Beerbrewing

En dokumentation av eget projekt

Idéen för det här projektet växte fram ur mitt intresse för att brygga öl. Jag ville skriva ett program där användaren kan registrera samtliga ingredienser som användes vid bryggning samt spara ner viktig information kring processen.

Bakgrund

Ölbryggning består av flera olika steg, där noggrannhet ofta är viktigt. Det finns en hel del att hålla koll på, både när det gäller förberedelsen av ingredienser och vidare själva bryggningsprocessen.

De ingredienser som klassiskt ingår i öl idag är: malt, humle och jäst. Här möter vi stora variationer i både typ, antal och mängd använda sorter. Vanligast används en till två maltsorter. Antalet humlesorter som brukar användas är svårare att säga, men oftast vill man ha minst två olika typer av humle. Sist tillsätts jäst. Här erbjuder marknaden många olika sorter, valet beror ofta på vilken typ av öl man vill brygga. Jäst finns att tillgå i två former: torr och flytande. Åsikterna kring vilken form som är att fördra varierar, och är inget jag tänker ge mig in i här.

Självklart finns det även andra tillsatser som kan ingå i ett ölrecept, och även här är variationen stor. Exempelvis kan man tillsätta frukt i olika former (koncentrat, puré eller juice) för att få in en fruktsmak i ölen.

Nu till själva processen, en kort och odetaljerad beskrivning. Bryggningen börjar med att värma vattnet till en temperatur mellan 65 och 78 grader. Malt tillsätts sedan i vattnet, och vi låter denna dra i en timme. Efter det ska malten silas bort, eller lakas som det heter inom ölbryggning. Nästa steg är att koka upp denna vätska, vi mäter tid 60 minuter från kokpunkten. Under koktiden tillsätts humle. Denna ger olika effekt beroende på när den läggs till. Tidig humle (60 minuter kvar av koktiden) bidrar främst till beska, mellan humle (30 minuter kvar) ger smak och den som tillsätts sist (5 minuter kvar) bidrar med doft.

När en timme har gått ska vätskan kylas. Först när den når rumstemperatur kan vi fortsätta att jobba med den. Vi mäter först ölets OG (original gravity) med hjälp av en hydrometer. Detta värde används senare för att räkna ut alkoholhalten i den färdiga ölen. Vidare tillsätts jäst, och kärlet sluts tätt med ett lock. Nu är det dags för jäsning. Vanligtvis tar denna process två till fyra veckor. Jäskärlet ska stå inte stå i direkt solljus och gärna ha en stabil rumstemperatur.

När jäsningen är klar mäter vi ölets FG (final gravity). Med hjälp av en formel kan vi nu baserat på ölets OG och FG räkna ut alkoholhalten. Vi tillsätter också sockerlag till ölet, som hjälper till att bilda kolsyra i flaskorna. Ölet tappas upp på flaska och behöver cirka två veckor på sig för att bilda kolsyran i drycken.

Avgränsning

Jag har gjort en första avgränsning för det här projektet, som kan ändras beroende på mängden tillgänglig tid. Nedan text är därför uppdelad enligt följande: vad ska ingå i programmet, vad ska uteslutas samt utvecklingsidéer om det finns tidsutrymme.

Till att börja med vill jag att programmet ska bestå av tre menyval:

1. Grundrecept – ett generellt recept för ölbryggning.
2. Lägg till ny brygd – i detta menyval kommer programmet att ställa en rad frågor till användaren, och sedan spara datan i ett nytt objekt ”Brew”.
3. Visa sparade bryggningar – Presenterar hittills sparade bryggningar på ett lättläst och överskådligt sätt.

I menyval 2 kommer användaren få svara på frågor kring en brygd utifrån ett klassiskt recept. Frågorna kommer att handla om: datum för brygden, ingredienser (malt, humle, jäst) samt deras mängd, jästid, OG och FG, antal liter som brygden gav, beskrivning, betyg och kommentar, där användaren har möjlighet att lägga till förslag eller tankar till framtida modifieringar. I det tredje menyvalet där bryggningarna presenteras ska programmet med hjälp av en formel räkna ut alkoholhalten och även skriva ut denna tillsammans med övriga data.

Några av de saker jag har valt bort i detta projekt övriga tillsatser i ölen, exempelvis frukt. Tidigt i projektet funderade jag på att strukturera upp programmet utifrån recept. Med det menar jag att man vid bryggning lägger till ett recept, och anger vilket datum bryggningen gjordes. Om man vid ett senare tillfälle bryggt enligt samma recept skulle man i så fall kunna ”återanvända” det tidigare receptet och bara lägga till ett nytt datum. Jag valde dock att istället formulera ett register utifrån vilket datum som brygden gjordes, där varje recept är unikt. Detta val grundade sig i att man, enligt min erfarenhet, ofta gör små ändringar i brygden även om man utgår från ett tidigare recept. Jag tror därför att det blir mer lättöverskådligt att se varje brygd som unik.

Om tiden räcker till har jag ytterligare några idéer på vidareutveckling av programmet. En tanke är att det ska vara möjligt att gå in och redigera vissa data i en sparad brygd, exempelvis kommentar, beskrivning och betyg. Jag utgår här ifrån att användaren kanske vill lägg till brygden direkt efter att denna är klar, men innan man hunnit provsmaka ölen. Då kan det vara lämpligt att ha dessa frågor som valfria med möjlighet till senare förändring. En annan idé är att ha fler menyval som hjälper till med uträkningar, exempelvis hur mycket sockerlag som ska tillsättas innan ölen tappas upp på flaska. En sista idé är att jobba med layouten av grundreceptet för att detta ska bli lättläst, exempelvis med hjälp av biblioteket TKinter.

Prioriteringslista

1. Bygga upp menyn med de tre menyvalen.
2. Lägga till ett grundrecept på öl.
3. Skapa en struktur för de frågor som användaren ska svara på i menyval 2.
4. Skriva en klass för Ingredient.
5. Skriva en klass för Brew.
6. Skriva koden för att skapa objektet ifrån klassen Brew baserat på den data som användaren uppger.
7. Skriva kod för menyval 3.
8. Arbeta med layouten av data i menyval 3.
9. Införa en funktion som låter användaren returnera till huvudmenyn från menyvalen.
10. Arbete med layouten för grundreceptet.
11. Möjlighet att ändra vissa parametrar i en sparad bryggd, exempelvis kommentar.
12. Ytterligare uträkningar, exempelvis mängd socker vid sockerprimning.