

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО».

ОБЛАЧНЫЕ И ТУМАННЫЕ ВЫЧИСЛЕНИЯ

ПРОЕКТ

Выполнил: Карапетян Э. А.

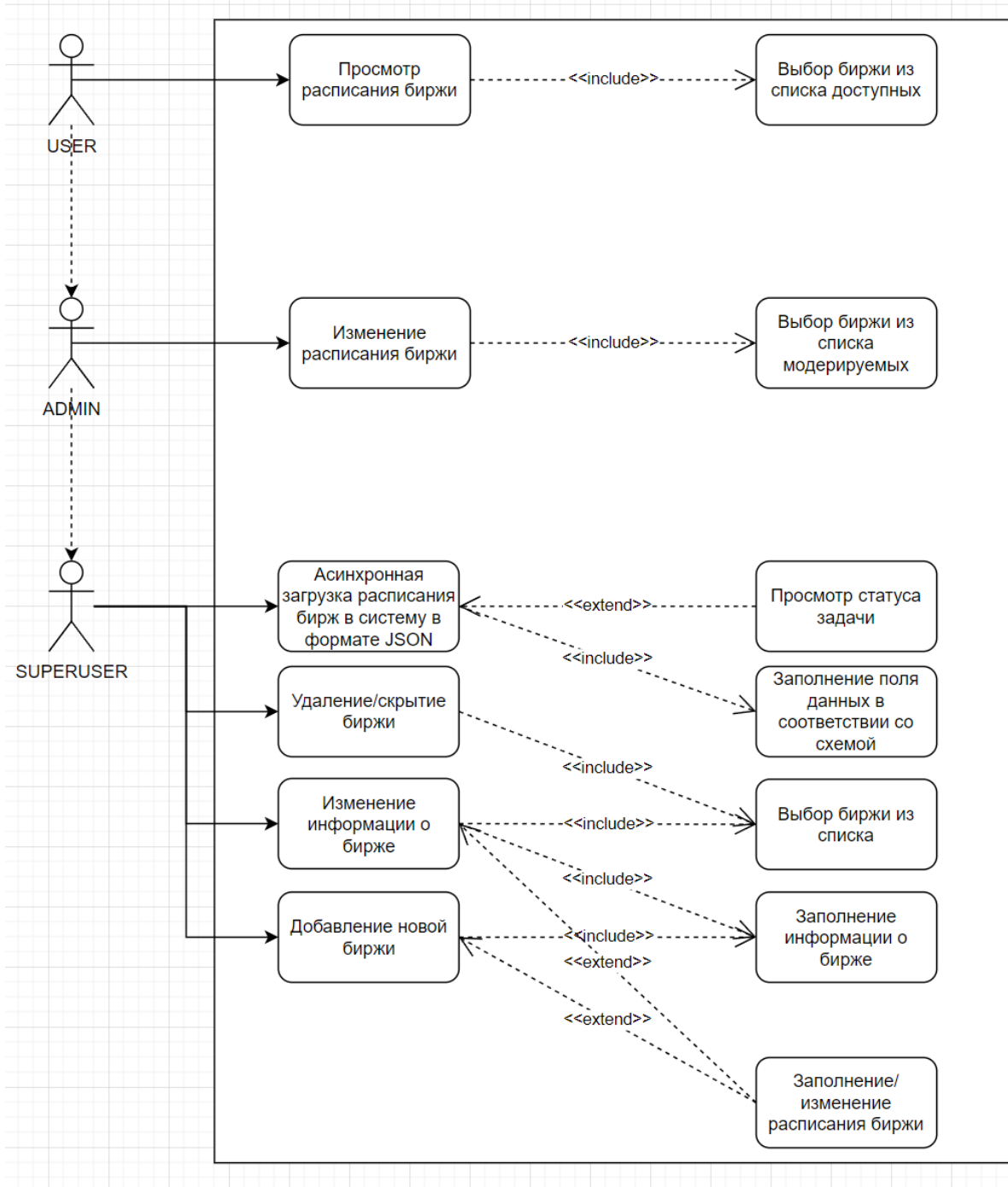
Преподаватель: Перл И.А.

Санкт-Петербург 2023 г.

Функциональные требования

1. Система должна предоставлять следующий набор прав:
 - 1.1. DEFAULT
 - 1.2. SCHEDULE_EDIT
 - 1.3. MARKET_EDIT
2. Система должна предоставлять следующий набор ролей с соответствующими им правами:
 - 2.1. USER (DEFAULT)
 - 2.2. ADMIN (DEFAULT, SCHEDULE_EDIT)
 - 2.3. SUPERUSER (DEFAULT, SCHEDULE_EDIT, MARKET_EDIT)
3. Система должна предоставлять возможность аутентификации и авторизации. Аутентификация должна производиться при помощи логина и пароля, после чего пользователю должен быть выдан JWT токен, который будет использоваться для авторизации.
4. Система должна автоматически приравнивать неавторизованного пользователя к пользователю с ролью USER.
5. Система должна предоставлять возможность просмотра списка бирж. Список бирж представляет собой плитки с логотипом и названием всех не скрытых бирж, зарегистрированных в системе с возможностью перехода на их страницу (DEFAULT).
6. Система должна предоставлять возможность просмотра расписания торгов бирж. Расписание включает в себя колонки по дням недели в которой будет информация о времени работе торгов в этот день в формате: “HH:mm:ss-HH:mm:ss” (DEFAULT).
7. Система должна предоставлять возможность ручного исправления расписания торгов биржи (SCHEDULE_EDIT).
8. Система должна предоставлять возможность добавления/удаления/скрытия биржи, изменения информации о ней (MARKET_EDIT).
9. Система должна предоставлять возможность асинхронной загрузки расписания бирж в систему в формате JSON с возможностью просмотра статуса задачи (MARKET_EDIT).

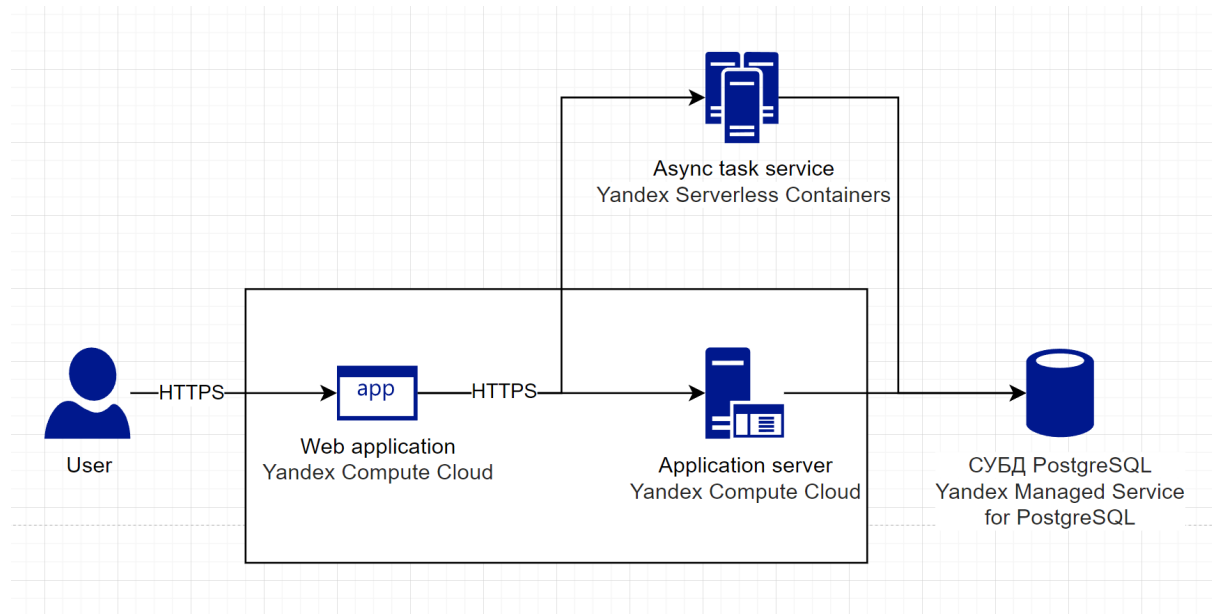
Use case диаграмма



Прецедент: просмотр расписания биржи
Краткое описание: Пользователь хочет узнать расписание торгов нужной ему биржи
Главный актер: USER
Второстепенные актеры: Нет
Предусловия: Пользователь знает название нужной ему биржи
Основной поток: <ol style="list-style-type: none"> 1. Пользователь находит нужную ему биржу в списке 2. Пользователь нажимает на плитку с этой биржей и переходит на ее страницу с информацией о расписании
Ожидаемый успешный результат: Пользователь перешел на страницу с информацией о бирже
Ожидаемый неуспешный результат: Биржи нет в списке/она скрыта, пользователь не перешел на страницу с информацией о бирже

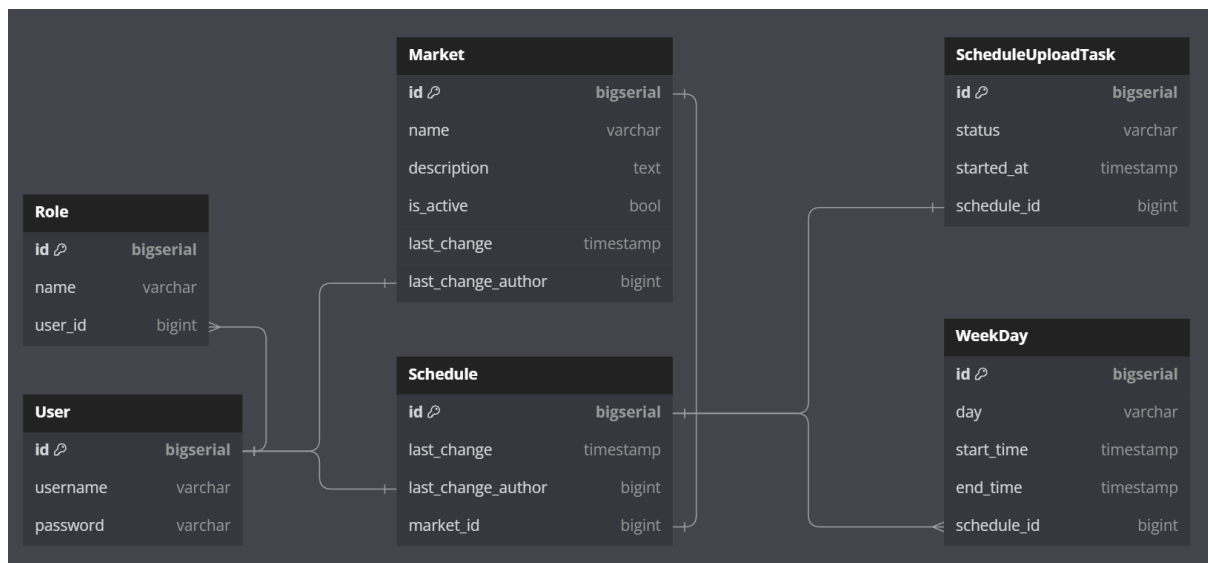
Прецедент: асинхронная загрузка расписания биржи в систему в формате JSON
Краткое описание: Пользователь хочет загрузить или обновить расписания некоторых бирж
Главный актер: SUPERUSER
Второстепенные актеры: Нет
Предусловия: Пользователь знает JSON схему, названия и расписания бирж для загрузки
Основной поток: <ol style="list-style-type: none"> 1. Пользователь вводит в текстовое поле необходимые данные в формате JSON в соответствии со схемой 2. Пользователь нажимает кнопку для загрузки данных в систему
Ожидаемый успешный результат: Система загрузила или обновила расписания торгов бирж
Ожидаемый неуспешный результат: Возникла ошибка при чтении данных (неправильная схема JSON, неверный формат даты/времени)

Архитектура системы



1. Web application - веб приложение с GUI пользователя с которым он взаимодействует по HTTPS.
2. Application server - сервер, к которому обращается Web application по HTTPS.
3. Async task service - сервер, к которому обращается Web application по HTTPS для выполнения асинхронных задач. Имеет возможность автоматического масштабирования при достижении определенной степени загрузки.
4. СУБД PostgreSQL - СУБД для хранения данных.

ERD



Целевая нагрузка системы

Система проектируется под пользователей, которые хотели бы заниматься биржами, однако имеют еще свои дела и работу. В постоянное время этим заниматься трудно, поэтому возникает вопрос, какие биржи сейчас доступны для торгов. Соответственно, пользователи не будут постоянно пользоваться данной системой, а будут проверять расписание несколько раз в день. По подсчетам НАУФОР в 2021 году общее количество россиян, открывших брокерские счета, по подсчетам НАУФОР, составляет 9,8 млн человек. Предположим 1% из них будут пользоваться данным сервисом регулярно 3 раза в сутки. Соответственно получаем 3 - 4 RPS.

Масштабирование

У системы есть возможность масштабирования, в частности использование Yandex Serverless Containers, которые имеют возможность автоматического масштабирования из коробки. Также можно увеличивать количество экземпляров Web application и Application server, добавив Load Balancer. СУБД PostgreSQL можно масштабировать также, добавив количество узлов. Вертикальное масштабирование будет применяться только в случае, если экземпляру какого-то компонента системы не будет хватать ресурсов для выполнения задачи (память, дисковое пространство). Горизонтальное масштабирование же будет более приоритетным при увеличении нагрузки на систему, так как является более простым и

предсказуемым. Масштабирование будет выполнено при серьезном увеличении нагрузки, вследствие которого система будет отвечать дольше, чем планируется. Изначально будет развернуто по одному экземпляру каждого компонента системы, со временем количество экземпляров будет инкрементироваться для достижения необходимого показателя эффективности - времени ответа сервера.