|  |
| --- |
|  |

|  |
| --- |
| **Stock Market Charting v1.0 Project Design & UI Specifications** |
| Software analysis |
|  |
| This document covers Software Design and UI Specification of Stock Market Charting, along with list of some parts of UML diagrams, database structures, Screen flows and Wireframe in this Software System |
|  |
| **Associate: Chen Kun 陈堃** |
| **7/4/2020** |
|  |

Table of Contents

[1. UML Diagram 1](#_Toc37750709)

[3.1. Use Case Diagram 1](#_Toc37750710)

[3.2. Activity Diagram 2](#_Toc37750711)

[3.3. Deployment Diagram 4](#_Toc37750712)

[3.4. Class Diagram 4](#_Toc37750713)

[2. Database design 5](#_Toc37750714)

[3. Screen flows and Wireframe 9](#_Toc37750715)

[3.1. Investor Register screen flow 9](#_Toc37750716)

[3.2. Investor Comparison Company Chart 9](#_Toc37750717)

[3.4. Investor IPOs List (not required) 10](#_Toc37750718)

# UML Diagram

This Software System lets Admin to upload Stock Price of a Company(which is listed in a Stock Exchange) at different points of time. It need to support multiple Stock Exchanges. And

## Use Case Diagram

Admin can perform below operations. It is mandatory to implement all the requirements, except the ones mentioned as optional.

A picture containing text, map

Description automatically generated

## Activity Diagram

A User can perform below operations. It is mandatory to implement all the requirements, except the ones mentioned as optional.

A close up of a map

Description automatically generated

## Deployment Diagram

A close up of text on a white background

Description automatically generated

## Class Diagram

A close up of text on a white background

Description automatically generated

# Database design

Based on above analysis, I can give the draft version of DDL script for MySQL database.

Besides the normal columns to support our business logic, I also considered the possibility to adopt the database cluster for micro service architecture, so each table I added the column “db\_source” to indicate the data come from which data source provided to end users, that’s only for demonstrate in development environment and debugging purpose.

The wholly DDL script as following:

CREATE DATABASE cloudDB01;

use cloudDB01;

CREATE TABLE

users

(

id BIGINT NOT NULL AUTO\_INCREMENT,

name VARCHAR(60) NOT NULL,

password VARCHAR(60) NOT NULL,

email\_addr VARCHAR(50),

mobile\_number VARCHAR(20),

type enum('admin','investor') NOT NULL,

onetime\_check\_code VARCHAR(10),

confirmed BIT DEFAULT b'0' NOT NULL,

db\_source VARCHAR(10),

PRIMARY KEY (id),

CONSTRAINT uniqueId UNIQUE (name)

)

ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE DATABASE cloudDB03;

use cloudDB03;

CREATE TABLE

company

(

id BIGINT NOT NULL AUTO\_INCREMENT,

name VARCHAR(60) NOT NULL,

code VARCHAR(20) NOT NULL,

turnover VARCHAR(30) NOT NULL,

ceo VARCHAR(50),

board\_directors TEXT,

fk\_sector\_id bigint NOT NULL,

brief\_writeup TEXT,

active BIT DEFAULT b'1' NOT NULL,

db\_source VARCHAR(10),

PRIMARY KEY (id)

)

ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE DATABASE cloudDB04;

use cloudDB04;

CREATE TABLE

stock\_exchange

(

id BIGINT NOT NULL AUTO\_INCREMENT,

name VARCHAR(60) NOT NULL,

brief\_writeup TEXT,

contact\_addr VARCHAR(255) NOT NULL,

remarks TEXT,

db\_source VARCHAR(10),

PRIMARY KEY (id)

)

ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE DATABASE cloudDB05;

use cloudDB05;

CREATE TABLE

stock\_price

(

id BIGINT NOT NULL AUTO\_INCREMENT,

fk\_company\_id BIGINT NOT NULL,

fk\_stock\_exchange\_id BIGINT NOT NULL,

price DECIMAL(9.3),

unit VARCHAR(10),

date4price VARCHAR(30),

time4price VARCHAR(30),

db\_source VARCHAR(10),

PRIMARY KEY (id)

)

ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE

company2stock\_exchange

(

fk\_company\_id BIGINT NOT NULL,

fk\_stock\_exchange\_id BIGINT NOT NULL,

stock\_code VARCHAR(60),

db\_source VARCHAR(10),

CONSTRAINT uniqueId UNIQUE (stock\_code)

)

ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE DATABASE cloudDB02;

use cloudDB02;

CREATE TABLE

sectors

(

id BIGINT NOT NULL AUTO\_INCREMENT,

name VARCHAR(60) NOT NULL,

brief VARCHAR(255),

db\_source VARCHAR(10),

PRIMARY KEY (id)

)

ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE

ipo\_details --Initial Public Offering

(

id BIGINT NOT NULL AUTO\_INCREMENT,

fk\_company\_id BIGINT NOT NULL,

fk\_stock\_exchange\_id BIGINT NOT NULL,

price\_per\_share DECIMAL(9.3),

unit VARCHAR(10),

total\_number\_shares INT,

open\_datetime DATETIME,

remarks TEXT,

db\_source VARCHAR(10),

PRIMARY KEY (id)

)

ENGINE=InnoDB DEFAULT CHARSET=utf8;

# Screen flows and Wireframe

I listed almost all important wireframes design in this section.

## Investor Register screen flow

A screenshot of a cell phone

Description automatically generated

## Investor Comparison Company Chart

Please pay attention, my implementation only support two companies to compare in one chart at the same time.

A screenshot of a cell phone

Description automatically generated

* 1. **Investor Comparison Sector Chart (not required)**

Ignore

## Investor IPOs List (not required)

ignore

* 1. **Investor Company Search List**

**A screenshot of a cell phone

Description automatically generated**

* 1. **Investor profile Update**

**A screenshot of a cell phone

Description automatically generated**

* 1. **Admin Import Excel Data file**

**A screenshot of a cell phone

Description automatically generated**

* 1. **Admin Manage Company**

A screenshot of a cell phone

Description automatically generated

* 1. **Admin create an new Company (not required)**

Ignore

* 1. **Admin update IPO related data (not required)**

Ignore