Нижче наведено поради та пропозиції щодо організації **фінансового блоку** у вашій системі. З урахуванням ваших вимог, почнемо з **мінімалістичної** реалізації виставлення рахунку, а потім розглянемо **перспективу розширення** цього модуля.

**1. Мінімальна (MVP) реалізація**

1. **Формування рахунку (Invoice) у додатку**
   * Після завершення робіт (або узгодження замовлення), система автоматично чи за запитом менеджера генерує “рахунок” для клієнта.
   * У рахунку вказуються:
     + Назва/логотип СТО, реквізити для оплати (IBAN, ЄДРПОУ/ІПН, або інші платіжні дані).
     + Перелік робіт і запчастин із сумами.
     + Підсумкова сума до оплати.
2. **Копіювання реквізитів і QR-код**
   * Для зручності клієнт бачить кнопку “Копіювати реквізити”, яка дає змогу швидко вставити їх у банківський додаток.
   * Додатково можна згенерувати **QR-код** (наприклад, за стандартом ЄП (якщо в країні прийнято), або за допомогою простого сервісу), щоб клієнт міг відсканувати його мобільним банківським додатком і перейти до оплати.
3. **Статус оплати (Manual)**
   * Менеджер відзначає в системі, що клієнт сплатив рахунок (отримав квитанцію чи інше підтвердження).
   * Система позначає статус замовлення як “Оплачено”.
   * На цьому етапі не обов’язково реалізовувати повну автоматичну інтеграцію з банками чи платіжними шлюзами — все відбувається вручну, але вже є базова зручність для клієнта.
4. **Збереження рахунку/чека**
   * Можна реалізувати просте **формування PDF** (інвойса) й зберігати його в хмарі (AWS S3 чи інше) із посиланням у базі даних. Клієнт може скачати цей PDF, роздрукувати чи переслати.

**2. Майбутнє розширення функціоналу**

1. **Відслідковування заборгованості**
   * Коли система зберігає всі виставлені рахунки й статуси “Оплачено/Не оплачено”, менеджери можуть швидко переглянути, хто з клієнтів має непогашені борги.
   * Можна реалізувати **внутрішню звітність**: сума виставлених рахунків за період, сума оплачених, сума прострочених боргів і т.д.
2. **Формування актів виконаних робіт і рахунків-фактур**
   * Для більш офіційної документації (наприклад, для юридичних осіб) важливо генерувати “Акт виконаних робіт” із підписами та печатками (у PDF-форматі), а також “Рахунок-фактуру”, якщо вимагає локальне законодавство.
   * Це можна додати окремим модулем, де:
     + Система підтягує дані про виконані роботи (робочі години, запчастини).
     + Додає реквізити СТО й клієнта.
     + Генерує документ, готовий для друку чи надсилання електронною поштою.
3. **Оплата онлайн (інтеграція з платіжними системами)**
   * Під’єднати платіжні системи (LiqPay, Stripe, PayPal тощо), щоб клієнти могли сплатити в один клік прямо з додатка.
   * Після успішної транзакції система **автоматично змінює статус** рахунку на “Оплачено”.
   * Реалізується через REST API/SDK платіжного провайдера й вебхуки (webhooks), які надсилають інформацію про успішну оплату.
4. **Бухгалтерський облік**
   * Якщо СТО потребує повноцінного бухгалтерського обліку (облік витрат, доходів, нарахування зарплат, податків тощо), можна **інтегруватися з 1С** (як згадували раніше) або з іншим бухгалтерським ПЗ.
   * Можна вивантажувати дані про виставлені рахунки/оплати в зовнішню систему бухобліку, щоб бухгалтер мав усю інформацію централізовано.

**3. Приклад базової структури (Mongoose, Express) для MVP**

js

CopyEdit

// models/Invoice.js

const mongoose = require('mongoose');

const invoiceSchema = new mongoose.Schema({

invoiceNumber: { type: String, required: true }, // Унікальний номер рахунку

orderId: { type: mongoose.Schema.Types.ObjectId, ref: 'Order', required: true },

clientId: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

items: [

{

description: String,

quantity: Number,

price: Number,

total: Number

}

],

totalAmount: { type: Number, required: true },

status: { type: String, enum: ['PENDING', 'PAID'], default: 'PENDING' },

createdAt: { type: Date, default: Date.now },

paidAt: { type: Date },

// Реквізити СТО (можна зберігати в довіднику чи в налаштуваннях)

stoRequisites: {

name: String,

iban: String,

edrpou: String,

// інші поля...

}

// ...

});

module.exports = mongoose.model('Invoice', invoiceSchema);

* items: коротка деталізація (наприклад, назви робіт, запчастин) та суми.
* status: відображає стан рахунку (оплачений чи ні).
* invoiceNumber: корисно формувати по шаблону, наприклад, INV-2025-0001.
* stoRequisites: можна зберігати у глобальному місці, а сюди підтягувати при створенні рахунку, щоб мати “історичні” дані (якщо реквізити зміняться пізніше).

**4. Як виглядатиме процес у користувача (MVP)**

1. **Менеджер чи майстер завершує замовлення** → система формує в “Замовленні” перелік виконаних робіт і запчастин.
2. **Натискає кнопку “Створити рахунок”** → у backend створюється запис в Invoice із потрібними даними.
3. **Клієнт бачить рахунок** (наприклад, у своєму кабінеті) і може:
   * Скопіювати реквізити;
   * Згенерувати QR-код і відкрити свій банківський додаток;
   * (Якщо реалізовано) натиснути “Оплатити онлайн” і перейти до платіжної системи.
4. **Менеджер перевіряє оплату** (manual) і ставить статус PAID.
   * Система може відправити підтвердження клієнту (Telegram, Viber, e-mail).

**5. Підсумок**

1. **MVP-фокус**: просте виставлення рахунку, копіювання реквізитів, QR-код, ручна зміна статусу на “Оплачено”.
2. **Подальше розширення**:
   * Автоматизоване відстеження заборгованості та нагадування.
   * Формування актів виконаних робіт, інтеграція з бухгалтерськими системами.
   * Можливість онлайн-оплати та автоматичне оновлення статусу.
3. **Врахування місцевих вимог**: якщо у вашій юрисдикції є специфічні правила оформлення документів (підписи, печатки, електронний цифровий підпис), можна додати їх на наступних етапах.