

GP1: Bank Accounts

We want to develop a software for a bank. This software creates a profile for each client in that bank. Each profile contains the following info:

- Name
- Address
- Balance
- Last 30 transactions

The software must

- keep the profile of all clients and add a profile for each new client
- update the profile of each client
 - Withdraw money
 - Deposit money
 - Transfer money (from one client to another)
- provide a summary of info for any client upon request:
 - All account info
 - Last transactions

The output must be also deliverable in a .txt output file.

Bonus: Creating a secure login username password for each customer is bonus.

GP2: What Car Should I buy?

We want to develop a software which helps finding the best car you can buy. The software has an extensive list of cars from a large range of automobile companies with the corresponding values and interest rates. For example:

- Company (e.g., Audi)
- Type (e.g., A6)
- The interest rate of leasing (as a function of leasing period)
- Leasing Period

The software should be able to return all possible options to a user with any filter. For example, a user might use the following filter

- A range of possible down payment
- The length of leasing period
- The maximum monthly payment

Or

- The maximum down payment
- The maximum monthly payment
- A set of favorite cars

And the software must be able to return the corresponding output with these two filters.

Bonus: Adding other filters is bonus!

*Use [this](#) link for monthly mortgage computation.

GP3: My Best-Fit School

We want to develop a software that consults us with finding the school which fits us the best. This software has a list of school with the following info:

- School name
- Public or private
- Tuition fee
- Minimum GPA and TOEFL requirements

Then, asks each user to enter the following info

- GPA
- TOEFL
- Maximum tuition payable
- Weight of each item

The software returns a ranked list of plausible list of school the user could apply for, accordingly.

Bonus: Creating other filters is bonus!

GP4: Create Your Own Social Network

We want to develop a software that creates a simple social network. This software keeps the following information for each user:

- Name
- Status
- Image (optional)
- List of friends
- Username
- Password
- Pending friend requests

Friendship in this software is reciprocal (i.e.; if A is a friend of B, then B is also a friend of A).

Each user can send a friend request to others and if it is accepted then they become friends.

Bonus: Listing three-hop friendship is bonus.

GP5: Battery Sizing for Residential

Households who have solar PV panels on their rooftop are generating energy. Unfortunately, regardless of how large the solar PV panels are, solar power merely is not sufficient to serve the energy demand of the households though. As an example, there is no sun power at nights. Thus, households are using a combination of solar power and national energy grid to serve their demand. As an alternative, households can buy batteries to serve their demand. A battery of size B , has an energy level of $b(t)$ at any time t , given recursively by

$$b(t) = b(t-1) + P_{in}(t)T_u - P_{out}(t)T_u$$

$$0 \leq b(t) \leq B$$

where $P_{in}(t)$ is the input power at time t , $P_{out}(t)$ is the output power at time t , and T_u is the time unit. Batteries are expensive, though. The price linearly grows as a function of size B . We want to develop a software that given

- Solar power traces
- Power demand traces
- Time unit (T_u)

Can compute the minimum size of a battery that can guarantee we can always serve the demand without using the grid.

Bonus: Computing the electricity bill if the battery size is not enough is bonus.

GP6: Payback Time Calculator for Solar PV Panels

We want to develop a software that calculates the payback time (PBT) for solar PV panels. Denote K the total initial investment to buy solar PV panels. Once installed, solar PV panels generate energy which can be sold to the electricity market and make money out of it. The payback time (PBT) is the time duration needed after installation until the total revenue of the solar PV panels equals the initial investment.

This software takes the following as input:

- K : The total investment
- C : Transmission capacity C
- $S(t)$: Solar power traces
- $p_e(t)$: Energy price traces

and returns PBT. The available solar power is computed by the following:

$$A(t) = \min(C, K * S(t))$$

$$\text{Rev}(t) = A(t) * p_e(t)$$

PBT is then time when

$$\text{Rev}(\text{PBT}) = K$$

Given C and a constant PBT^* , the software returns the maximum initial investment K that satisfies $\text{PBT} < \text{PBT}^*$.

Bonus: Computing ROI is bonus!

GP7: Account Manager Decision Support

Your goal is to design and implement a software that will assist the Account Manager of a company. The software must receive inputs related to the customer's:

- Name
- Address
- Amount of product Y purchased in the past month
- Total amount of revenues coming from this particular customer in the past month

And present:

- The number of customers
- As well as the names of the customers that spent over Z amount of money in the past month for purchases of product Y and are entitled to a discount A% on their next purchase.

Present the customers sorted in descending order.

Bonus: Creating a forecast for the aggregate amount of product Y expected to be sold in the coming months, is bonus!

GP8: Electricity Tariff Recommender System

Your goal is to design and implement a software that will recommend electricity tariffs to customers.

The system will receive as inputs the customer's:

- Name
- Address
- Total Electricity consumption over the past month
- Total amount spend on electricity over the past month

And present the customers with 2 tariff options:

- The customers who spent more than X amount of electricity in the past month will be presented with the tariff option 1 which charges 0.12 euros/KWh
- The customers who spent less than X amount will be presented with the tariff option 2 which charges 0.14 euros/KWh.

Each customer will also be presented with:

- An estimated monthly cost based on the consumption of the previous month.

Bonus: Creating a leaderboard with the customers who were more efficient spenders (lower cost over electricity ratio) and give a discount of 5% to the first 5 members of the leaderboard is bonus!

GP9: Rotterdam's Best Restaurants

Your goal is to design and implement a software that will give food recommendations to people that visit Rotterdam. The system will receive as input the customer's:

- Name
- Address in Rotterdam
- Food preferences (Asian, American, French or Italian)
- Maximum amount of money they are willing to spend

And present to the customers the:

- Top 5 restaurants of this type (food preference) corresponding to their budget (maximum amount willing to pay)

The customer selects the restaurant of his/her preference and the system records this choice.

If the customer selects to do so, the system:

- Presents the customer with a link to the restaurant's website so that the customer can make a reservation

Bonus: Keeping track of how many times each customer selected each restaurant and presenting the user with his/her favorite (most frequently selected) restaurant, is a bonus!

GP10: Airport Helper

Your goal is to design and implement a software that will help airline passengers get information about the airport they are flying from. The system will receive as input the customer's:

- Name
- Flight number
- Airport he/she is flying from
- Date of departure
- Food preference (Asian, American, French or Italian)

The will present the customer with:

- the latest flight updates (if the flight is on time, Terminal the flight departs from, check in desk number)
- as well as a list of the restaurants in the Terminal in question corresponding to the preferences
- and the duty free discounts in this particular airport

Bonus: Keeping track of how many times each passenger flew from each airport and presenting the passenger with his/her favorite (most frequently chosen for flight departure) airport, is a bonus

GP11: School System

Your goal is to design and implement a software that supports teachers at school
A evaluate the performance of their students. The system will get as input the student's

- Name
- Student number
- Course id

And will present the teacher with:

- the grade of this particular student
- as well as the average and standard deviation of the grades in this course this year and in the 5 past years

Bonus: presenting a graph with each students performance in all the courses over the years the student is enrolled in the school, is bonus!

GP12: Library Decision Support

Your goal is to design and implement a software that will assist the manager of the student library to get updates about the books that are currently missing from the shelves. The system will get as inputs the book's:

- Title
- Unique ID
- Date

And will return:

- the name and address of the student that currently has borrowed this book
- as well as the number of the days that he/she is allowed to keep the book.

Also, the system will present the manager with:

- the total number of times that this book was borrowed in the past year
- and the book with the highest number of borrowing times in the past year

Bonus: Create a bonus system that rewards people depending on how fast they return a book (the faster the return, the higher the bonus points). The users with the highest bonus points will receive vouchers to be used in the library's café-restaurant.