



Inbank Software Engineering Internship 2025 Take Home task

Assignment

- Validate TICKET-101 and write a conclusion for it as .md in the project directory. Make sure that the code works according to the requirements. Make sure to highlight what the intern did well as well as places for improvement. *Tip: The more SOLID the better.*
- Highlight the most important shortcoming of TICKET-101 with explanation. Fix it!
- Implement TICKET-102.

Description

Your team is working towards having an automated loan decision engine for fintech. Another member of the team - an intern - implemented the first piece as part of TICKET-101. Fork the repository and help your team to deliver the full project.

[Decision Engine frontend](#) - Repository

[Decision Engine backend](#) - Repository

Backlog:

TICKET-101

TITLE: Implement MVP scope of decision engine

STATUS: Code review

CONTENTS:

Task:

Please design a decision engine which takes in personal code, loan amount, loan period in months and returns a decision (negative or positive) and the amount.

The idea of the decision engine is to determine what would be the maximum sum, regardless of the person requested loan amount. For example if a person applies for €4000,-, but we determine that we would approve a larger sum then the result should be the maximum sum which we would approve.

Also, in reverse, if a person applies for €4000,- and we would not approve it then we want to return the largest sum which we would approve, for example €2500,-. If a suitable loan amount is not found within the selected period, the decision engine should also try to find a new suitable period. In real life the solution should connect to external registries and compose a comprehensive user profile, but for the sake of simplicity this part can be mocked as a hard coded result for certain personal codes.

As the scope of this solution you only need to support 4 different scenarios - a person has debt or a person falls under a different segmentation.

For example :

49002010965 - debt

49002010976 - segment 1 (credit_modifier = 100)

49002010987 - segment 2 (credit_modifier = 300)

49002010998 - segment 3 (credit_modifier = 1000)

If a person has debt then we do not approve any amount. If a person has no debt then we take the identifier and use it for calculating a person's credit score taking into account the requested input.

Constraints:

Minimum input and output sum can be 2000 €

Maximum input and output sum can be 10000 €

Minimum loan period can be 12 months

Maximum loan period can be 48 months

Scoring algorithm. For calculating credit score, a really primitive algorithm should be implemented. You need to divide the credit modifier with the loan amount and multiply the result with the loan period in months. Finally, the outcome should be divided by 10.

If the result is less than 0.1, then we would not approve such a sum.

If the result is larger or equal than 0.1 then we would approve this sum.

$$\text{credit score} = ((\text{credit modifier} / \text{loan amount}) * \text{loan period}) / 10$$

TICKET-102

TITLE: Implement age related restrictions to decision engine

STATUS: Backlog

CONTENTS:**Task:**

In order to make a better loan decision we need to take customer age into account. There are several reasons for it:

- a) we have restrictions on when loans can be offered at all.*
- b) our data shows that younger customers are more prone to have debts.*

In this ticket please implement a solution to restrict our service for customers outside the approved age range. No loans should be given if the customer is underage or older than the current expected lifetime of each respectable country minus our maximum loan period. Frontend should give feedback to the customer if loan was not given due to age constraints. At first only Baltic scope should be implemented.

Constraints or clarifications:

For simplicity sake let's say that personal code is in the same format in all countries and expected lifetimes can be arbitrary.

Upload Criteria:

Please create a public Github or Gitlab repository that can be accessed. Upload it through the link that is in the initial email that has been sent to you which is located under the signature.