

Sound Concept of Software Architecture

1 Wahl des Betriebssystems: Windows oder Linux vs. Embedded Systeme

Für unser Kassensystem haben wir uns für ein vollwertiges Betriebssystem wie Windows oder eine Linux-basierte Distribution entschieden. Diese Wahl bietet mehrere Vorteile gegenüber einem reinen Embedded System. Erstens bieten Windows und Linux eine größere Flexibilität bei der Integration und dem Betrieb der Software. Dazu gehört die Unterstützung einer Vielzahl von Hardwarekomponenten und Peripheriegeräten, die für den Betrieb eines Kassensystems erforderlich sind, wie z.B. Drucker, Barcode-Scanner und Zahlungsterminals.

Ein weiteres Argument gegen ein eingebettetes System ist die eingeschränkte Skalierbarkeit und Erweiterbarkeit. Ein vollwertiges Betriebssystem bietet eine robustere Plattform für die Entwicklung und Implementierung von Softwarelösungen, die leicht aktualisiert und erweitert werden können. Zudem profitieren wir von der breiten Unterstützung durch die Open-Source-Community (im Falle von Linux) oder von umfassenden kommerziellen Supportleistungen (im Falle von Windows).

2 Objektorientierte Programmierung in C++

Die Entscheidung, objektorientierte Programmierung (OOP) mit C++ zu verwenden, basiert auf der Notwendigkeit, ein robustes, wartbares und erweiterbares System zu entwickeln. OOP ermöglicht einen modularen Aufbau des Codes, bei dem die einzelnen Komponenten des Kassensystems wie Benutzeroberfläche, Datenverarbeitung und Hardwareschnittstellen klar voneinander getrennt und unabhängig voneinander entwickelt werden können. Dies erleichtert nicht nur die Wartung und Weiterentwicklung der Software, sondern auch das Testen und Debuggen.

Die Programmiersprache C++ bietet zudem eine hohe Performance und Effizienz, was insbesondere bei Systemen mit Echtzeitanforderungen, wie z.B. einem Kassensystem, von großer Bedeutung ist. Die umfangreichen Standardbibliotheken und die breite Unterstützung durch bestehende Frameworks und Tools machen C++ zur idealen Wahl für unser Projekt.

3 Firmware und Bibliotheken für Schnittstellen

Ein zentrales Element unseres Designs ist die Notwendigkeit, für alle relevanten Schnittstellen bereits existierende Firmware und entsprechende Bibliotheken in C/C++ zur Verfügung zu haben. Dazu gehören Waagen, Barcode-Scanner, Zahlungsterminals und andere Peripheriegeräte, die für den Betrieb des Kassensystems erforderlich sind. Durch die Verwendung von standardisierter und gut dokumentierter Firmware sowie vorhandener Bibliotheken können wir sicherstellen, dass alle Hardwarekomponenten nahtlos und zuverlässig in das System integriert werden, ohne diese selbst implementieren zu müssen.