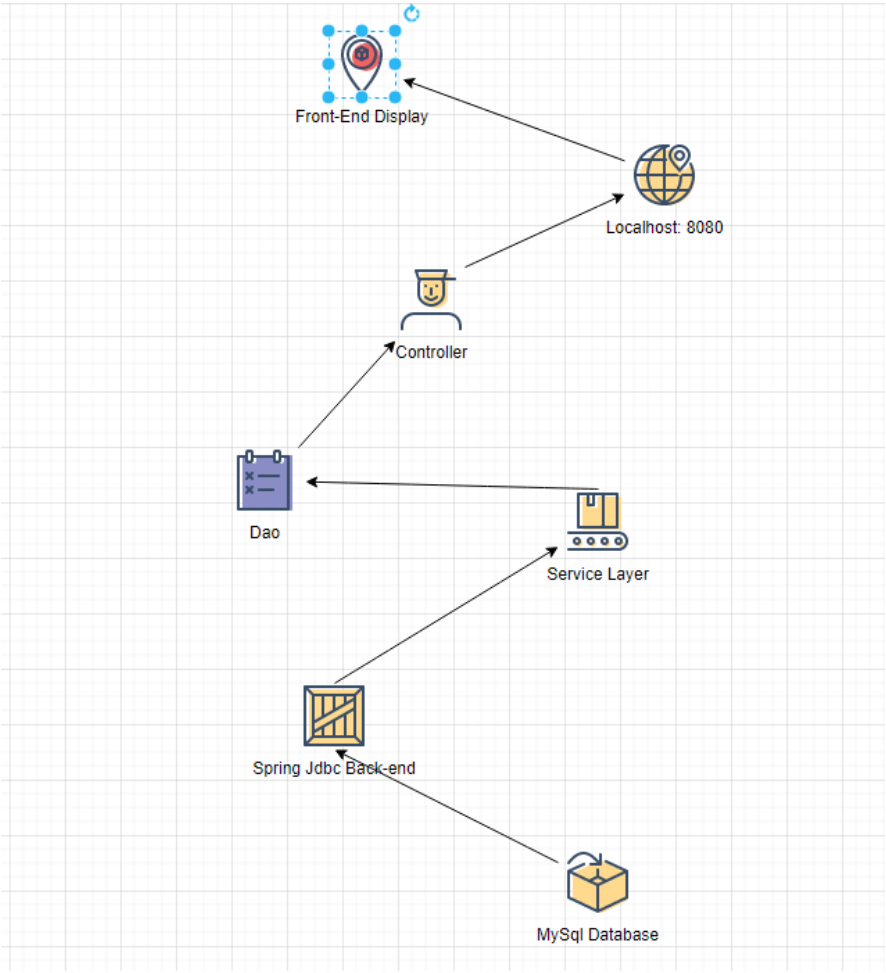
The system will be subjected to several testing operations (unit testing, system testing) before being deployed.

2.5 Development Tools

Programming: NetBeans IDE, VSC
Database: MySql
Diagrams: Draw.IO
Documentation: MS Word, Excel
Front-end: React

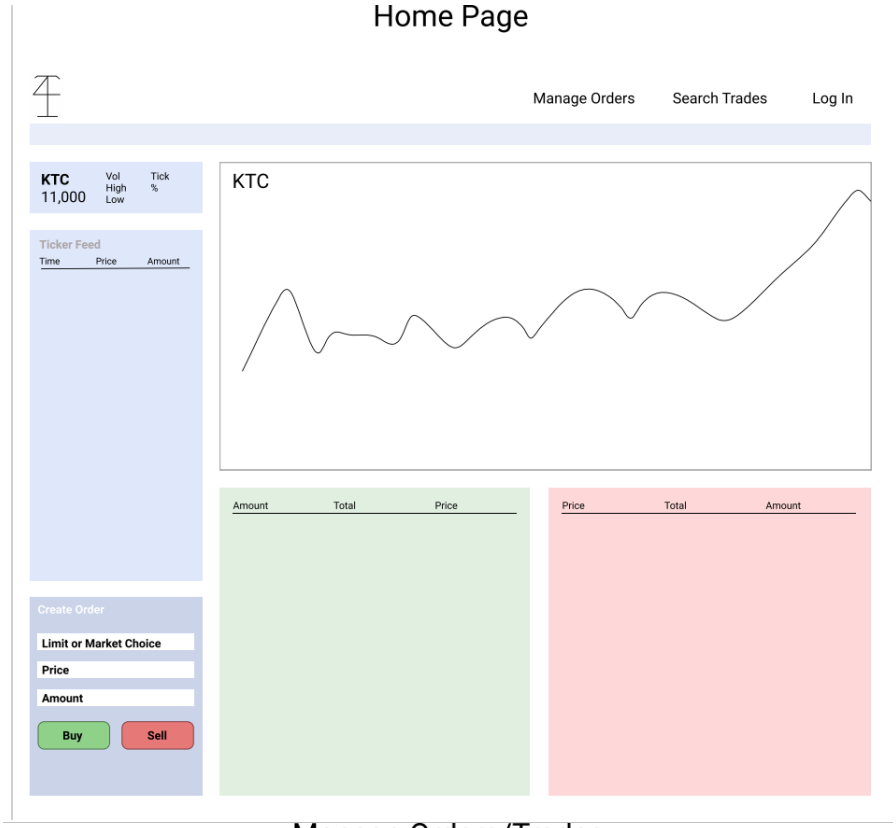
2.6 Schedule

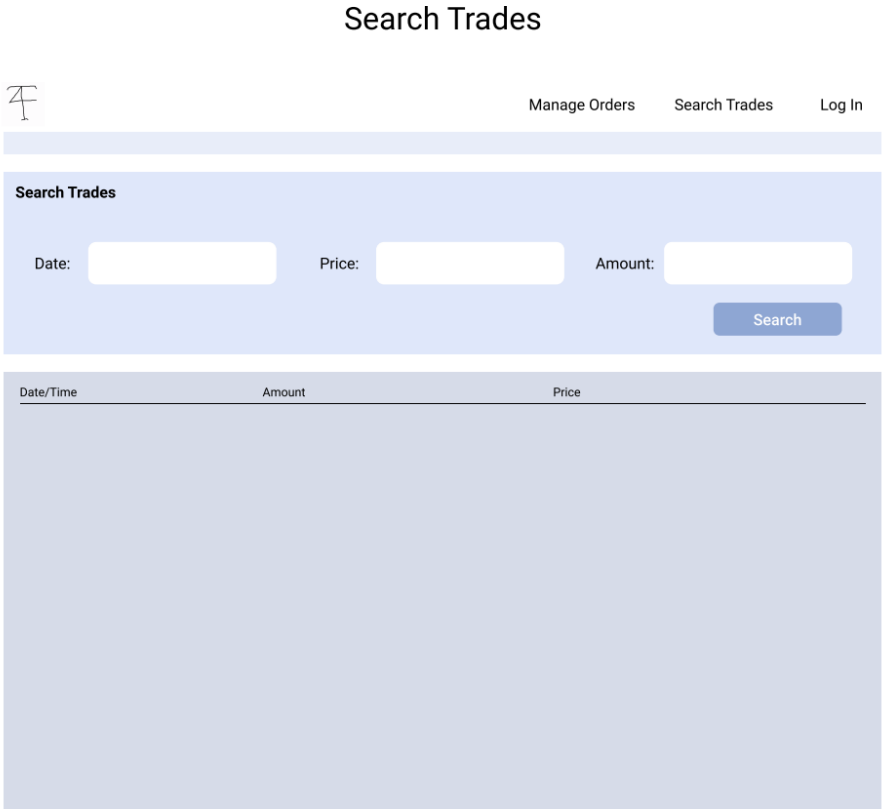
	ID	Task	Start Date	Finish Date
1	Proj10	**Analysis**		
2	Proj1	Requirement Meetings	09/10/20	09/11/20
3	Proj2	Documentations	09/10/20	09/23/20
4	Proj3	System Requirements	09/10/20	09/11/20
5	Proj17	Data Dictionary	09/10/20	09/22/20
6	Proj11	**Design**		
7	Proj4	Design Logo	09/10/20	09/10/20
8	Proj5	Design Database Schema	09/10/20	09/11/20
9	Proj6	Test Database	09/11/20	09/18/20
10	Proj7	Software Design	09/11/20	09/14/20
11	Proj9	Interface Design	09/11/20	09/14/20
12	Proj12	Design Finished	09/10/20	09/18/20
13	Proj13	**Development**		
14	Proj14	DTO development	09/11/20	09/15/20
15	Proj15	DAO development	09/14/20	09/17/20
16	Proj18	Service Layer development	09/15/20	09/22/20
17	Proj19	Front-End development	09/15/20	09/24/20
18	Proj20	Development Finished	09/11/20	09/24/20
19	Proj21	**Test**		
20	Proj22	DAO Testing	09/15/20	09/18/20
21	Proj23	Service Layer Testing	09/17/20	09/19/20
22	Proj24	Front-end Testing	09/22/20	09/24/20
23	Proj25	System Testing	09/23/20	09/24/20
24	Proj26	Testing Finished	09/16/20	09/24/20



3. System

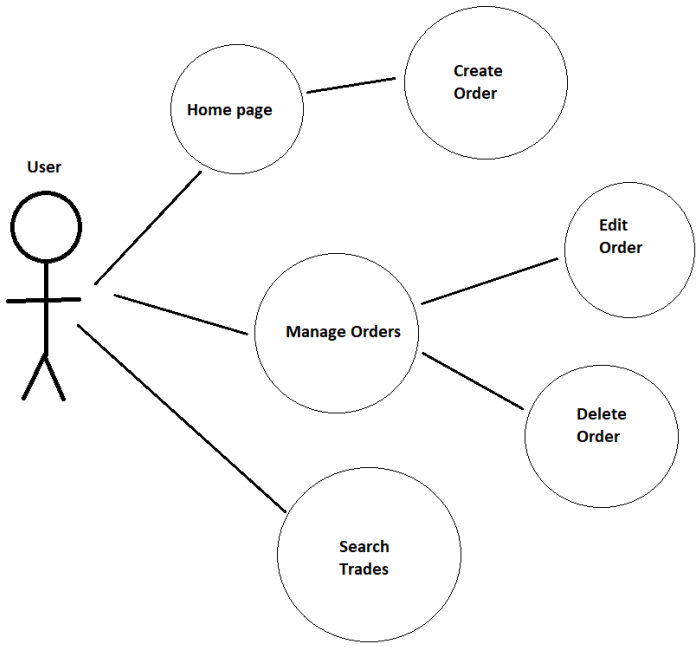
4. Wireframes





5. Use-Case

5.1 Use-case diagram



5.2 Use-case realizations

#1 – Create Order

Use case name: Create an order
Triggering event: User creates a Katacoin buy or sell order
Brief description: When the user creates an order, they will input the price, amount, and whether it is a buy or sell. The order is created and stored in the database until it is completed or deleted.
Flow of events: User loads home page, where the create order component appears. User enters information then creates the order.

#2 – Manage Orders

Use case name: Manage orders
Triggering event: The user navigates to the manage orders page from the navigation bar.
Brief description: When the user accesses their manage orders page, a list will appear that includes only their incomplete orders available to edit or delete.
Flow of events: User loads home page, then navigates to the “Manage Orders” in the navigation bar.

#3 – Edit Order

Use case name: Edit order
Triggering event: The user clicks the edit button
Brief description: On the manage orders page, an order that appears can be edited. The amount and price of Katacoin can be changed.
Flow of events: The user clicks on the edit button next to the order on their list of incomplete orders. A new page is loaded. The user can then manually change the amount of KTC and price of the KTC they are after. The user can then click save to save the new data for the order or they can click cancel, which will keep the order from saving and return them to the order page.

#4 – Delete Order

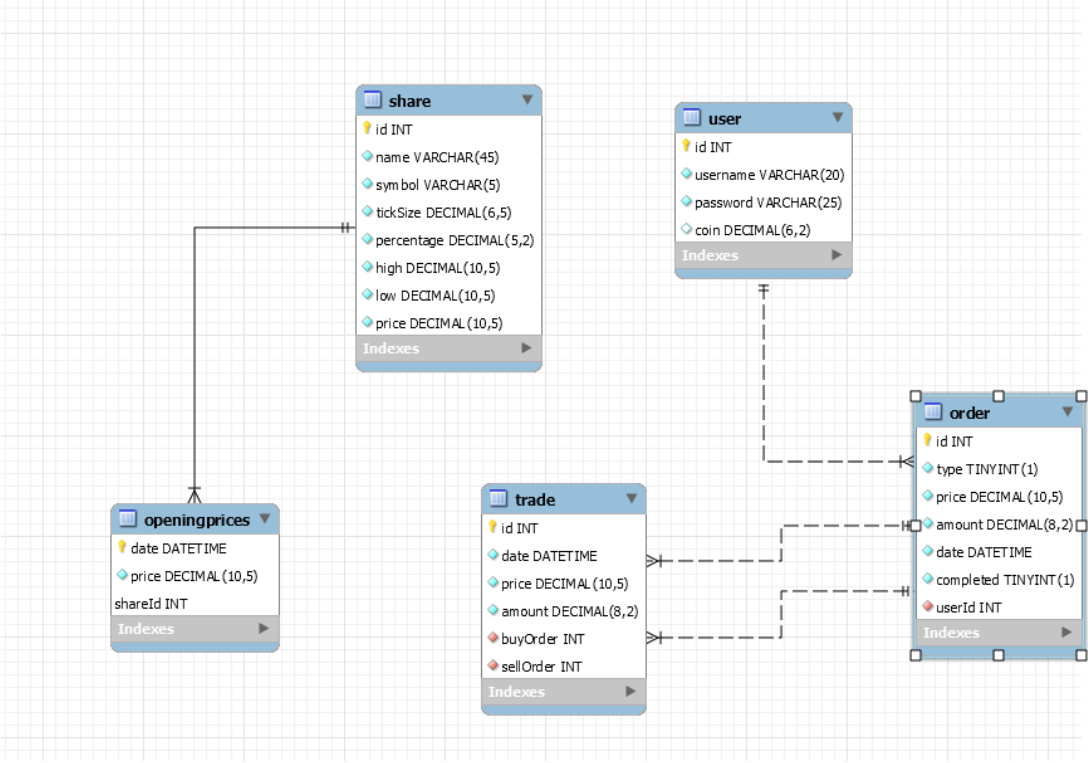
Use case name: Delete order
Triggering event: The user clicks the delete button
Brief description: On the manage orders page, an order that appears can be deleted.
Flow of events: The user clicks on the delete button next to the order on their list of incomplete orders. The user is then prompted whether they want to delete the order or not can confirm to complete the deletion.

#5 – Search Trades

Use case name: Search trades

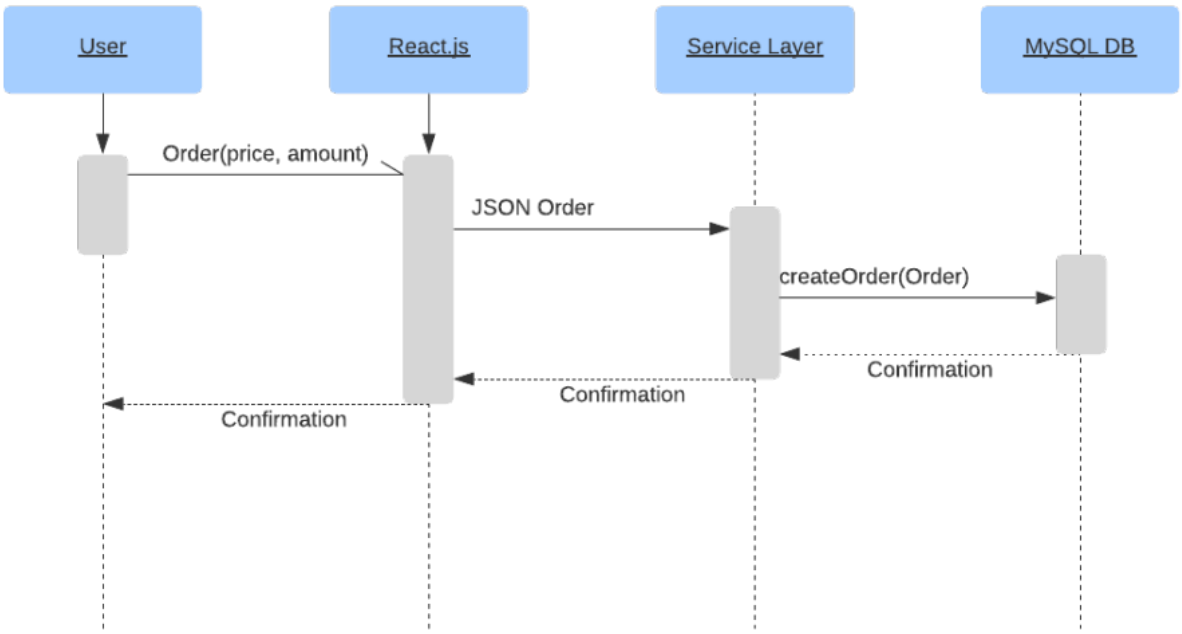
Triggering event: The user searches for trades
Brief description: On the search trades page, the user can search all the trades in the system by amount, price, or date.
Flow of events: The user navigates to the Search Trades page via the navigation bar. A menu at the top of the page allows for three choices to search by. The choices are amount, price, or date. After choosing one of these options from a drop-down menu, the user then gives the search function a range. This range is a number for amount and price but a range of dates for date. After clicking search, the list will change to reflect the parameters of the user's search.

6. ERD

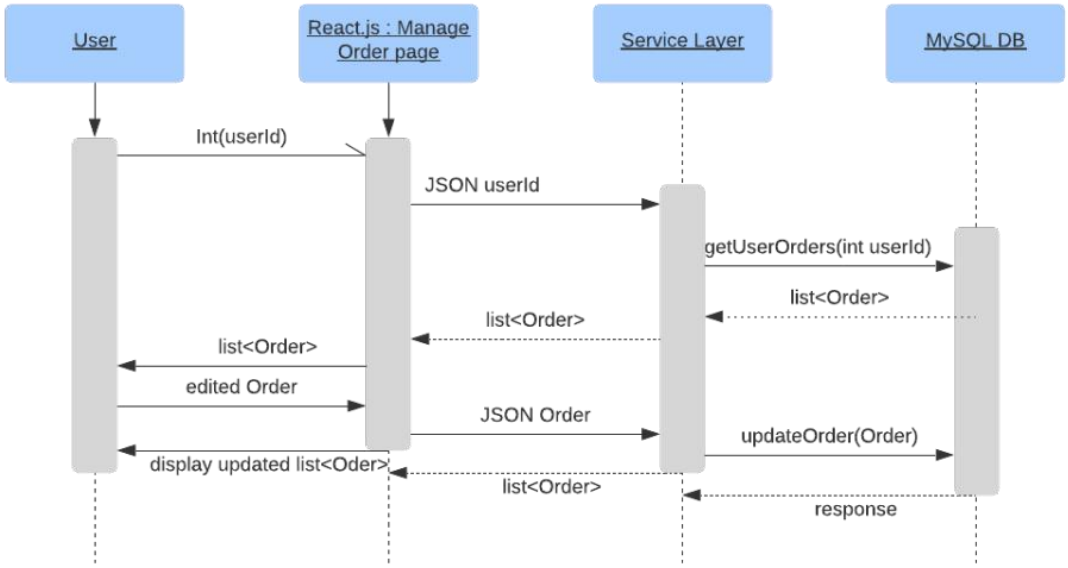


7. Process View

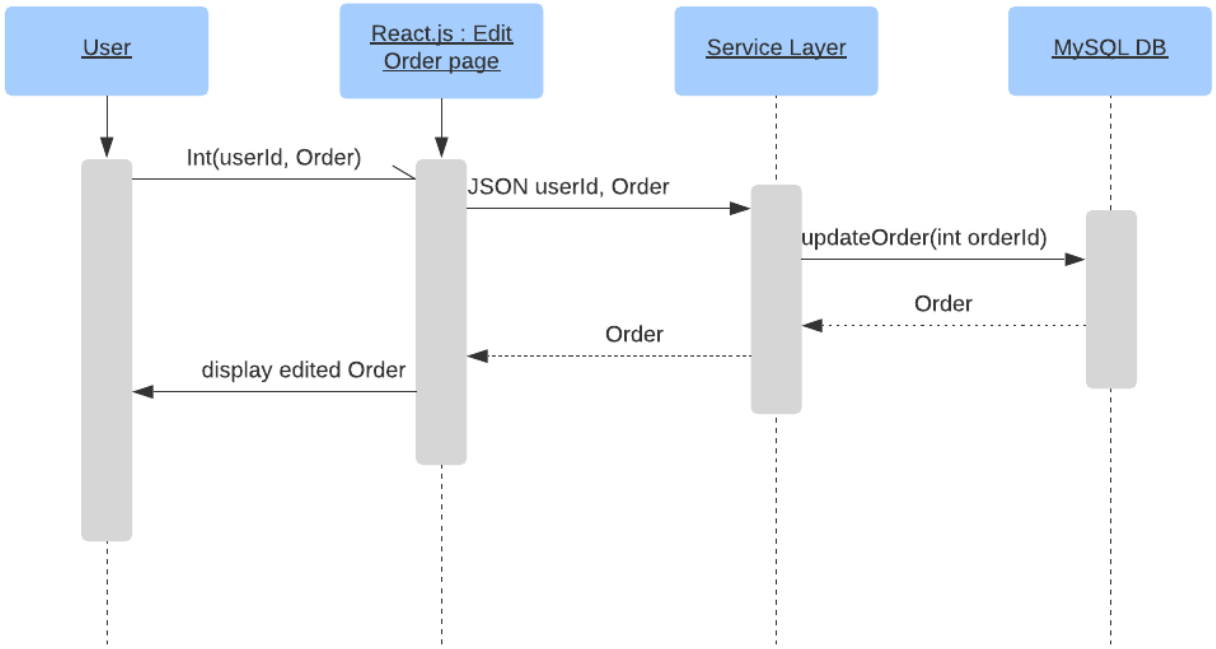
7.1 Create Order



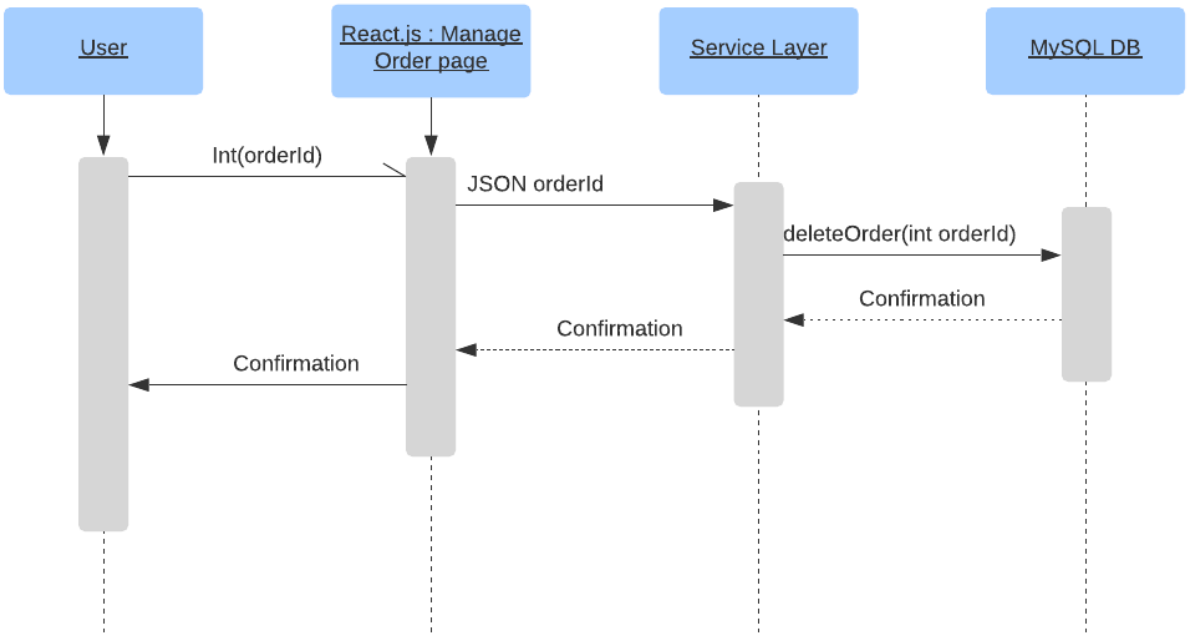
7.2 Manage Orders



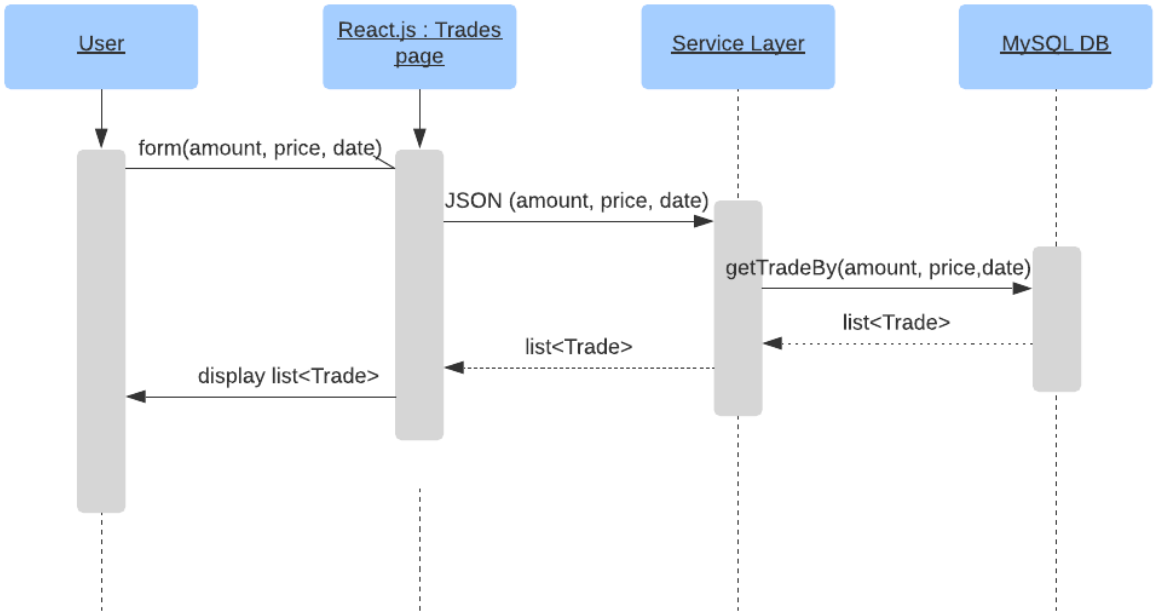
7.3 Edit Order



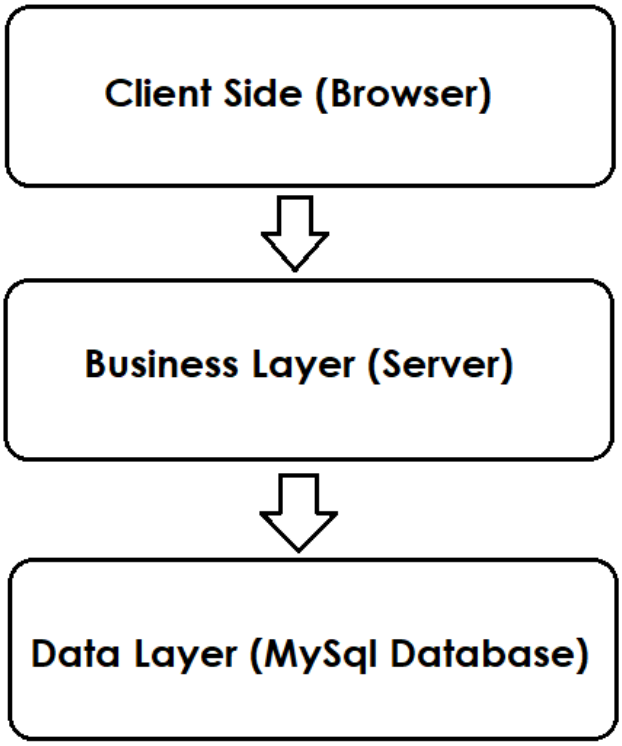
7.4 Delete Order



7.5 Search Trades



8. Implementation View



Katacoin Live is a web application that follows the MVC architecture pattern. This pattern separates functions into layers, improving maintainability and reusability.
The client side contains the graphical user interfaces (web pages). The actions of these web pages are handled by controller classes. Controller classes invokes and instantiates objects of model classes that contains the business logic. Separating software packages in this manner reduces the complexity.
The model classes can be subdivided in to two layers: business layer and data layer. Data layer is manipulated using Jdbc framework. Business layer contains the main entity classes such as User, Order and Trade. For each entity in the database there exists a class in the business layer that provides data access operations to that entity.

9. Size and Performance

Katacoin Live will be run off the localhost for demonstrative purposes. The user does not need to install programs to use it. The software is estimated around 100 MB. Performance will occur within the business layer, not on the client side. The client will, however, need a browser capable of displaying the page.

10. Quality

Refer to architectural goals and constraints (**2**).