

제 안 요 청 서

소프트웨어 공학 프로젝트:
POS 기 시스템

2018060646 이한승

2019031858 김재명

2021030873 이동오

2019008813 문원찬

2021051321 가제고 달린

소개

POS (Point of sale) 시스템은 매장 운영에 매우 중요한 역할을 합니다. 현금, 카드, 모바일 결제 등 다양한 결제 방식을 편리하고 안전하게 처리하여 고객이 쉽게 구매할 수 있도록 도와주는 시스템입니다. 결제 외에도 POS 시스템은 거래를 효율적으로 관리하고 제품 명세와 결제 방법을 기록합니다. 이 거래 데이터는 실시간 재고 관리에 도움이 되어 과잉 재고를 방지합니다. 따라서 POS는 사업을 더 효과적으로 운영할 수 있도록 도와주는 시스템이라고 할 수 있습니다.

목표

1. 직원과 고객 모두에게 효율성 향상:

POS 시스템은 직원과 고객 모두에게 효율성을 향상시키는데 중요한 역할을 합니다. 직원은 거래를 빠르게 처리할 수 있도록 간편한 사용자 인터페이스와 효율적인 기능을 활용할 수 있습니다. 이로써 결제 및 거래 처리 시간이 단축되어 고객들은 금방 결제를 완료할 수 있습니다.

2. 재고 관리 향상:

POS 시스템은 실시간으로 재고를 관리합니다. 제품이 판매될 때마다 시스템은 재고에서 해당 제품을 자동으로 차감합니다. 따라서 제품의 과잉 재고나 부족을 방지할 수 있습니다. 결과적으로, 재고 관리 향상은 관리 비용 절감과 제품 유통의 효율성을 증가시킬 수 있습니다.

3. 거래 처리 단순화:

POS 시스템은 거래 처리를 단순화하여 직원과 고객 모두에게 편의를 제공합니다. 예를 들어, 고객은 다양한 결제 수단을 편리하게 사용할 수 있으며, 직원은 시스템이 자동으로 총액과 잔돈을 계산하여 계산 오류를 줄여줍니다.

대상고객

POS 시스템은 다양한 사업 분야에서 사용할 수 있는 유연한 솔루션으로써 슈퍼마켓, 커피숍, 편의점, 소매점, 식당, 그리고 일시적인 판매자를 대상으로 합니다. 슈퍼마켓과 편의점에서는 고객이 다양한 상품을 쉽게 구매할 수 있도록 빠르고 안정적인 결제가 필요하며 커피숍과 식당에서는 주문 및 지불이 효율적이어야 합니다. 소매점과 일시적인 판매자는 재고를 효과적으로 관리하고 판매 데이터를 추적하기 위해 신속한 POS 시스템을 필요합니다. 이러한 다양한 사업에서 POS 시스템을 활용함으로써 사업 운영을 최적화하고 고객 서비스를 향상시킬 수 있습니다.

기능 및 특징

- **제품 데이터베이스 및 재고관리:**

이 시스템은 제품에 대한 정보를 저장하고 실시간으로 재고를 추적합니다. 이것은 상품을 추가하거나 제거하고 재고 수준을 실시간으로 업데이트하는데 사용됩니다. 이를 통해 상품의 다양성과 가용성을 관리할 수 있고 고객에게 실시간으로 정보를 제공할 수 있습니다.

- **결제 처리:**

POS 시스템은 다양한 결제 수단을 편리하게 처리합니다. 고객은 현금, 신용카드 또는 모바일 지불로 결제할 수 있으며, 이러한 다양한 결제 방법을 손쉽게 사용할 수 있습니다.

- **현금 처리 및 잔돈 계산:**

현금 거래 시스템은 정확한 금액을 계산해 잔돈을 반환할 수 있도록 도와줍니다. 이로써 빠르고 정확한 현금 거래가 가능하며 오류를 최소화합니다.

- **매출 보고:**

POS 시스템은 매출과 관련된 정보를 자동으로 기록할 수 있습니다. 이를 통해 사업 주인은 판매 통계를 쉽게 확인하고 사업 운영을 최적화할 수 있습니다.

- **영수증 만들기:**

시스템은 거래를 기록하고 영수증을 생성합니다. 영수증은 거래 내역을 명확하게 보여주어 고객에게 투명성을 제공합니다.

사용할 도구 및 자원

1. Development Environment:

Visual Studio Code: 소스 코드 편집 및 개발 환경으로 Visual Studio Code를 사용할 예정입니다. 이 편집기는 다양한 언어와 환경을 지원하는 확장 기능을 제공하여 본 프로젝트의 다양한 요구사항을 만족시키기에 적합합니다.

2. Version Control:

Git: 소스 코드의 버전 관리를 위해 Git을 사용합니다. 이를 통해 코드 변경 사항의 추적 및 여러 팀원 간의 협업이 용이합니다.

GitHub: Git 리포지토리 호스팅 및 코드 리뷰, 이슈 트래킹 등의 기능을 제공하는 온라인 플랫폼으로, 프로젝트의 중앙 리포지토리로 사용될 예정입니다.

3. Programming Language:

JavaScript: 프로젝트의 주요 프로그래밍 언어로서, 웹 기반 애플리케이션 개발에 사용됩니다.

4. Database:

MongoDB: NoSQL 데이터베이스로서, 유연한 스키마와 확장성을 제공합니다. 프로젝트 데이터 관리 및 저장에 사용될 예정입니다.

5. Communication Tools:

Zoom: 원격 회의 및 팀원 간의 커뮤니케이션 도구로 사용됩니다.

KakaoTalk: 실시간 메시지 전송 및 간편한 알림을 위한 메신저로 사용됩니다.

6. Cloud Services:

AWS (Amazon Web Services): 클라우드 호스팅 및 기타 서비스 제공을 위한 플랫폼으로, 프로젝트의 배포 및 운영에 사용될 예정입니다.

Challenge

- **하드웨어 호환성 문제:**

POS 시스템의 하드웨어 호환성 문제는 다양한 측면에서 발생할 수 있습니다. 이 도전과제를 극복하기 위해 각 하드웨어 구성 요소와의 호환성을 테스트하고, 필요한 경우 드라이버나 소프트웨어 업데이트를 적용하여 해결책을 찾아야 합니다.

- ✓ **프린터 호환성:** POS 시스템은 영수증을 출력하기 위해 프린터를 사용합니다. 다양한 제조사의 프린터들이 다른 프로토콜이나 드라이버를 사용할 수 있어 호환성 문제가 발생할 수 있습니다.
- ✓ **스캐너 및 리더 호환성:** 바코드 스캐너, 카드 리더, NFC(Near Field Communication) 리더 등 다양한 장비들이 POS 시스템과 호환되어야 합니다. 특히, 통신 프로토콜이나 카드 형식 등에서 문제가 발생할 수 있습니다.
- ✓ **화면 해상도 및 크기:** POS 시스템의 화면이나 터치스크린의 해상도 및 크기가 다를 경우, 소프트웨어 인터페이스가 올바르게 표시되지 않을 수 있습니다.
- ✓ **통신 인터페이스 호환성:** POS 시스템이 다른 하드웨어와 통신해야 할 경우, 통신 인터페이스의 호환성 문제가 발생할 수 있습니다. USB, Ethernet, 시리얼 등 다양한 통신 방식이 쓰이는데, 이에 대한 호환성을 고려해야 합니다.
- ✓ **보안 인증 및 암호화 요구사항:** 결제 시스템에는 높은 수준의 보안이 필요하므로, 특정 하드웨어가 보안 요구사항을 충족하지 못하는 경우 호환성 문제가 발생할 수 있습니다.

- **비기술적 사용자를 위한 쉽게 이용할 수 있는 인터페이스 설계:**

POS 시스템의 인터페이스를 비기술적 사용자가 쉽게 이용할 수 있도록 설계하는 것은 중요합니다. 따라서 사용자 편의성과 직관성이 중요합니다. 사용자들이 쉽게 학습하고 사용할 수 있는 인터페이스를 설계하는 것은 도전적인 과제입니다. 비기술적 사용자를 고려한 POS 인터페이스 설계는 사용자 경험을 향상시키고 오류를 최소화하여 효율성을 증가시킵니다. 이는 매출 증가와 함께 직원과 고객 모두에게 긍정적인 영향을 미칠 수 있습니다.

- ✓ **직관적 디자인:** 인터페이스는 직관적이어야 합니다. 사용자가 어떻게 상품을 찾고 결제하는지에 대한 프로세스가 명확해야 합니다.
- ✓ **단순한 내비게이션:** 메뉴와 기능은 단순하게 구성되어야 합니다. 사용자가 원하는 기능을 적은 클릭이나 터치로 달성해야 합니다.
- ✓ **큰 아이콘과 버튼:** 화면에 크고 명확한 아이콘과 버튼을 사용하여 사용자가 쉽게 식별하고 터치할 수 있도록 합니다.
- ✓ **반복적 프로세스 간소화:** 반복되는 작업이나 프로세스는 최소화되어야 합니다. 자주 사용되는 기능에는 빠른 단축키나 간편한 접근 방법을 제공합니다.
- ✓ **피드백 제공:** 사용자의 행동에 대한 피드백을 즉각적으로 제공하여, 사용자가 무엇을 어떻게 해야 하는지 이해할 수 있도록 합니다.

- **소프트웨어 개발 경험이 없는 구성원이 있다:**

프로젝트 팀 내에 소프트웨어 개발 경험이 없는 구성원이 있습니다. 이들이 효과적으로 프로젝트에 기여하도록 지원할 필요가 있습니다. 이런 구성원을 위한 교육 프로그램을 제공하거나, 지원을 해줄 수 있습니다. 또는 다른 팀원들과의 지속적인 협력을 통해 지식 및 기술을 공유하여 이 도전과제를 극복할 수 있습니다. 팀의 협력과 지원을 통해 모든 구성원이 프로젝트의 성공에 기여할 수 있습니다.

- ✓ **교육 프로그램:** 기본적으로 각자 가진 소프트웨어 개념에서 프로젝트와 관련된 자료를 제공할 수 있습니다. 예를 들면, 프로젝트 관리 도구나 협업 도구의 사용법을 설명하거나 어떻게 사용하는지 보여줄 수 있습니다.
- ✓ **협력과 지원:** 경험이 풍부한 팀원이 상대적으로 경험이 부족한 팀원으로부터 질문에 답하거나 도움을 줄 수 있게 유도할 수 있습니다. 이같이 어느 팀원이 뒤처지는 것을 방지해야 합니다.
- ✓ **지식 및 기술 공유:** 팀원 간 역할을 분배하면서 다른 팀원에게 도움이 될 수 있는 지식이나 기술들을 서로 공유할 수 있습니다.

혹은 문제가 발생했을 때 다른 팀원이 이에 대한 해결책을 제시할 수 있습니다.

- ✓ **의견 수렴:** 각자의 부분을 혼자 해결하기보다 팀원들과 피드백을 주고받을 수 있습니다. 이를 통해 팀 내에 서로 지원하는 분위기를 조성할 수 있습니다.
- ✓ **리소스 제공:** 만약 다른 팀원이 직접적으로 해결책을 제시할 수 없는 경우에는 온라인 자료를 통해 같이 의논할 수 있습니다. 이런 리소스를 서로 제공하면서 서로의 발전을 도모할 수 있습니다.