GSoC Proposal 2011
Cutecash

I have been interested in programming for five years. I have been learning C++ language by myself through reading books and writing small projects on my own, like simple games, etc. In high school programming was limited to basics of Visual Basic. During this time I have been developing my math skills and working on basics algorithms and data structures. During the studies I have taken classes in Pascal and C language.
I have been using Dev C++(Mingw compiler) and Borland C++ Builder on Windows, now I am working on Fedora and Arch Linux, where I use: NetBeans IDE(rarely)(g++) and Geany + g++/gcc + simple makefile.
I have also got basic experience in Microsoft Visual Studio. If it is necessary, I am able to work on both operating systems and different IDEs.
I have compiled Gnucash and Cutecash from source, certainly.

Project

I use Gnucash almost every day and it was obvious to me, that it is a great opportunity to develop software, which is useful for almost everyone - not only for the ordinary man, but also for people, who are running small businesses.
I chose to work on Cutecash project and when I was doing the research to find some new, good ideas, which could improve this program, I have found one interesting demand on Uservoice.
Many users complain about the lack of possibility to classify and manage accounts and transactions. Bugzilla has given me an idea of a system, which could solve these problems and could make the working with Cutecash much easier for many people.
My idea is to implement system with two important classes: Tag and Filter.
That simple solution gives a user the option to sort and manage the transactions without creating another subcategories, labels, etc. He just needs to add a tag (for example "Summer Project") for one or more transaction. Of course, some of these incomes/expenses can have another tags, so the user can assign one transaction to many his own ideas, projects or expenses and he does not need to give these information to Cutecash, to create groups, labels, etc.
Filers are the second part - their responsibility is to allow the user to sort the transactions exactly as he wants to. I have prepared two different methods, both to implement:

- ❒ a) a simple filter, which will allow the user (by using GUI) to define at least one and not more than three conditions, each with one type („Can't be", „Can be", „Must be"); the user can define in an easy way, that he wants to see only transactions that:
  - have every tag given to condition „Must be"
  - have at least one of the tags given to condition „Can be"
  - do not have any tags given to condition „Can't be"
- ❒ b) filter defined by an expression - for users, who need more complex conditions; my decision is to implement four types of command: OR, AND, XOR, WITHOUT, but the structure of parser will be easily expandable; for example, expression:
  ((„Project" AND „Catering") OR ((„Delegation" AND „Accommodation Costs") WITHOUT „Paris"))
  will create a filter, which shows only transactions, that have tags „Project" and „Catering" or transactions with tags „Delegation" and „Accommodation Costs" and without tag „Paris".

Benefits

Each user will gain easy, simple and (I hope) fast working tool to add unlimited number of his own tags for each transaction, to remove and modify them. This will give to the users a possibility to manage the transactions in their account, assign them to many groups and create filters. I think, that it can make many things easier and simpler and that it will save the time of the users.
Uservoice shows that a feature like that is highly desirable.

Technical problems

My goal was to make this tool as independent as possible, maybe even to allow porting code to Gnucash. I have been searching for solutions for problems in the documentation of Gnucash, but I have delayed some decisions to time of preparing detailed project's plan.
I think, that the best way is to keep this far away from GUI code and just allow it to create its objects, make changes and get data using its functions and showing the results will depend on GUI programmers.

Tag system is based on two trees (fast searching). One is sorted by "tag" key and every node has its list of pointers, ID's or another data, which gives the access to all transactions with that tag. Second tree is very similar. It is sorted by transactions and every node contains a list of tags. This tree will be used to get fast all of the tags for that transaction(and modify them).

Simple filter system can be an easy implement using Composite pattern.
Filter system based on expressions requires parsing function; I have prepared a simple algorithm with stack. The result is a set of pairs - operation and two arguments: tags or pointers to result of another pair. Computation requires only simple recursive function.
The Filter class should return the list with transactions, which can be shown in that filter.

The hardest part of this project is saving tags and filters. I want to change as little as possible in the code, but modifying the saving functions seems to be necessary.
One option is to save filters in another file – a user will have the possibility to create sets of filters and load only these, which will be useful in that particular moment.

Timeline

I had to move some tasks from June to May and July due to my four exams. Fortunately, I do not have many classes and I will be finishing all my laboratories in May, so I will be able to start coding before 23rd May; I am going to make the Summer of Code my absolute priority and I will make up the time I have lost during the exams period by spending more than 40 hours per week in July on developing(I will include weekends). I do not have any jobs or other important things during the summer.
During the project I am going to make frequent reports(the frequency depends on mentor's will) of these information: what should be done, what is done, what is not done and why, problems and my ways to solve them, code. Apart from that I am going to contact with my mentor with each difficult situation, where I do not know how to fix it or I am not sure that my solution is good.

Before 23rd May:

- ❐ staying in contact with mentor, asking questions, solving problems in projecting
- ❐ staying in contact with community to discuss with them problems, modifications, unclear requirements

- ❒ reading Gnucash documentation
- ❒ all this will be used to write detailed project’s plan; after this I should have all milestones clear and firm and I should be able to start coding
- ❒ implementing interfaces for both classes

May 23 - May 30
- ❒ developing Tag class
- ❒ testing, debugging, writing documentation

May 30 - June 6
- ❒ developing Filter class
- ❒ testing, debugging, writing documentation

June 6 - June 27
- ❒ providing simple GUI in Qt, just to show that everything is working
- ❒ testing existing code

June 27 - July 11
- ❒ integration with Cutecash
- ❒ testing, writing documentation
- ❒ polishing existing code, correcting comments

July 11 - August 2
- ❒ adding support for saving tags and filters(XML, sql)
- ❒ testing existing code, writing documentation
- ❒ polishing, implementing last small changes and fixes

August 2 - August 15
- ❒ buffer for unforeseen problems and delays
- ❒ preparing test files, testing for many combinations of tags and filters
- ❒ writing and correcting documentation