

Guldkort - rapport

(efter projektet är klart)

Arbetsgång

Överlag gick projektet bra. Slutprodukten följer, i stora drag, aktivitetsdiagrammet samt pseudokoden. Jag började med klasshierarkin, sedan skapade en mekanism som importerar all data från både databaser. Därefter ett sätt att få kontakt med klientprogram över TCP och sedan hur serverprogrammet skulle svara. Själva gränssnittet kom sist.

Det som blev svårt var att hålla koll på många olika variabler, objekt och datatyper. Jag insåg även att jag saknade mycket kunskaper vad gäller databaser, så databasanvändning i slutprodukt blev väldigt enkel. Jag har detaljerat exempel på en del problem som uppkom i nästa delen.

Tanken var även att grafiken skulle vara en variabel/egenskap i varje underklass av typ kort. Tiden räckte inte till, tyvärr, och efter att vara nära att gå vilse i olika resursmappar samt referenser i projektarkitekturen, la jag ned det.

Svårigheter, problem och lösningar

Att iterera igenom min egen generisk klass

Jag hade skapat en egen generisk klass för att kunna använda för lagning av kort- samt användare referenser. Det jag inte förstod var att "foreach"-iteration i .Net Framework behöver en "GetEnumerator"-metod för att kunna itererar igenom alla element i listan. Jag hade inte tid eller kunskap nog att kunna lägga till en sådan metod i min egen generisk klass. Jag till och med tittade på källkoden för List<T> här (<https://referencesource.microsoft.com/#mscorlib>) men kunde inte implementera själv.

Den generiska "Samling<T>" klass används då endast med enkla strängvektorer, som egentligen används inte annars av programmet. Referenser till kort- samt användarobjekt som skapas utifrån datan importerade från databaserna lagras i en vanlig "List"-objekt istället.

Logg

Programmet löser situation där ett kort kan lösas in två gånger genom att, efter ett vinnande kort har lösts in, ta bort det från listobjektet (med möjlighet för att ta bort från databasen). I

praktiken vill företaget nog ha en historiskt logg på vilka kort har vunnits av vilka användare, till exempel för framtida marknadsföring eller bokföring o.s.v.

Förslagsvis kan en attribut (kolumn) läggas till i databasen för kort som visar huruvida ett kort har lösts in eller ej, samt en attribut som länkar användarNr (postnyckeln) till relevant kund i kunddatabasen. Vid lyckad inlösning av ett kort kan programmet skriva information direkt till kortdatabasen, istället för att ta bort posten helt.

Loggen som finns i programmet var mer för att hjälpa med min utveckling av programmet, men jag lämnade den kvar iallafall. Den är ett väldigt enkelt sätt att se att serverprogrammet har tagit emot ett meddelande och när det hanterats och kan även exporterats till .txt-fil.

Databas

Databashantering i programmet är väldigt enkel. Programmet i princip bara läser in från en befintlig databas i början och har möjlighet (fast kanske inte på bästa sättet) att ta bort en post. Eftersom företaget vill kanske ha servern på gång hela tiden behöver det finnas ett sätt att ändra databasen direkt i programmet, eller ett sätt att regelbundet uppdatera inläsningen.

Detta hänger även ihop med "Logg" ovanför. Se förslag för vidare utveckling!

Grafik

Programmet kan inte i nuläget hanterar överföring av grafik över TCP-uppkopplingen. Som ett exempel på hur ett meddelande kunde se ut för användaren, öppnas en annan Windows Form inifrån serverprogrammet.

För att kunna lägga till grafik i kundmeddelandet skulle jag behöva utveckla programmet till att konvertera de olika .png-filer till byte[] som sedan kan skickas över till klientprogrammet. Jag skulle även behöva utveckla klient-programmet för att sedan hantera sådana byte[] för att kunna visa för kunden.

Förslag för vidare utveckling av programmet:

1. Möjlighet för företaget att lägga till guldkort samt användare direkt in i programmet, som uppdateras i databasen. Både EDM samt LINQ krävs, bl.a.
2. Möjlighet för uppdatering av databasen manuellt alternativt vid jämna intervaller
3. Möjlighet för tilläggning av nya *typ* av guldkort

4. Utveckling av klientprogrammet så att det kan ta emot grafiker samt felhantering direkt till användaren vad gäller användaruppgifter bl.a.
5. Undantagshantering skulle ytterligare kunna förbättras. Till exempel, felinmatade poster och/eller fält i databasen kommer inte att igenkännas av metoderna. Vid felstavning av ett guldkortnamn i databasen kommer inte ett objekt skapas upp och även vid rätt inmatning av kortnummer av användaren kommer ingen pris tilldelas.