

**ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ АРХИТЕКТУРЫ
ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ УПРАВЛЕНИЯ СКЛАДСКИМИ
ПРОЦЕССАМИ С ИСПОЛЬЗОВАНИЕМ БЛОКЧЕЙН-ПОДХОДА**

Васильев С.А. (Университет ИТМО)

Научный руководитель – кандидат физико-математических наук, доцент

Ромакина О.М. (Университет ИТМО)

Введение. В складском управлении важно быстро и обоснованно назначать задачи сотрудникам смены с учётом квалификаций и текущей загрузки, а также иметь верифицируемый аудит бизнес-событий для разбора инцидентов и анализа причин отклонений [1]. Обычно назначение строится на простых правилах, а аудит сводится к журналированию без механизмов проверки целостности [2]. Цель работы – разработать интеллектуальный сервис поддержки принятия решений для назначения складских задач, объединяющий доменно-ориентированный контур назначения, резервный сценарий и криптографически проверяемый аудит событий.

Основная часть. Прототип системы реализован на Python/Django и включает доменную модель сотрудников, смен, задач и складских сущностей. Доменные ограничения (квалификации, доступность, переходы статусов, согласованность зон и размещения) реализованы в сервисном слое приложения. Регламентные действия (перенос задач из пула в активные смены и периодические процедуры) выполняются через Celery и Celery Beat, что позволяет масштабировать фоновые процессы [3].

Ключевое отличие предлагаемой системы от аналогов – ML-ранжирование кандидатов из числа сотрудников для назначения на задачу. Среди кандидатов, прошедших доменный предфильтр (квалификации, доступность), модель CatBoost прогнозирует ожидаемое время выполнения задачи конкретным исполнителем в минутах; кандидаты ранжируются по возрастанию прогнозируемого времени, и наиболее предпочтительным считается кандидат с минимальным прогнозом [4]. Для устойчивости предусмотрен резервный сценарий: при недоступности модели, недостатке данных или нарушении ограничений назначение выполняется эвристически.

Аудит фиксирует бизнес-события как AuditEvent со снимками «до/после». Для детерминированного хэширования применяется канонизация JSON [5]; события агрегируются в блоки с вычислением корня Меркла и связываются в цепочку через хэш предыдущего блока. Функция verify_chain пересчитывает хэши и выявляет постфактум изменения. Оценка на синтетической истории операций показала MAE = 4,551 мин и RMSE = 6,283 мин против базовой эвристики (MAE = 12,500 мин, RMSE = 15,725 мин).

Выводы. Разработан прототип интеллектуального сервиса назначения задач, где ML-ранжирование интегрировано в доменно-ориентированный контур с проверкой ограничений и резервным сценарием, а аудит событий реализован в виде криптографически проверяемой цепочки блоков. Предлагаемый подход повышает воспроизводимость и прозрачность назначения сотрудников на задачи и наблюдаемость операций. Дальнейшее развитие разработки предполагает накопление реальных логов и оценку влияния работы системы на ключевые показатели эффективности склада.

Список использованных источников:

1. De Koster R., Le-Duc T., Roodbergen K. J. Design and control of warehouse order picking: a literature review // European Journal of Operational Research. – 2007. – Т. 182. – №

2. – С. 481–501. DOI: 10.1016/j.ejor.2006.07.009. URL: <https://doi.org/10.1016/j.ejor.2006.07.009> (дата обращения: 14.02.2026).

2. Crosby S. A., Wallach D. S. Efficient Data Structures For Tamper–Evident Logging // Proceedings of the 18th USENIX Security Symposium. – 2009. – С. 317–334. URL: https://www.usenix.org/legacy/event/sec09/tech/full_papers/crosby.pdf (дата обращения: 14.02.2026).

3. Celery. Periodic Tasks – Celery documentation [Электронный ресурс]. URL: <https://docs.celeryq.dev/en/main/userguide/periodic-tasks.html> (дата обращения: 14.02.2026).

4. Prokhorenkova L., Gusev G., Vorobev A., Dorogush A. V., Gulin A. CatBoost: unbiased boosting with categorical features // Advances in Neural Information Processing Systems 31 (NeurIPS 2018). – 2018. – С. 6638–6648. URL: https://papers.nips.cc/paper_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf (дата обращения: 14.02.2026).

5. Rundgren A., Jordan B., Erdtman S. JSON Canonicalization Scheme (JCS). RFC 8785. – June 2020. – [Электронный ресурс]. DOI: 10.17487/RFC8785. URL: <https://www.rfc-editor.org/rfc/rfc8785> (дата обращения: 14.02.2026).