Sky Technology - Android Programming Test

Loyalty Rewards

1 Scenario

A groundbreaking broadcaster has decided to reward its most loyal customers. A software engineering team, developing the customer account website, is working on the story below.

Display customer's available rewards

As a customer, if I am eligible for rewards, then I want to see which rewards are available based on my channel subscriptions.

Alongside this there will be an Eligibility service that checks whether a customer is eligible for the rewards based on loyalty and billing status.

Determine Customer Eligibility

As a customer, my eligibility for rewards will be determined based on my account number.

2 Instructions

The Account Management team has asked you to create a basic Android app implementation to demonstrate the feasibility of this concept.

Unfortunately the services are not yet fully developed so you will have to mock the interaction with them in your implementation.

3 Acceptance Criteria

3.1 Rewards Service

The following table describes the codes for the channel subscriptions and the associated rewards.

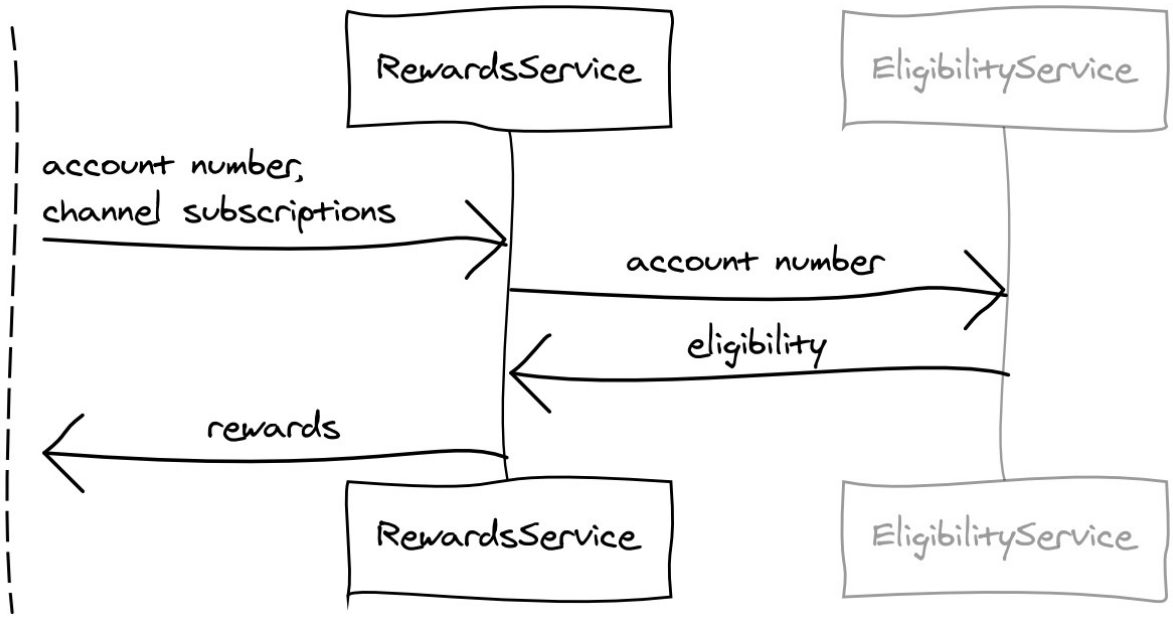
Channel	**Reward**
SPORTS	CHAMPIONS_LEAGUE_FINAL_TICKET
KIDS	N/A
MUSIC	KARAOKE_PRO_MICROPHONE
NEWS	N/A
MOVIES	PIRATES_OF_THE_CARIBBEAN_COLLECTION

3.2 Eligibility Service

The following table describes the EligibilityService output and the expected result:

EligibilityService output	**Description**	**RewardsService result**
CUSTOMER_ELIGIBLE	Customer is eligible	Return relevant rewards according to the customer's portfolio
CUSTOMER_INELIGIBLE	Customer is not eligible	Return no rewards
Technical failure exception	Service technical failure	Return no rewards
Invalid account number exception	The Supplied account number is invalid	Return no rewards and notify the client that the account number is invalid

This is a simple diagram of the interaction between the services:



4 What we look for

We are especially interested in how you structure your code so that it's fully testable, easily extensible, complies with best object-oriented practices and is easy to modify / understand by others.

Please spend no more than TWO HOURS and supply us with your source code and any tests you have written.