

Test Scenario Document: Application Setup and Installation

Document ID: TS-SETUP-001 Version: 1.0 Date: December 2025 Author: Test Team

1. PURPOSE

This document describes the test scenario for setting up and installing the E-commerce Application. This includes database setup, application configuration, and initial environment preparation.

2. PREREQUISITES

- Windows 10/11 or compatible OS
- Microsoft SQL Server Express installed
- Python 3.8 or higher
- Network access to SQL Server instance
- Administrative privileges for installation

3. SETUP STEPS

3.1 Database Setup - Install Microsoft SQL Server Express - Connect using SQL Server Management Studio - Use credentials: sa / student - Execute the database_schema.sql script - Create database 'app1' - Create application user 'app1user' with password 'student' - Grant db_owner permissions to app1user

3.2 Application Environment Setup - Install Python 3.8+ - Install required packages: pip install pyodbc - Create project directory structure: - /src - /config - /data/import - /reports - /docs

3.3 Configuration Setup - Copy config.json to project root - Update server name in config.json (replace PC000 with actual server name) - Verify database credentials match the created user

3.4 Application Deployment - Copy all source files to project directory - Verify all modules can be imported without errors

4. VERIFICATION STEPS

4.1 Database Connection Test - Run basic_connection.py to test database connectivity - Expected result: "Připojeno." message appears without errors

4.2 Schema Verification - Verify all 6 tables exist in database: Customers, Categories, Products, Orders, OrderItems, Suppliers - Verify 2 views exist: CustomerOrderSummary, ProductSalesSummary - Verify M:N relationships exist: OrderItems (Orders-Products), ProductSuppliers (Products-Suppliers)

4.3 Application Startup Test - Run console__ui.py - Expected result: Application menu displays without errors

5. SUCCESS CRITERIA

- Database connection established successfully
- All tables and views created as per schema
- Application starts without errors
- All required modules can be imported
- Configuration file is properly loaded

6. FAILURE CONDITIONS

- Database connection fails
- Schema creation errors
- Missing dependencies
- Configuration loading errors

7. ERROR HANDLING

- If database connection fails, verify server name and credentials
- If tables are missing, re-run the schema script
- If Python modules are missing, install required packages
- If configuration is invalid, check config.json format and values

8. POST-CONDITIONS

- Database 'app1' exists with proper schema
- Application can connect to database
- All configuration settings are validated
- Application is ready for functional testing