

TC

KOCAELİ ÜNİVERSİTESİ

BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ

VERİTABANI YÖNETİM SİSTEMLERİ

ADI: OĞUZHAN

SOYADI: TALAN

NUMARASI: 151307040

MARKET-STOK TAKİP OTOMASYON

İÇİNDEKİLER:

1. STOK GRUBU İÇERİĞİ
   1. STOK KARTI
   2. STOK LİSTESİ
   3. STOK GRUP
2. CARİ GRUBU İÇERİĞİ
   1. CARİ AÇILIŞ KARTI
   2. CARİ GRUPLARI
   3. CARİ LİSTESİ
3. KASA GRUBU İÇERİĞİ
   1. KASA AÇILIŞ KARTI
   2. KASA LİSTESİ
   3. KASA DEVİR/İŞLEM KARTI
   4. KASA TAHSİLAT/ÖDEME
4. SÜREÇ MODELLERİ
   1. KLASİK ÇEVRİM (ÇAĞLAYAN, ŞELALE) MODELİ
   2. KLASİK ÇEVRİM (ÇAĞLAYAN, ŞELALE) MODELİ SORUNLARI
5. KAYNAKÇA
6. STOK GRUBU İÇERİĞİ:

Yeni ürün girişi için stok kartı formu bulunmaktadır. Bu formdan stok gruplarına da erişilebilmektedir. Stok grupları bakliyat, çay, şeker gibi gruplamaya yarar. Stok listesi ise girilen kayıtları Sql’ den çekip gösterir.

Stok açılış kartında stok kodu, stok adı, fiyat bilgileri, kdv bilgileri bulunur. Stok gruplarında program, kategorilerin girmesini bekler ve girilen kayıtları listeler. Stok listesinde ise daha önce girilen tüm kayıtlar listeli şekilde bulunmaktadır. Arama butonundan kolayca arama yapılabilmektedir.

1. CARİ GRUBU İÇERİĞİ:

Cari açılış kartı, cari grupları ve cari listesi formları bulunmaktadır. Cari açılış kartında ön cari bilgileri ve temel cari bilgileri bölümlerinde her türlü bilgi girişi mevcuttur. Cari grupları ve cari listesi stok grubunda ki mantığa göre çalışmaktadır.

1. KASA GRUBU İÇERİĞİ:

Kasa açılış kartı, kasa listesi, kasa devir/işlem türü ve kasa tahsilat/ödeme formları bulunmaktadır. Kasa açılış kartında banka bilgileri ve banka listesi bölümleri bulunmaktadır. Kasa devir ve kasa tahsilat formlarında işlem bilgileri yapılmaktadır. Borç ödeme, kazanç gibi veriler burada tutulmaktadır.

1. SÜREÇ MODELLERİ:

Yazılım üretim işinin genel yapılma düzenini ve nasıl uygulanacağını tanımlar. Süreçlere ilişkin ayrıntılarla ya da süreçler arası ilişkilerle ilgilenmezler. Uygun yazılım geliştirme modelleri kullanılması, yazılımın daha emniyetli, doğru, anlaşılabilir, test edilebilir ve bakım yapılabilir olarak geliştirilmesinde çok önemli rol oynar. Projeyi geliştirirken klasik çevrim (Çağlayan, Şelale) modeli kullanıldı.

* 1. KLASİK ÇEVRİM (ÇAĞLAYAN, ŞELALE) MODELİ:

İyi tanımlı projeler ve üretimi az zaman gerektiren projeler için uygundur. 70’li yılların ortalarında yapısal programlama ile başlandı. Bölümlendirmeye dayalı ve yönetim kolaylığı sağlayan bir modeldir. Belirsizlik oranı düşükse ve az zaman alacağı öngörülüyorsa (küçük boyutlu kamu sistemleri, personel, bütçe vb.) kullanımı önerilir. Yaşam döngüsü temel adımları baştan sona en az bir kez izleyerek gerçekleştirilir. Barok modelin aksine belgeleme işlevini ayrı bir aşama olarak ele almaz ve üretimin doğal bir parçası olarak görür. Barok modele göre geri dönüşler iyi tanımlanmıştır.

Bir adımın tamamlanmasından sonra diğerine geçilir. Eksiklikler veya hatalar fark edilirse bir önceki aşamaya dönülür. Bu modeldeki aşamalar üst üste gelebilir ve birbirleriyle haberleşebilirler. Örneğin tasarımda gereklilikle ilgili problemler tanımlanır, kodlamada tasarım problemleri ortaya çıkabilir. Yazılım süreci basit lineer bir süreç değildir ve geliştirme faaliyeti sırasında bir dizi tekrar gerekir. En son aşamaya gelindiğinde ortaya çıkan problemleri gidermek için önceki aşamaların tümünü tekrar etmek gerekebilir.

* 1. KLASİK ÇEVRİM (ÇAĞLAYAN, ŞELALE) MODELİ SORUNLARI:

Gerçek yaşamdaki projeler genelde yineleme gerektirir. Genelde yazılımın kullanıcıya ulaşma zamanı uzundur. Gereksinim tanımlamaları çoğu kez net bir şekilde yapılamadığından dolayı, yanlışların düzeltilme ve eksiklerin giderilme maliyetleri yüksektir. Yazılım üretim ekipleri bir an önce program yazma, çalıştırma ve sonucu görme eğiliminde olduklarından, bu model ile yapılan üretimlerde ekip mutsuzlaşmakta ve kod yazma dışında kalan (ve iş yükünün %80’ini içeren) kesime önem vermemektedirler. Üst düzey yönetimlerin ürünü görme süresinin uzun oluşu, projenin bitmeyeceği ve sürekli gider merkezi haline geldiği düşüncesini yaygınlaştırmaktadır.

Gereksinimler iyi anlaşıldığında başarılı bir modeldir. Prototip yapımından kaçınır. Müşteri ile iletişim ilk ve son aşamada yer almaktadır. Değişen müşteri ihtiyaçlarına cevap veremez. İlerleyen aşamalarda bir sorunla karşılaşılırsa, geliştirme süresi uzamaktadır. Hiç model kullanmamaktan iyidir.

Bir sonraki aşamaya geçmeden, önceki aşama neredeyse tümüyle tamamlanmış olmalıdır (ör. gereksinim tanımlama aşaması bitmeden tasarım aşamasına geçilemez.) Bu şekilde geliştirme boyunca değişen müşteri isteklerinin sisteme yansıtılması zorlaşır. Gereksinimleri iyi tanımlı ve değişiklik oranı az olacak sistemler için daha uygundur. Çok az sayıda iş sisteminin gereksinimleri başlangıçta iyi şekilde tanımlanabilir. Bu zorluğu aşmak için; gereksinim tanımlama aşamasından önce iş gereksinimlerinin anlaşılması ve tanımlanması faydalı olabilir. Daha çok, geniş kapsamlı sistem mühendisliği projeleri için tercih edilir.

1. KAYNAKÇA:

<https://www.youtube.com/watch?v=3_xxJvOmtTM&t=7s>

C# ile proje geliştirme KODLAB

T-SQL ile sql server 2012 KODLAB