

TNMK30 - Projekt

Överblick och syfte

Projektuppgiften går ut på att designa och implementera ett webbaserat gränssnitt för att söka på information i en befintlig databas. Webbsidan ska vara komplett med navigering, överblick, sökning och strukturerad presentation av stora mängder data. Databasen som vi föreslår att ni använder är given och förslag på möjliga uppgifter följer. Även egna idéer kan diskuteras om särskilt intresse finns.

Syftet med uppgiften är att ni ska kombinera kursens alla delmoment för att lösa problemet på ett väl genomtänkt och motiverat sätt. Uppgiften är vald för att vara realistisk, konkret, väl avgränsad och klart definierad, för att ni skall hinna med att utföra den på den begränsade tid som står till förfogande. Syftet är också att ge er en allmän övning i problemlösning och projektarbete.

Konkret problemområde

En databas som vi har fått vänligt tillstånd att kopiera och använda för icke kommersiellt bruk är den omfattande Lego-databasen som nyligen byggts upp på BrickLink (www.bricklink.com). Databasen innehåller en katalog över Legobitar och Legosatser för att underlätta för Lego-hobbyister och samlare världen över. Förutom SQL-tabelldata finns även en bilddatabas med bilder på samtliga satser och samtliga bitar. Liknande databaser finns även på Peeron (www.peeron.com) och en mindre avancerad på LUGnet (www.lugnet.com).

Databasen används av Lego-byggare i alla åldrar för att hålla koll på samlingen. Det finns två sorters typiska användare på BrickLink. De flesta är kreativa personer i åldrarna 15-50 år som har barnasinnet i behåll och som har Lego-byggande som en "seriös" hobby, alltså något som man ägnar ganska mycket tid åt då och då för att koppla av och roa sig själv och andra. En andra sortens användare är personer som har som jobb att sälja begagnat Lego på nätet. Ibland har de det som heltidssyssla, men oftast som ett extrajobb vid sidan av. Databasen används för en rad olika ändamål, bland annat för att hålla koll på en stor Lego-samling, för att kunna sortera ut en viss sats ur en blandad hög med bitar, och för att bena ut vilka satser som egentligen ingår i en blandad hög med Lego man köpt på eBay utan instruktioner.

Några typiska praktiska och konkreta funktioner finns inbyggda i det gränssnitt som redan finns på BrickLink, Peeron och LUGnet. Funktionerna är användbara och ganska bra, men gränssnitten är gjorda på frivillig basis av personer som mestadels är amatörer. Lösningen har därför flera brister, och kan alltså tjäna som inspiration och underlag för kritik och vidareutveckling snarare än ett ideal att sträva efter. Ni har kompetens att skapa något lika bra eller bättre, eller i vart fall annorlunda, själva. På den korta tid som står till buds för projektet kan ni inte göra så mycket, så uppgifterna vi föreslår är klart avgränsade delar av ett tänkt större gränssnitt, till exempel något liknande det som redan finns på BrickLink.

Uppgifterna nedan är något olika för omväxlingens skull och har något olika fokus, men de har ungefär samma omfattning och svårighetsgrad. I flera fall finns det en existerande eller mycket liknande lösning på BrickLink, Peeron eller LUGnet. Om ni väljer att kopiera en sådan befintlig lösning till funktion och/eller form så är det fullt tillåtet, men en egen idé eller en väl genomtänkt variation på en befintlig idé bedöms nästan helt säkert med en högre betygsgradering. Även om den egna idén kanske inte slår fullt så väl ut som ni hoppats så vill vi belöna att ni tänker kreativt i något andra banor än en redan existerande lösning.

Om ni har helt andra förslag till uppgifter än dem i listan nedan ber vi er att beskriva dem på samma detaljnivå och presentera dem för godkännande innan ni börjar. Vi vill försäkra oss om att era projekt inte blir för smala, för svåra eller för enkla.

Förslag till uppgifter

1. Skapa ett gränssnitt som gör det möjligt att söka bland alla Lego-satser, hitta rätt sats och se dess detaljerade innehåll.
 - Sök sats på nummer eller (del av) namn, red ut tvetydigheter.
 - Visa en lista av alla ingående bitar i en viss sats, med bilder.Tänkt användare: samlare och säljare som vill sortera ut satser ur en blandad hög med bitar.
2. Skapa ett gränssnitt som gör det möjligt att söka bland alla Legobitar, hitta rätt bit och se vilka satser den ingår i.
 - Sök bit på nummer eller (del av) namn, red ut tvetydigheter.
 - Visa en lista på alla satser som har biten, med information om färg samt med bilder.Tänkt användare: samlare och säljare som vill veta var en viss bit kommer ifrån, eller var man kan hitta den i en ny eller begagnad sats.
3. Skapa ett gränssnitt där man kan ange en lista av olika bitar och se vilka satser som innehåller någon eller några av dessa bitar, sorterat efter hur många bitar som ingår.
 - Ange en lista av bitar i ett formulär, eller ange en sekvens av sökbegrepp som ackumuleras och successivt smalnar av sökningen tills man hittat rätt.Tänkt användare: samlare och säljare som köpt en blandad hög med bitar utan instruktioner, och vill sortera ut så många kompletta satser som möjligt ur högen. Detta gör man genom att först identifiera ovanliga bitar och se vilka satser de kan tänkas härröra från.
4. Skapa ett gränssnitt för att beräkna olika slags statistik över databasen och presentera det på ett överskådligt sätt (t.ex. som grafer i ett <canvas> element).
 - Räkna antalet satser i katalogen, och totalt antal olika typer av bitar i katalogen
 - Visa ett histogram över antalet bitar i varje sats
 - Visa ett tidsdiagram över antalet nya satser per år
 - Summera antalet satser och antalet bitar i en viss persons samling
 - Skapa en topplista över vilka bitar som det finns flest av i satser, vilka bitar som ingått i flest eller minst antal satser, vilka satser som har flest bitar totalt eller flest bitar av en viss sort eller färg, vilka bitar som existerat längst och kortast tid, och annat som kan vara intressant att veta.Tänkt användare: folk som av ren nyfikenhet är intresserade av vad Lego gjort genom åren, men också säljare som vill hålla koll på vilka bitar som är lätta och svåra att få tag på, och därigenom kunna sätta kloka och rimliga priser på begagnat Lego.
5. Det finns även en databas över minst en persons personliga samling av Lego-satser, med satsnummer och antal klart angivet. Skapa ett gränssnitt där det presenteras vilka bitar som finns, vilka bitar som finns men i fel färg, samt vilka bitar som saknas i samlingen för att kunna bygga godtycklig annan Lego-sats.
 - Välj satsnummer och ta reda på vilka bitar som ingår
 - Sök matchande bitar i samlingen i rätt färg
 - Sök för de bitar som saknas en match i godtycklig färg
 - Presentera resultatet på ett överskådligt sätt

Tänkt användare: samlare som vill försöka bygga ihop officiella satser som de inte äger, eller se vad de får för tillskott i sin samling genom att köpa en viss sats.

6. Utifrån en lista (i en separat databas) över en persons samling angivet som satsnummer och antal, presentera en total översikt över samlingen organiserad efter typer av bitar, med möjligheter att titta närmare på exakt varifrån bitarna av en viss typ kommer (satsnummer och årtal).
 - Summera och presentera totala antalet av varje sorts bit i samlingen
 - Presentera en lista över källorna till varje sorts bit

Tänkt användare: samlare som har så mycket Lego att de inte längre har någon koll på var alla bitar kommer ifrån, men som vill skaffa sig den överblicken.

Handledning

Viss begränsad handledning kommer att ges för att säkerställa och följa upp att projektgrupperna gör rätt saker och disponerar sin tid och sin arbetsinsats rimligt väl, men själva arbetet är ett inlärningsprojekt som skall utföras i stort sett utan handledning i grupper om fyra studenter.

Frågor skall i första hand besvaras genom egna studier i kursmaterialet och diverse dokumentation, samt genom egna praktiska försök. För databasfrågorna finns handledare som är kunniga i SQL till hands för att hjälpa till med konkret SQL-frågeformulering, eftersom ni ännu inte läst någon kurs i databaser, och för tekniska problem av mer krånglig karaktär finns examinator och kursassistenter tillgängliga för att svara på frågor.

Projektarbetet sker primärt i form av självstudier. Mellan vecka 49-51 finns det 2-3 labsalar bokade för projektarbetet, 4 timmar per gång. Information om tider och labsalnummer för dessa hittar ni under Schema. Under dessa schemalagda tillfällen kommer det finnas 2-3 labhandledare tillgängliga i 2 timmar per tillfälle för att svara på frågor.

Länkar

Länkar till material och nyttig information finns på kurshemsidan.

Redovisning av projekt

- Redovisning av projektuppgiften sker via e-post till Niklas Rönnberg, niklas.ronnberg@liu.se
- Rubrik på e-postmeddelandet ska vara: 'TNMK30 redovisning grupp X ' Er gruppnummer kan ni ta reda på under Projektgrupper.
- E-postmeddelandet ska innehålla följande 3 länkar:
 1. Slutresultatet av er projektuppgift som en länk till den fungerande lösningen.
 2. En länk till en komprimerad mapp (zip-fil) innehållande all er källkod (alla filer för projektet) som kan laddas ner för närmare granskning. Källkoden skall vara strukturerad och kommenterad så att en insatt läsare lätt förstår dess funktion.
 3. En länk till en kort projektrapport i pdf format (se nedan).

Deadline för redovisning

Deadline för redovisning av projekten är fredagen den 13 januari 2017, kl 24.00. Projekten kommer att granskas snarast därefter, resultatet skall som för tentamen vara klart senast två veckor efter inlämning. Förutom betyget kommer det även att ges en kort motivering till projektresultatet.

Om resultatet inte skulle bli godkänt kommer det att ges chans till en snabb komplettering inom två veckor efter ni har fått besked. Om ni är missnöjda med betyget för hela gruppen gäller samma sak: komplettering skall ske inom två veckor. Om någon är missnöjd för egen del, kontakta examinator snarast efter examinationen för närmare besked. Personlig komplettering är möjlig, men bara om du anmäler att du önskar en sådan senast en vecka efter examinationen.

Senare inlämning än 13 januari medför att examination inte sker förrän i omtentaperioden, och betyget registreras i så fall på vårterminen i stället för höstterminen. Om examination önskas i samband med omtentaperioden måste projektet redovisas senast 21 april 2017.

Projektrapport

Redovisningen av projektuppgiften ska omfatta en projektrapport. Rapporten ska publiceras som PDF och ska inkludera följande rubriker:

- Projektgrupp och projektuppgift
- Projektmetod och arbetsfördelning
- Skiss, mockup, lofi av layouten
- Självvärdering
- Tidsåtgång

Projektrapporten ska vara 2-3 A4 sidor i 12 punkters font; dvs. 4500-6000 tecken utan mellanslag.

I rapporten ska ni ska redogöra för er arbetsmetod och arbetsfördelning. Om ni har upplevt en skev arbetsfördelning i gruppen som borde leda till olika betyg för olika gruppmedlemmar så bör ni lyfta detta i er rapport. Eftersom betygen bestäms utifrån slutresultatet av era projekt och projektrapporten så kommer alla medlemmar i gruppen få samma betyg om inte annat anges.

Examinationskriterier för projekt

Alla redovisade delar examineras:

1. Slutresultatet av er projektuppgiften: den fungerande lösningen
2. Källkoden
3. Projektrapporten

För godkänd i projektet krävs godkänd i alla tre delar.

Kriterier projektuppgift

Uppgiften är betygsgraderad, och för att den skall kunna genomföras med gott resultat krävs det att ni använder samtliga verktyg och metoder som gåtts igenom i kursen. Projektet berör följande problemområden som alla är centrala för kursen, och som kommer att bedömas till ett sammanvägt projektbetyg:

1. Korrekt användning av aktuella standarder för webpublicering.
HTML5, CSS3, JavaScript, PHP och MySQL ska vara korrekt använda. CSS-mallar och JavaScript-kod bör ligga i separata filer. Tänk på att dela upp er kod med hänsyn till presentation, innehåll och funktionalitet.
2. Interaktionsdesign för enkel och begriplig navigation.
Användaren ska kunna navigera på ett enkelt sätt och inte "bjudas på några överraskningar" vid navigerandet. Webbsidans design bör vara konsistent och navigeringen tydlig. Utgå från skiss, mockup, lofi. Bifoga denna i rapporten.

3. Användbar funktionalitet i ett sammanhållet gränssnitt.
Funktionaliteten måste omfatta det som er målgrupp av användare behöver utföra.
Att det finns en sökfunktion är ett minimikrav och den ska vara utformad så den täcker användarens behov och uppgiftens mål.
4. Överskådlig och grafiskt tilltalande presentation.
Sökresultatet bör presenteras på ett välstrukturerat sätt. Skapa gärna en moodboard för att få en enhetlig och genomtänkt känsla i layouten.
Korrekt användning av bilder, såväl tekniskt som presentationsmässigt.
De bör laddas i rimlig tid och ha rätt avpassad upplösning.
5. Korrekt användning av enklare databasfrågor.
6. Serverscriptprogrammering för logiken i sökningarna.
Tänk på vad du använder SQL-frågor till och vad du kan strukturera med PHP.
7. En viss mängd klientscriptprogrammering för tydligheten i gränssnittet.
8. Felhantering med vänliga och konstruktiva felmeddelanden.
Er webbplats bör klara av att hantera fel i indata från användaren och fel som kan uppstå i serverskriptkoden, t.ex. att en SQL-fråga inte returnerar något resultat.
9. Gör webbsidan responsiv så att den anpassas efter upplösning på skärmen, och fungerar på handhållna enheter såväl som på datorer.
10. Gränssnitt och hjälpsystem som passar en målgrupp med blandade förkunskaper.

OBS! För godkänd på projektuppgiften krävs godkänd på problemområden 1-6, problemområden 7-10 används för att plussa betyget.

Kriterier källkod

Källkoden kommer att granskas utifrån följande kriterier:

1. Struktur & tydlighet
2. Uppdelning av ansvarsområden
3. Meningsfulla val av variabel & funktionsnamn
4. Kommentarer & dokumentation

Dessutom:

- Inga färdiga CSS frameworks får användas i projekten
- Om ni har lånat kod ska ni referera tydligt till källan

Kriterier projektrapport

Rapporten ska följa anvisningarna som står specificerade under Redovisning. Den ska inkludera följande rubriker:

- Projektgrupp och projektuppgift
- Projektmetod och arbetsfördelning
- Layout
- Självvärdering
- Tidsåtgång

Projektrapporten ska vara 2-3 A4 sidor i 12 punkters font; dvs. 4500-6000 tecken utan mellanslag.