УДК 336.49

9. АВТОМАТИЗИРОВАННАЯ СИСТЕМА СБОРА, ХРАНЕНИЯ И РАСПРОСТРАНЕНИЯ ФИНАНСОВОЙ ИНФОРМАЦИИ

Суховаров А.Д., студент гр.072304

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Голда О.А. – ст. преподаватель каф. ЭИ

Аннотация. Актуальность разработки программной поддержки сбора, хранения и распространения финансовой информации обоснована растущим интересом пользователей к финансовой информации и управлению своими финансовыми потоками. Подобный сервис сможет занять свою нишу в экономической сфере, таким образом получив свое развитие и обеспечивая прибыть пользователям.

Ключевые слова. Финансовая информация, сбор, хранение, распространение, бизнес-процесс, база данных, Web-разработка, UX-элементы, язык программирования Python, Django Framework, архитектурные решения.

Интерес пользователей к финансовой информации и управлению своими финансовыми потоками постоянно растет. Однако ручной сбор и обработка финансовой информации требуют больших затрат времени и ресурсов. Каждый раз, когда необходимо обновить или передать финансовые данные, требуется значительное количество времени пользователей, что может снизить эффективность работы. Решение этих проблем достигается внедрением компьютерных систем, способных автоматизировать процессы сбора, хранения и распространения финансовой информации. Это позволит добиться высокой точности, актуальности и надежности данных, повысить эффективность работы, обеспечить безопасность и рационализировать процессы аналитики и отчетности.

Таким образом, целью работы является минимизация временных затрат пользователей при выборе инвестиционного инструмента за счет автоматизации процесса сбора, хранения и распространения финансовой информации.

Объектом исследования являются процессы сбора, хранения и распространения финансовой информации.

Пользовательский интерфейс системы реализуется на языке Python с помощью технологии Django Framework (см. рисунок 1, 2).

Основными требованиями к интерфейсу являются:

- интерфейс должен быть простым и легким в использовании даже для пользователей без технических навыков или финансового образования;
- разделение информации на логические категории и предоставление простого способа переключения между ними;
- наличие анимации для каждого элемента экрана, который меняет состояние или положение;
- цветовая гамма должна быть выполнена, используя три основных брендовых цвета ПП;
- навигация между экранами должна быть анимирована;
- содержимое экранов должно быть адаптировано для планшетов и горизонтальной ориентации экрана, где это возможно технически;
- экраны должны сохранять и восстанавливать свое состояние;
- производительность отрисовки стабильна все время пользования приложением.

Навигация происходит путем нажатия на кнопки в секции вверху экрана. Помимо указанных выше экранов, разработка пользовательского интерфейса потребовала также создание более 20 различных макетов. Таким образом, была проведена спецификация пользовательского интерфейса и определение основных назначений для навигации пользователя. Была сформирована дизайн-система и макеты интерфейса.

В соответствии с дизайн-стандартами, описанными выше, была реализована последовательность основных страниц приложения.

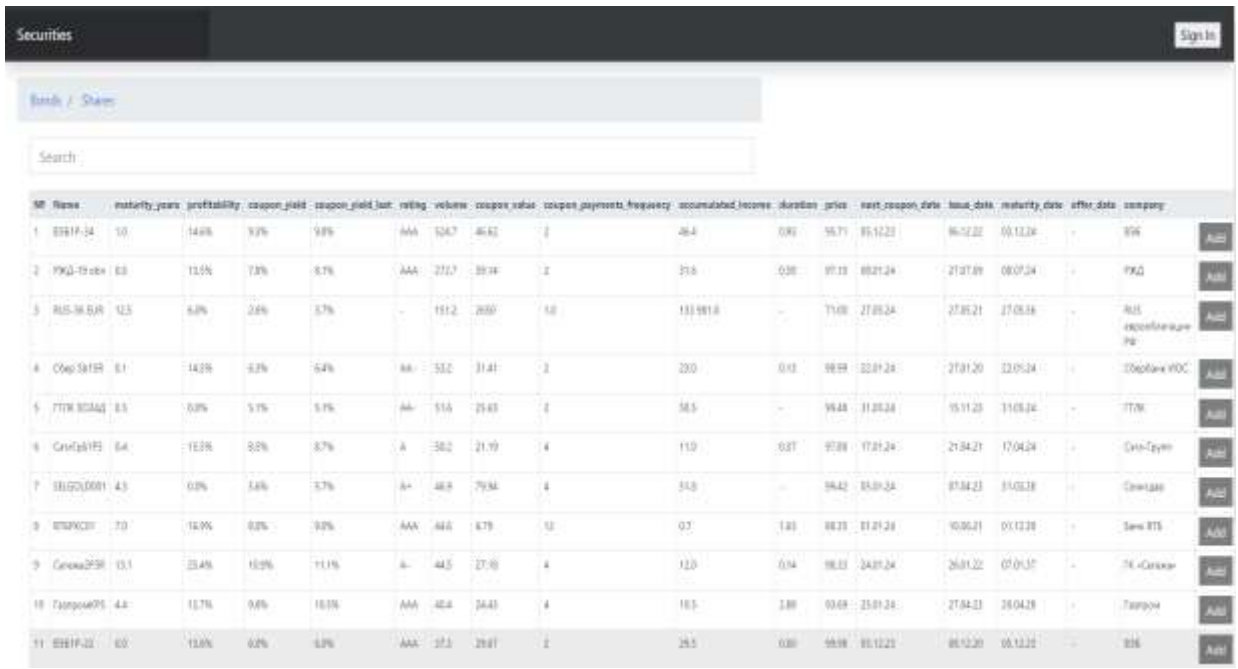

Рисунок 1 – Экран «Bonds»

Рисунок 2 – Экран «Share schedule»

Проект организован в соответствии с принципами Clean Architecture, а также модуляризован согласно последним стандартам Google. Выполнение длительных операций осуществляется на основе фоновой обработки задач библиотеки DramatiQ на базе Redis. Доступ к базе данных реализован с использованием асинхронных вызовов для соответствия стандартам производительности.

На рисунке 3 изображена диаграмма вариантов использования конечной системы. Ключевые варианты использования программного средства выглядят следующим образом:

- установка приложения;
- онбординг пользователя – вход в аккаунт, создание предпочтений, добавление ценных бумаг;
- онбординг главного экрана;
- просмотр деталей элемента корзины предпочтения;
- добавление акций, облигаций;

- импорт и экспорт данных;
- генерация отчетов по предпочтениям;
- уведомление о предпочтениях других пользователей;
- просмотр статистики и гистограмм.

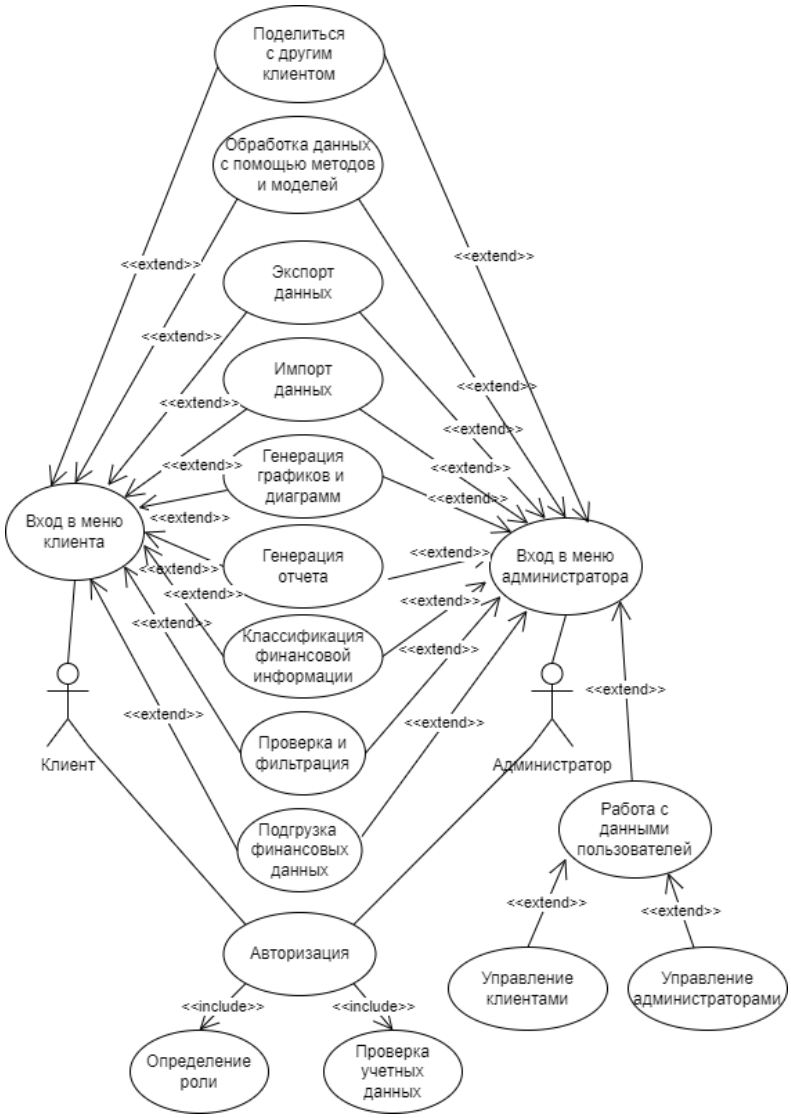

Рисунок 3 – Диаграмма вариантов использования программного средства

На рисунке 4 представлена диаграмма базы данных серверного приложения. На данной диаграмме можно заметить наличие всех основных таблиц, созданных при помощи Django ORM. Внутри Django приложения определяются модели, которые представляют таблицы в базе данных. Каждая модель соответствует отдельной таблице, а поля модели - столбцам в таблице. Определяются типы полей, их ограничения и связи между моделями (отношения).

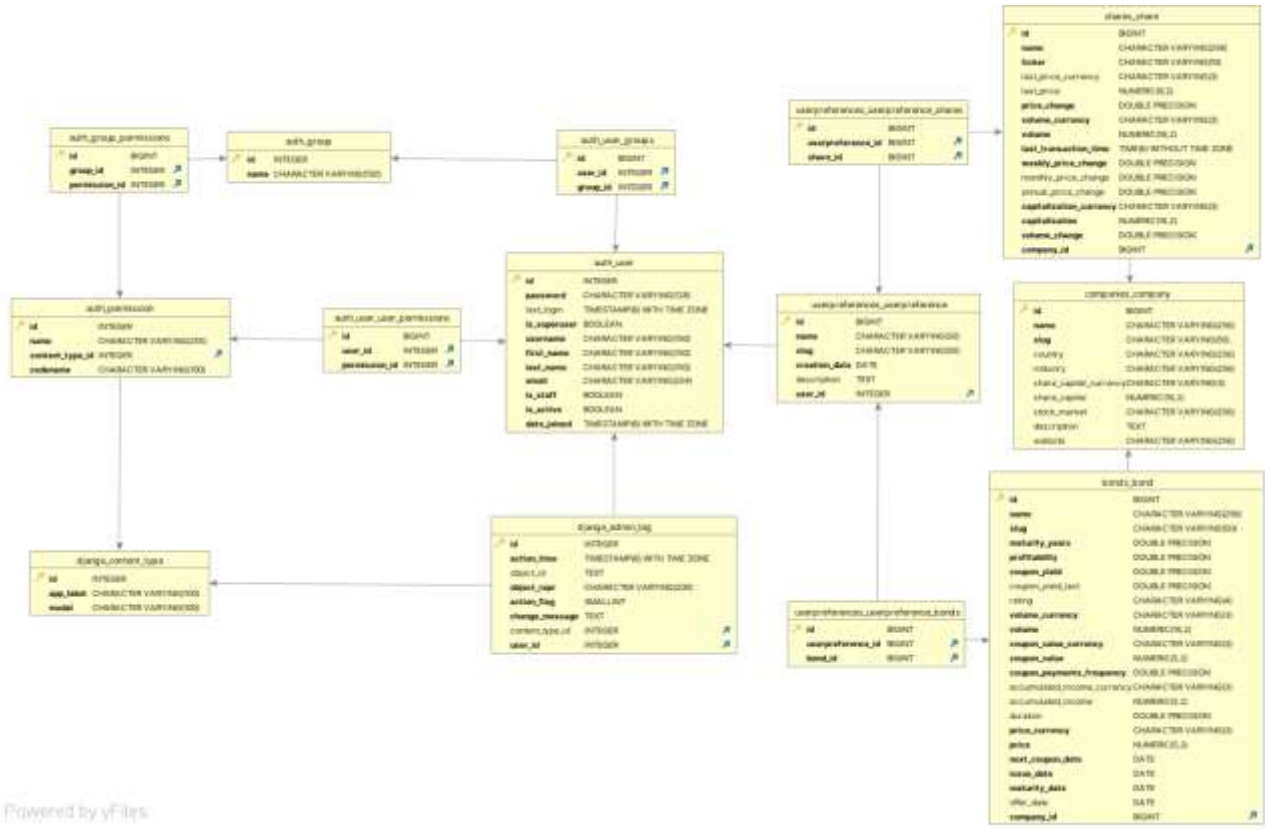

Рисунок 4 – Схема базы данных удаленного сервера

Помимо таблиц моделей, указанных на схеме выше, схема базы данных сервера включает в себя вспомогательные таблицы django_content_type и django_admin_log, использующихся для хранения публичных данных. Таблица django_content_type включает в себя текстовые данные о названии и значении ключей, тем самым предоставляя возможность в режиме реального времени редактировать поступающие значения.

Архитектура серверного приложения следует принципам поддержки неограниченной расширяемости и единства с принципами работы Web-приложения. Сервер выполнен на базе архитектуры MVC с модуляризацией по функционалу, что дает возможность оставить его монолитной структурой, а в дальнейшем перейти к использованию микросервисов, вынеся общий код в распространяемую мультиплатформенную библиотеку (фреймворк). Это обеспечит расширяемость кодовой базы. Производительность обеспечивается асинхронным, основанным на рабочих потоках механизмом выполнения операций (корутинах). Ввиду наличия более 500 классов в исходном коде конечного программного средства, а также более 20 модулей, более сотни зависимостей между ними, в данной статье диаграммы классов или модулей не приводятся.

Таким образом, для реализации требуемых спецификаций, описанных ранее, была разработана уникальная архитектура приложения, а также использованы рекомендуемые архитектурные практики и технологии.

Одной из причин подобных решений стало наличие значительного количества взаимодействующих друг с другом элементов, модулей, систем, исполняемых компонентов. Коммуникация приложения осуществляется не только с удаленным сервером, но и со сторонними сервисами, в основном предоставленными Google. Коммуникация с базой данных, репликация и синхронизация в реальном времени в фоновом режиме работы устройства также повлияли на комплексность системы. Диаграмма развертывания представлена на рисунке 5.

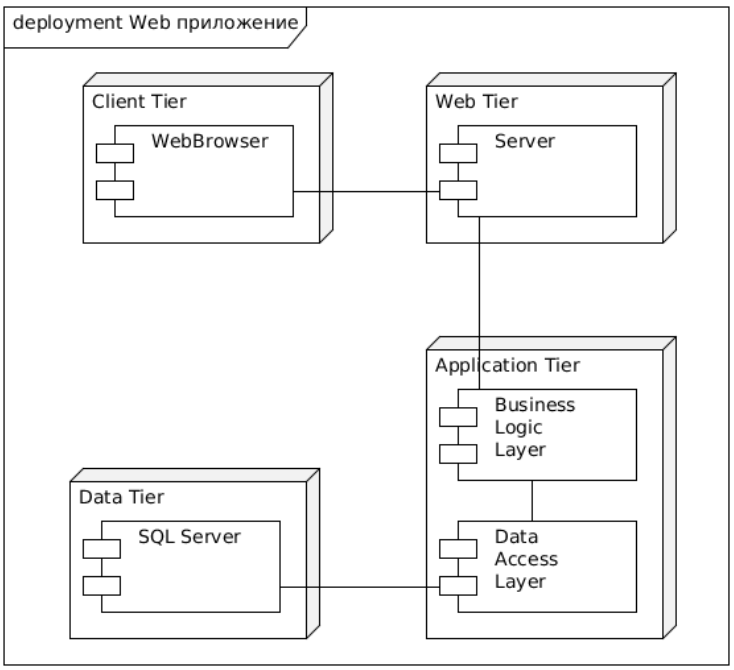

Рисунок 5 – Диаграмма развертывания

Подводя итог, была реализована парадигма (поведенческая модель) работы приложения и определены стандарты, которым впоследствии будет соответствовать программный код приложения для оптимизации скорости разработки, стабильности, и производительности системы.

Создана программная реализация, которая обеспечивает быстрый доступ к актуальной информации, позволяет пользователям выбирать инструменты для инвестирования и отслеживать их эффективность во времени.

Было проведено как автоматизированное, так и ручное тестирование ключевых компонентов приложения, а также реализована как удаленная, так и локальная валидация содержимого, генерируемого пользователями, что удовлетворяет установленным спецификациям. Описанный функционал был реализован в полном объеме как с точки зрения функциональных, технических и бизнес-требований, так и с точки зрения пользовательского интерфейса приложения.

Разработка минимизирует время пользователя на подбор базовых активов, позволяет сохранять их в избранном и отслеживать по времени, сравнивать избранные активы по разным характеристикам и визуализировать их, а также делиться собранным портфелем активов с другими пользователями.

Список использованных источников:

1. Рынок ценных бумаг: учебник для М.: Издательство Юрайт, 2016. да, академического — 443 с

2. Финансовый рынок: Рынок ценных бумаг: М.: НИЦ ИНФРА -М, 2014. - да, 4 Учебное пособие / И.В. 281 с.

3. Фаулер, М. Архитектура корпоративных программных приложений / М. Фаулер. – М.: Вильямс, 2016 – 544 с.