Chat analysis part: (Robert) In the app, as discussed with other teams, users will have the ability to chat with other users that are using the app. In the case of abusive language from a user, the recipient can report that message (this is done by the social team). We will have a database table with chat reports where we get the reported message and the reported user's cnp from. If the reported message is deemed offensive by the admin, he can punish the user and then we will get that user's cnp and lower his credit score by 15. Otherwise, he can just dismiss the report. Each time we tax someone's credit score for offensive chats the count of punishments in the users table for this user is incremented. If that person has at least 3 punishments in the past then the credit score will instead drop by that 15 * number of previous punishments. There will also be an API call to the PurgoMalum api that suggests if a text is offensive or not so that it helps the admin in making his decision.

Bill Splitting analysis: (Bolos Mihai) When a registered user wants to split a bill along with other registered users, a new group is created (this group will have the same structure as the group implemented by the social part team, we will take it from them) with all the users that are part of the bill split. In this group there will be, right next to each user, in the list, the share of their bill in bold and the date when they paid their share in italic, right next to the share. If they did not pay their share yet, the field of the date will be empty (no data is shown). Also the payer user will have the report option, as a button, for each user that did not pay their share. When the user chooses to report the user (press the button), the report is sent and the software analyzes the following data: date of the transaction, owed money, history of transactions of both the reported user and the report initiator. It checks the reported user transaction history from the date the bill split was and when the report is made, to see if the reported user made a transaction with the exact amount of money owed to the user with a description that contains any of the words "bill" or "share", but the payment is not made through the bill splitting group. After the check, a gravity factor will be computed (which is a score from 0 to 100) where half of it is based on the time since the share was supposed to be paid (1 day - 3 weeks). This is an interval from 0 to 100, taken proportionally from the 1 - 21 days (so 10 days = 25 gravity score as an example, and anything over 21 days is considered maximum and will get a 50 score). The other half of the risk factor is based on the money owed(1 currency - 1000 currency). The same logic applies here, where this is proportionally from the 1 - 1000 (for example 500 currency is 25 gravity score), and anything over is considered maximum from this interval.

GravityFactor = 1/2(min(50,(T-1) * 50 /20)+min(50,(M - 1) * 50/999))
T = number of days past due
M = amount of money owed

The software will also check to see if the user currently has enough money to pay their share or if the sum of the outgoing transactions made since the report was initiated is greater than their share of the bill (if the reported user couldn't pay their share, even if they wanted to, the gravity factor is decreased by 10%). Also, the software will check if the user paid their shares of other bills at least on 3 different occasions in the past (-5% on the gravity factor), if the user made at least 5 transaction in the previous month to the reporter (-5% on the gravity factor because this might indicate they are friends) and the number of offences(increase the gravity factor with the floor ot 10% of the number of offences (eg: if they have 50 offences +floor(10% of 50)% = +5% to the gravity factor)). These checks will help the software understand the report and make a calculated decision on how much to decrease the credit score. When the user pays their share or 21 days have passed (the maximum time allowed for the user to pay their share), the new credit score is computed based on the gravity factor: newCreditScore = floor(oldCreditScore - 0.2 * gravityFactor). After that the user's credit score is updated, the report and the group are closed.

Stocks transaction history analysis part: (Mihai Balau): A user's credit score will be adjusted based on their stock trading performance, represented by the risk score (an integer in the range [0, 100]) and ROI (return of investment). If they lose more than 35% of their trades, they are considered a bad trader, and their risk score will increase by 5 (cannot exceed 100) for each new investment until they have fewer than 35% losing trades (A losing trade refers to a transaction where the investor sells an asset for less than

its purchase price, in our case the stock market). Their risk score decreases by 5 points per profitable trade, regardless of whether they are considered bad traders or not. Investment frequency risk is classified as low risk for 1-2 trades per day (-5 risk score added), moderate risk for 3-4 trades per day (no changes to risk score), and high risk for more than 5 trades per day (5 risk score added), averaged over the last week and applied at the end of the week. Investment size relative to monthly income is considered low risk if below 5% (-5 risk score), moderate risk between 5-10% (no changes to risk score), and high risk if exceeding 10% (+5 risk score), also applied weekly. The risk score will be applied to the credit score weekly in the following way: the risk score, ranging from 1 to 100, will be divided by 100 and subtracted as a percentage from the credit score. For example, if the user has a credit score of 400 and a risk factor of 20, the new credit score will be 400 - 400 * 0.2 = 320. The credit score cannot go below 100. The base formula for ROI is Amount Gained / Cost of Investment. For each week, it will be calculated for each investment and after this will be an average expressed. Then, if it is lower than 1, the credit score will be decreased with the formula CreditScore = CreditScore - 25 / ROI. If it is greater than 1, the credit score will be increased with the formula CreditScore = CreditScore + 25 * ROI, again credit score must be between [100, 700].

Credits/loans analysis: (Paul) Whenever a user wants to get a new loan, we have to decide if they are eligible for it or not and what interest rate we can give them.

- When deciding eligibility of a user and computing interest rate we need to take into account the next factors:
  - User Credit Score will be in the range [100, 700]
  - User Risk Score will be in the range [1, 100]
  - Interest rate will be computed with the formula
    IR = UserRiskScore / UserCreditScore * 100. This ensures that interest rate is proportional to risk score and credit score
  - Monthly payments are computed by the formula:
    MP = LoanAmount * InterestRate / NoOfMonths + PenaltyFactor
    - Number of Months is computed by the formula:
      NoOfMonths = (RepaymentDate - ApplicationDate) / 12
- A user is eligible for a loan if they meet all of these conditions
  - LoanAmount <= MonthlyIncome * 10
  - UserCreditScore >= 300
  - Monthly Income > 0
  - No open credits that are past their repayment period and the user has yet to pay
  - DebtToIncomeRatio <= 60%. This is computed with the formula:
    DTIR = Total Monthly Debt Amount / Monthly Income * 100
  - User Risk Score <= 70
- Whenever the user misses a monthly payment, a penalty factor will be added to their payment
  - Penalty factor will be computed by the formula:
    PF = DaysOverdue/10

- They will also be assigned a standard time frame in which they can pay off the loan, and if they manage to pay it off in that interval, then their CS will go up, if not, it will go down.
    - CS updates will be computed by the formula:

NewUserCreditScore = min(UserCreditScore + (Loan amount / Monthly Income) + min(TotalDaysInAdvance, 30) - min(TotalDaysOverdue, 100)), 700

- TotalDaysInAdvance is computed by the formula:

TDIA = RepaymentDate - CurrentDate

Loan Data:

- Loan ID, unique for every loan
- User ID, the user identifier
- Loan Amount, which will be given to the user
- Application Date, for when the loan was requested.
- Approval Status: pending, approved, rejected. (initial value will be “pending”)
- Interest Rate (initial value will be 0 until it is assigned)
- Repayment Date, until which the user must repay the loan
- Monthly Payment Amount
- Monthly Payments Completed (initial value = 0)
- Repaid Amount (initial value = 0)
- Penalty, for missing monthly payments (initial value = 0)

Chuck Norris based coin flip system for updating credit score (Boar Victor): The admin will have a button with the text “Chuck Norris” on it which he can press. We will connect to the Chuck Norris joke api (https://api.chucknorris.io/) and extract a joke at the press of that button. We will add the ascii values of all the characters in that joke, modulo them by 10, and subtract or add that amount based on a coin flip (a random number is generated between 0 and 1, anything < 0,5 is heads and the rest is tails). If it is heads we increase the credit score otherwise we lower it with that amount. Also, we will generate a random number between -10 and 10 and add that to the credit score, updating it accordingly in the database.

User analysis profile: (Cristiana) *The system shall allow the admin to view the credit score changes of users on a bar chart, as well as the account information and a list with the activities from which the credit score increased or decreased. The bar chart will be done using clear visual indicators green for increases, red for decreases and yellow for neither, for changes of the credit score based on the previous value (e.g if in day 1 the credit score is 50, and in day 2 the credit score is 60/ 50/ 40, the bar of day 2 is green/ yellow/ red).The shown chart data can be filtered to be weekly, monthly or yearly. The weekly button will show the data from the last 7 days, the monthly button will show the data from the last set of days that add up to a full month and the yearly button will show the data from the last set of days that add up to a full year. The account information that will be displayed is:* userId, First name, Last name, CNP, email, Phone number*; the displayed information will be read-only. In the list with activities, the admin will see each action that modified the user score, with the specified motive and the amount of score gained/ lost in green/ red. The data will reflect real-time credit score updates based on users' financial activities and the interaction*

with the other users. The system will maintain the history of the credit score changes of each user in a time span of a year and a half.

Credit score tips: (Haj Laith) Users will receive tips on how to improve their credit score, and also congratulations (whenever their credit score increases) and shaming messages (whenever their credit score decreases). Tips will be given according to the user's credit bracket. Credit brackets are as follows: 100-299 low, 300-549 medium, 550 to 700 high. Tips will be given when someone's credit score changes. So we must look at all the functions from other features that change the credit score, and in there send tips according to the user's new bracket. Congratulations and shaming messages will be sent every 3rd tip, the low bracket and medium brackets will get shaming messages (we are dystopian after all), and the high bracket gets congratulations messages. Each user will have a tips tab in their menu that contains all the tips and congratulations/shaming messages given to them with the date they were sent next to them.

Credit Brackets:

- Low Credit: 100 to 299
- Medium Credit: 300 to 549
- High Credit: 550 to 700