

Kursnamn: YSYS-OP1  
 Klass: Systemutvecklare.NET 4  
 Termin: HT 2020

# NAMN PÅ UPPGIFTEN: SHOPPINGAPPLIKATION FÖR GOLDSTARR TRADING

## INLEDNING

---

Bakgrundsbeskrivning	<p>Projektuppgiften är tänkt att visa på ett typiskt användningsområde för mjukvaruutveckling där butikförsäljning och lagerhantering är branscher med stort intresse för era framtida tjänster. Ni kommer att märka under er karriär att användningsområdet för applikationerna ni utvecklar vanligtvis inte påverkar kod och designval i någon större utsträckning, så uppgiften blir på så sätt även relevant för utveckling inom andra användningsområden.</p>
Varför ska ni utföra detta arbete?	<p>Projektuppgiftens syfte är att träna studenten i praktiskt arbete med verktyg och kod, att samarbeta i grupp och att planera och leverera en färdig produkt. Projekt inom branschen varierar ofta i omfattning, men är sällan så små att en ensam utvecklare kan utföra arbetet.</p> <p>Resultatet ska vara en körbar applikation och dess källkod som ska bedömas efter de betygskriterier som gäller för projektuppgiften. Den här bedömningen är en del av det slutgiltiga kursbetyget.</p>
Vad ska ni leverera?	<p>Ni ska leverera en körbar applikation som uppfyller kundens krav och behov enligt projektbeskrivningen nedan. Applikationen ska ha ett användargränssnitt som är användarvänligt och tydligt. Eftersom källkoden ska bedömas så ska den uppfylla designprinciper som diskuteras i kursen och vara läsbar för människor.</p>

## ER PROJEKTUPPGIFT

---

Vad ska ni göra? Projektuppgiften går ut på att utveckla ett system för att kunna hantera en butiks affär och lager. Applikationen ska utvecklas som en UWP applikation. Gruppen ska göra två leveranser.

Leverans 1:

Grundläggande lager och affär. Denna leverans ska redovisas för handledare innan gruppen får gå vidare och utveckla leverans två.

Leverans 2:

Utökad funktionalitet. Bygg vidare på applikationen och försök uppfylla fler av kundens krav.

För betyget Godkänd (G) ska gruppen klara av leverans 1, dvs. implementera en fungerande applikation enligt kundens grundkrav. Presentera applikationen och lämna in rapport.

För betyget Väl Godkänd (VG) ska gruppen implementera kundens övriga krav. Presentera applikationen och lämna in rapport.

### Följande beskrivning kommer från GoldStarr Trading

Era kunder är Mr Gold och Mr Starr, som planerar att starta upp en butik för att sälja varor av något slag. De har ett mindre antal produkter som de ska börja med men de hoppas kunna utöka det när försäljningen ökar.

Inför invigningen har de redan en överskommelser med ett antal kunder och leverantörer men hoppas även där att det kommer att växa.

I den bakre delen av butiken har de ett lagerutrymme, de behöver hålla koll på lagersaldot så att de alltid vet vad som finns på hyllorna när produkter säljs eller när det kommer leveranser från leverantörer

### Grundkrav

- En lista över alla nuvarande kunder
- En vy för att kunna lägga till en ny order för en kund
- En lista över alla nuvarande produkter
- En vy som listar alla produkter och hur mycket som finns på lager
- En vy för att kunna hantera en leverans av produkter till lagret.

## Ytterligare krav

- En vy för att lägga till nya kunder
- En vy för att lägga till nya produkter
- All data ska spara på fil och läsas in vid programstart
- En kunds ordrar ska sparas i systemet
- En lista över leverantörer
- En vy för att lägga till nya leverantörer
- Möjlighet för kunden att beställa flera olika produkter
- En vy för att se alla ordrar
- Sortera ordrar på kund
- Sortera ordrar på datum
- Sortera produktvyn på leverantörer eller produkter
- En kö för ordrar som inte kan levereras om lagret är tomt
- En vy för att skicka köade ordrar när lagret har fyllts på
- En mobil applikation för att kontrollera lagret hemifrån
- En lista för ordrar mot leverantörer
- Utöka kundlistan med onlinekunder för den kommande online-affären
- Möjlighet att ta bort kund utan att tappa orderhistorik
- Möjlighet att ta bort leverantör utan att tappa produkter i lager

## Planering

När uppdraget startas upp måste teamet (gruppen) gå igenom uppgiften och bryta ner arbetet i mindre delar för att skaffa sig en uppfattning om arbetsmängden och svårigheten. Denna nedbrytning underlättar också fördelningen av arbetet i teamet.

Det kan vara så att kundens önskemål är orimliga i förhållande till hur mycket tid som finns till förfogande för Release 2. Då måste teamet kontakta kundens representant (**handledaren**) för att diskutera prioritet på önskad funktionalitet och vad som är ett rimligt innehåll till Release 2.

## Data i applikationen (Grundkrav)

Applikationen behöver åtminstone kunna hantera följande data.

Klass för kund **Customer**

Kunddata ska implementeras i klassen **Customer**. Kunddata ska minst innehålla följande information

- Namn
- Adress
- Telefonnummer (valfritt att lämna för kunden)

Klass för produkter **Merchandise**

Varutyperna/produkterna ska implementeras i klassen **Merchandise**. Varutyp ska innehålla minst följande information

- Namn
- Leverantör
- Nuvarande antal varor av denna typ i lagret

En order ska implementeras i klassen **CustomerOrder**. Orderdata ska innehålla minst följande information

- Tid och datum när ordern gjordes
- Beställare (Namn på kund)
- Produkt
- Antal

Ni får lov att lägga till extra data eller klasser om ni anser att det hjälper applikationen.

Exempeldata för Release 1:

```
Customer[] customerList =
{
    new Customer { name = "Lisa Underwood", address = "Smallhill 7", phone = "555-1967" },
    new Customer { name = "Olle Bull", address = "Djäknegatan 13", phone = "555-0344" },
    new Customer { name = "Ben Knota", address = "Stengränd 11", phone = "555-4932" },
    new Customer { name = "Vilma Hypoxia", address = "Nikkaluokta", phone = "555-3356" },
    new Customer { name = "Ken Barbie", address = "Dockgatan 3", phone = "555-3282" }
};

Merchandise[] merchandiseList =
{
    new Merchandise { name = "Hydrospanner", supplier = "Acne AB", stock = 0 },
    new Merchandise { name = "Airscoop", supplier = "Acne AB", stock = 0 },
    new Merchandise { name = "Hyper-transceiver", supplier = "Corelian Inc", stock = 0 },
    new Merchandise { name = "Nanosporoid", supplier = "Corelian Inc", stock = 0 },
    new Merchandise { name = "Boarding-spike", supplier = "Joruba Consortium", stock = 0 },
};
```

## Beteenden i applikationen (Grundkrav)

Man vill kunna lista kunder i applikationen (endast namnet behöver visas i listan)

Man vill kunna skapa en order för en kund, där man väljer en (1) varutyp och anger antalet varor, dock inte fler än vad som finns på lager, tills kunden är nöjd och betalar, då är ordern avklarad och lagerstatus ska uppdateras.

Man vill kunna lista lagerstatus, dvs. alla varutyper och det nuvarande antalet i lager

Man vill kunna fylla på lagret när leveranser kommer från leverantören, vilket ska ske genom att öppna en enkel dialog där man matar in den nya lagerstatusen för respektive produkt.

## Krav och specifikationer för Release 2

Gör först Release 1 och få den godkänd av handledare. Kod och rapport ska också levereras för Release 1 men ni kan bygga vidare på det i Release 2.

Kravlistan ska analyseras med avseende på funktionalitet och tid. För att förstå vad kunden faktiskt menar med kraven så kan gruppen behöva diskutera detta med kundrepresentant (**handledare**).

Kravens prioritet, förtydligande detaljer och den totala omfattningen (vilka krav som ska implementeras) av Release 2 ska dokumenteras i rapporten.

Hur ska ni lösa uppgiften?

Namnge projektet enligt "GoldStarr-YSYS-OP1-Grupp[gruppens nummer]".

Ni ska i grupp först diskutera hur ni kan bryta ner arbetet i mindre delar.

När ni har brutit ner uppgiften i lagom stora delar som är lätta att förstå och implementera, ska ni prioritera dem och planera in dem i den ordning de behöver implementeras.

Ni behöver göra en tidsplan för hur ni ska arbeta. Den görs lämpligast efter att ni brutit ner och prioriterat era deluppgifter.

Det kan är vanligt att använda ett agilt förhållningssätt och utveckla en eller ett fåtal krav/funktioner åt gången och få de att fungera innan ni tar tag i nästa del. Kontakta handledare om ni vill visa en delleverans.

Kom ihåg att det är Gruppens ansvar att få projektet färdigt inte en enskild programmerare. Ledordet är samarbete.

Struktur för arbetet:  
arbetsfördelning  
och tidsplan

Följ de fastställda gruppindelningarna.

Handledare finns tillgänglig men det är alltid bra att boka ett möte för viktigare grupp-diskussioner med handledare. Handledare kan också boka möte med gruppen för att stämma av hur det går för er.

**Vid rapporter ska de studerande använda Newtons rapportmall**

## INLÄMNING OCH REDOVISNING

Inlämning

Inlämning sker via Newtons utbildningsportal senast den 15 november kl 23: 59.

Viktigt att tänka på vid inlämningen.

- Handledaren måste ha tillgång till projektplatsen (ert git repo) för att kunna bedöma koden.
- Ange URL för att komma åt git repot.
- Alternativt kan projektet laddas upp på Newtons Portal i form av en Zip-fil, som då ska namnges med projektets namn, "GoldStarr-YSYS-OP1-Grupp[gruppens nummer]".

Redovisning

Projektarbetet ska presenteras muntligen inför klassen och kursansvarig. Använd gärna Powerpoint för lämpliga delar av presentationen.

I presentationen ska ni redovisa hur ni genomförde er planering och hur ni delade upp arbetet. Berätta även hur ni samarbetat under projektets gång.

För Release 2 ska ni även ange vilken prioritet era krav hade, förtydligande detaljer av kraven om ni fått sådana och samt den totala kravomfattningen, d.v.s. vilka krav som implementerades.

Presentera UML klass diagram över utvecklade klasser. Beskriv kortfattat hur ni resonerade kring er design.

Presentationen avslutas med en demonstration av er applikation där ni visar att ni uppfyllt kravspecifikationen.

## BEDÖMNING OCH ÅTERKOPPLING

---

Bedömning sker  
mot följande  
betygskriterier:

Den här projektuppgiften mäter

Kunskaper i

- Syntax för C#
- Objektorienterad programmeringsteknik

Färdigheter i att

- Planera för och konstruera objektorienterade program
- Planera för och konstruera grafiska användargränssnitt
- Kommunicera med objektorienterade termer

Kompetenser i att

- Bedöma tekniska lösningar
- Formulera och lösa problem

För att få godkänt (G) på projektarbetet ska följande krav uppfyllas

- Applikationen (Release 1) ska kunna köras i Visual Studio utan problem
- Applikationens funktionalitet ska kunna användas från det grafiska användargränssnittet
- Applikationen ska uppfylla alla grundläggande krav som är specificerade enligt projektbeskrivningen ovan
- Koden ska följa objektorienterade principer
- UML klassdiagram som beskriver klassers beroende till varandra ska lämnas in tillsammans med koden

För att få väl godkänt (VG) på projektarbetet ska följande krav uppfyllas

- Applikationen (Release 2) ska kunna köra i Visual Studio utan problem
- Applikationens funktionalitet ska kunna användas från det grafiska användargränssnittet
- Applikationen ska uppfylla alla de krav som är överenskomna och specificerade med kundrepresentanten för er grupp
- Koden ska följa objektorienterade principer
- UML klassdiagram som beskriver klassers beroende till varandra ska lämnas in tillsammans med koden och ska innehålla klasserna för både den grundläggande versionen och den utökade versionen

Återkoppling

Du kommer att få återkoppling genom Newtons bedömningsmall för projekt- och inlämningsuppgifter.

Återkopplingen sker skriftligen och finns tillgänglig på Newtons portal den 30 november.