Title: Brew Day!

Home brewing, the process of producing beer on a small scale for personal purposes, is an activity that receives growing attention among beer enthusiasts. Every home brewer owns a brewing equipment (kettles, fermenters, pipes, etc.) with a certain maximum brewing capacity, the number of litres the equipment is able to handle in a single "batch". Brewing also requires ingredients, whose actual amounts vary from recipe to recipe; these are various kinds of malts, hops, yeasts and sugars (and of course, water). Brewers like to log their recipes, for future reference, and maintain an updated list of available ingredients, for shopping before the next brew.

The goal of this project is to develop an application for home brewers thats allows them to maintain a list of recipes, and adapt existing ones. The application must also maintain a list of available ingredients, update this list after a batch and when new ingredients are bought, and produce shopping lists for the next batch. A special characteristic of the application is the "what should I brew today?" feature: it goes through the recipes, and taking into account the available ingredients and brewing capacity of the equipment selects a recipe that can be brewed with the available ingredients, maximizing the use of the ingredients, and the batch size.

Project Description

Brew Day! is an application that allows home brewers to maintain an organized database of their beer recipes. The application allows users to create, store and modify recipes, and later on delete them, if the user wishes to do so. The application is intended for "all-grain" brewers only, and thus all recipes are for this kind of brews (the "extract" brews are not supported).

Every home brewer has a specific equipment, whose characteristics leads to a particular "batch size", the maximum number of litres that can be brewed on a single run. Recipes involve, besides water:

- malts
- hops
- yeasts
- sugars
- additives

While brewers prefer to create recipes referring to concrete values, like kilograms of a specific malt or grams of a specific hops, the application must store these recipes in some "absolute" measure, that allows for a direct conversion of the recipe when the equipment, and consequently the batch size, is updated. For instance, expressing malt quantities as a percentage of the total, and hops as grams per litre of mash, is a possibility.

Besides the actual recipes, the application must maintain recipe instances, i.e., particular brews based on a recipe; these instances can be accompanied by notes to refer to issues that may affect the resulting beer and the brewers would like to keep logged. A particular kind of note is the tasting notes, that allows brewers to keep track of opinions on a beer from a particular brew.

Besides these more traditional features of Brew Day!, the application maintains a list of available ingredients. This allows brewers to be notified about missing ingredients for the next brew. A recipe instance, i.e., a particular brew, should allow users to update the available ingredients list, substracting used ingredients from the available ones. Related to this information, Brew Day! must support a useful feature for brewers: "what should I brew today?" goes through the recipes database, and chooses the recipe that maximizes the use of the available ingredients, taking into account the equipment capacity, of course.

Project Scope

The project must implement the features described above, i.e., creation, modification and deletion of recipes, creation of recipe instances (brews), support for notes on brews, and keeping track of available ingredients. The "what should I brew today?" is a mandatory feature. Optionally, developers may choose to allow ingredients availability manually, as opposed to do it automatically from brews information.

Process Requirements

In order to track the project's activity and progress, the following rules should be followed:

- For each major set of development tasks, a milestone characterising them must be defined.
- For each development task, an issue describing it must be defined, within the corresponding milestone.
- Task granularity is up to the teams. For larger teams a small granularity is better (work units may be method implementations, method documentation/contract provision, unit tests for methods, etc.).
- Prior to creating a new issue, users must check if the same issue has been previously created (important for large teams) to avoid work redundancy.
- The use of branches for working on partial solutions or larger issues is recommended.
- All users must make sure their partial solutions do not break the system (tests that passed before must still pass, project must compile, etc.).