Azercell Card System

App is about simple card system including money transfer

Farid Adashov
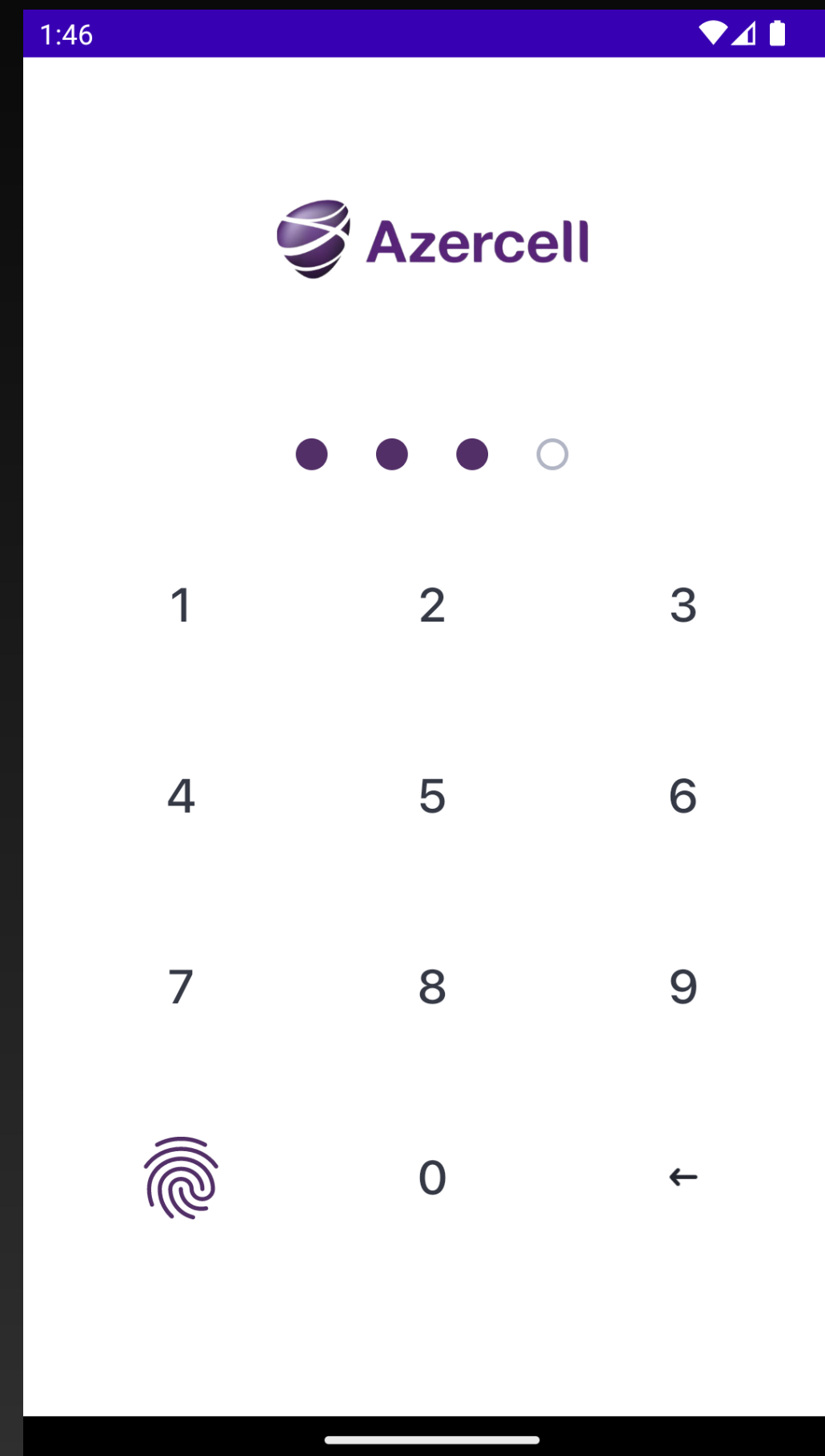
Presentation

- App functionalities will be shown with images and video recording
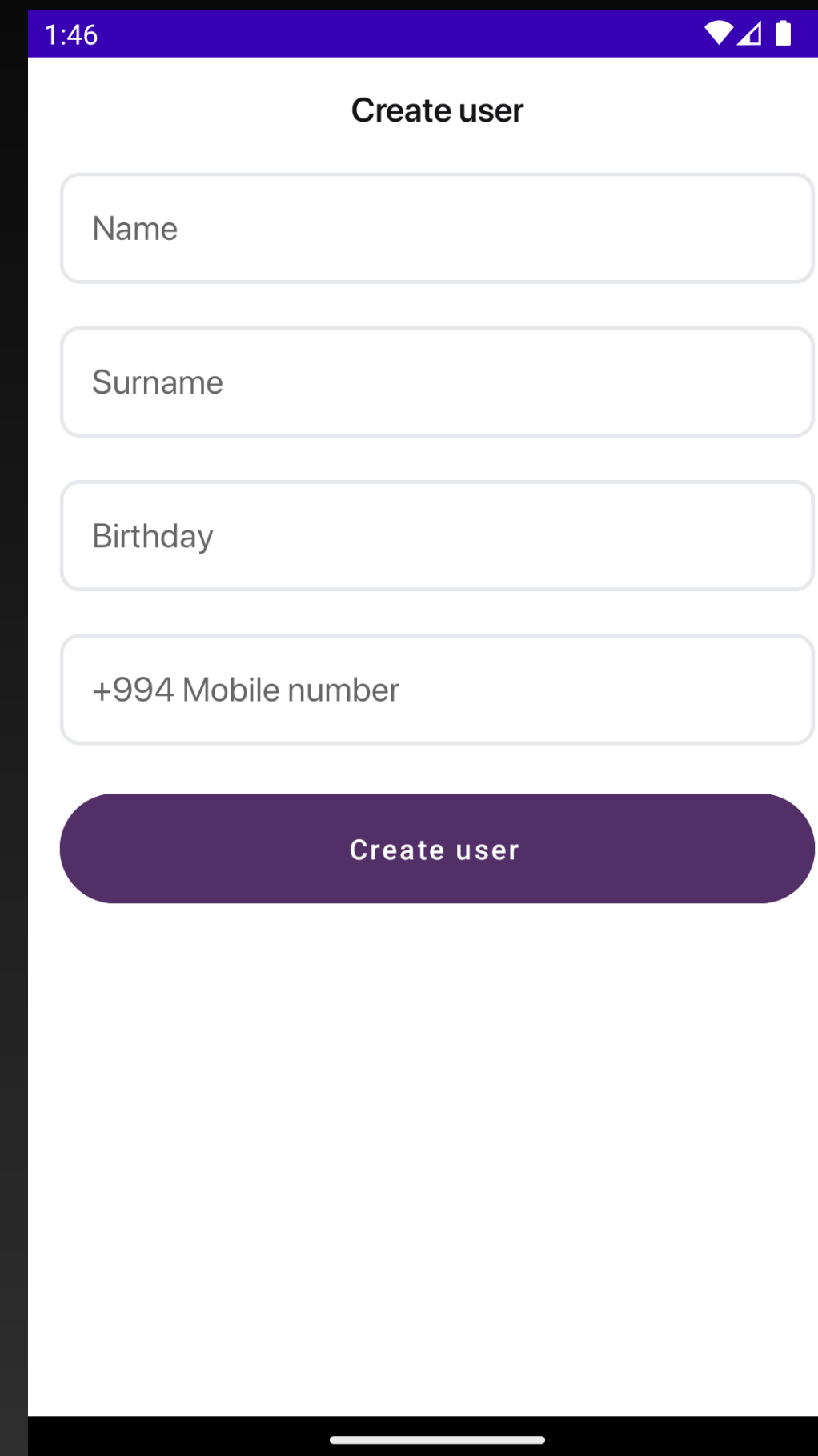
- App structure will be explain

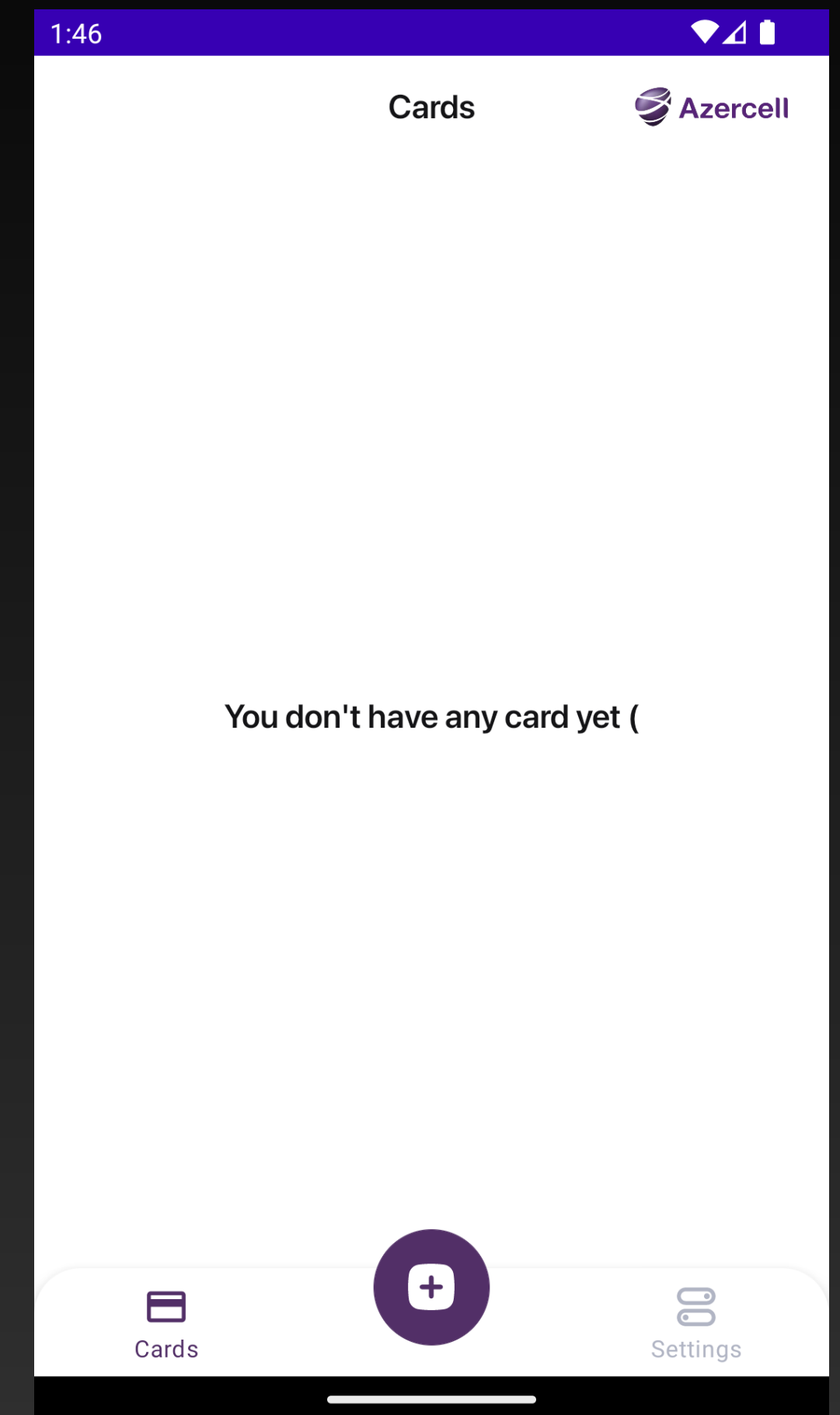- https://github.com/FeridAdashov/AzercellCardSystemApp

Launcher

Pin Login

Default pin is: 1111

Create User

Create user

Name

Surname

Birthday

+994 Mobile number

Create user

You can continue without creating user

Empty cards page

Cards

You don't have any card yet (

Cards   Settings

If there is not any card screen will be like that

Settings

1:47

Settings

🌐 Username ›

🌐 Language
English ›

⬅ **Log out** ›

Cards · Settings

Change language

1:47

Settings

🌐 Username ›

🌐 Language
English ›

⬅ **Log out** ›

Change language ⊠

English ⦿

Russian ◯

Azerbaijan ◯

Logout dialog

1:47

Settings

🌐 Username ›

🌐 Language
English ›

⬅ **Log out** ›

Are you sure? ⊠

Log out

Cancel
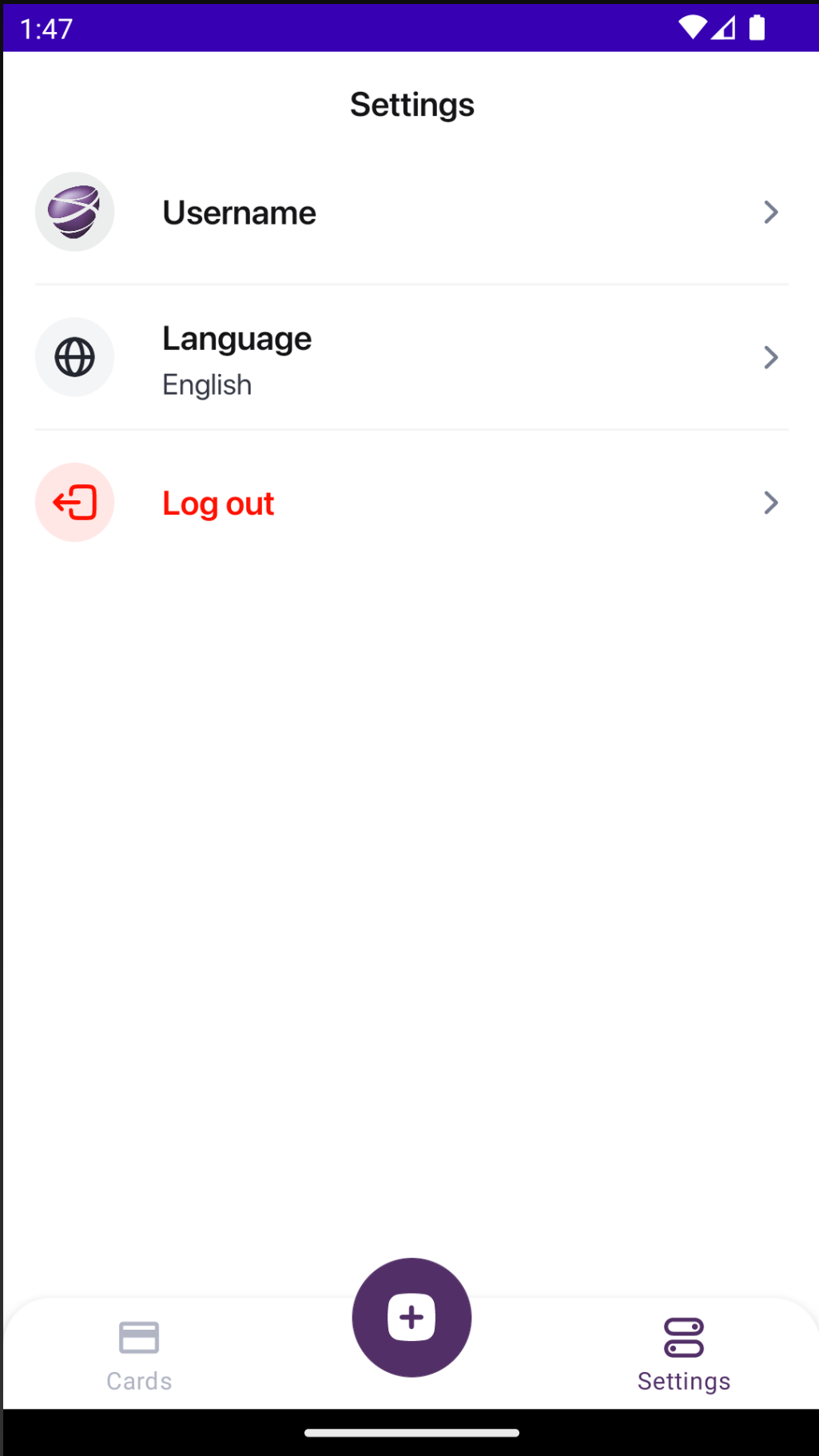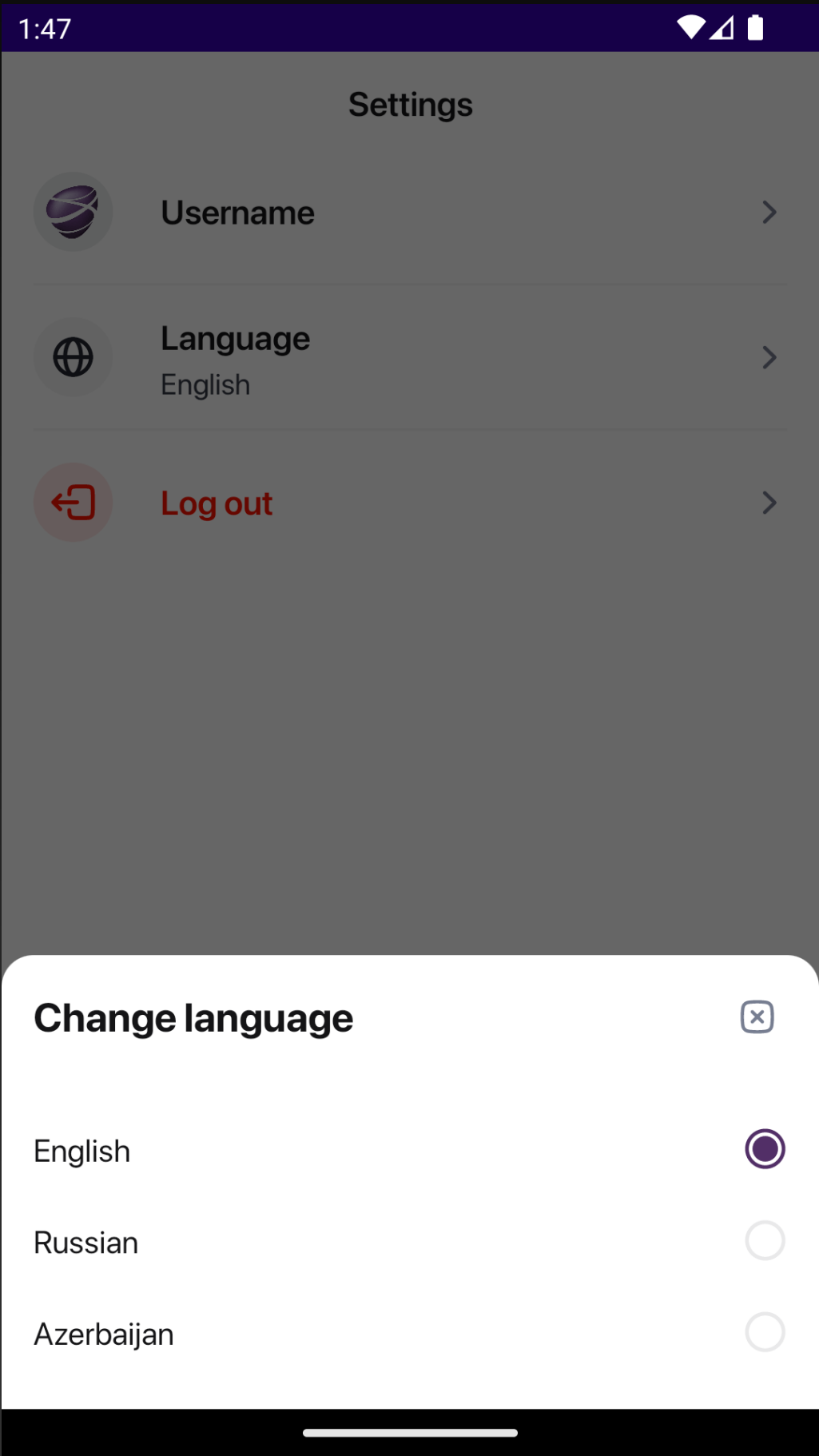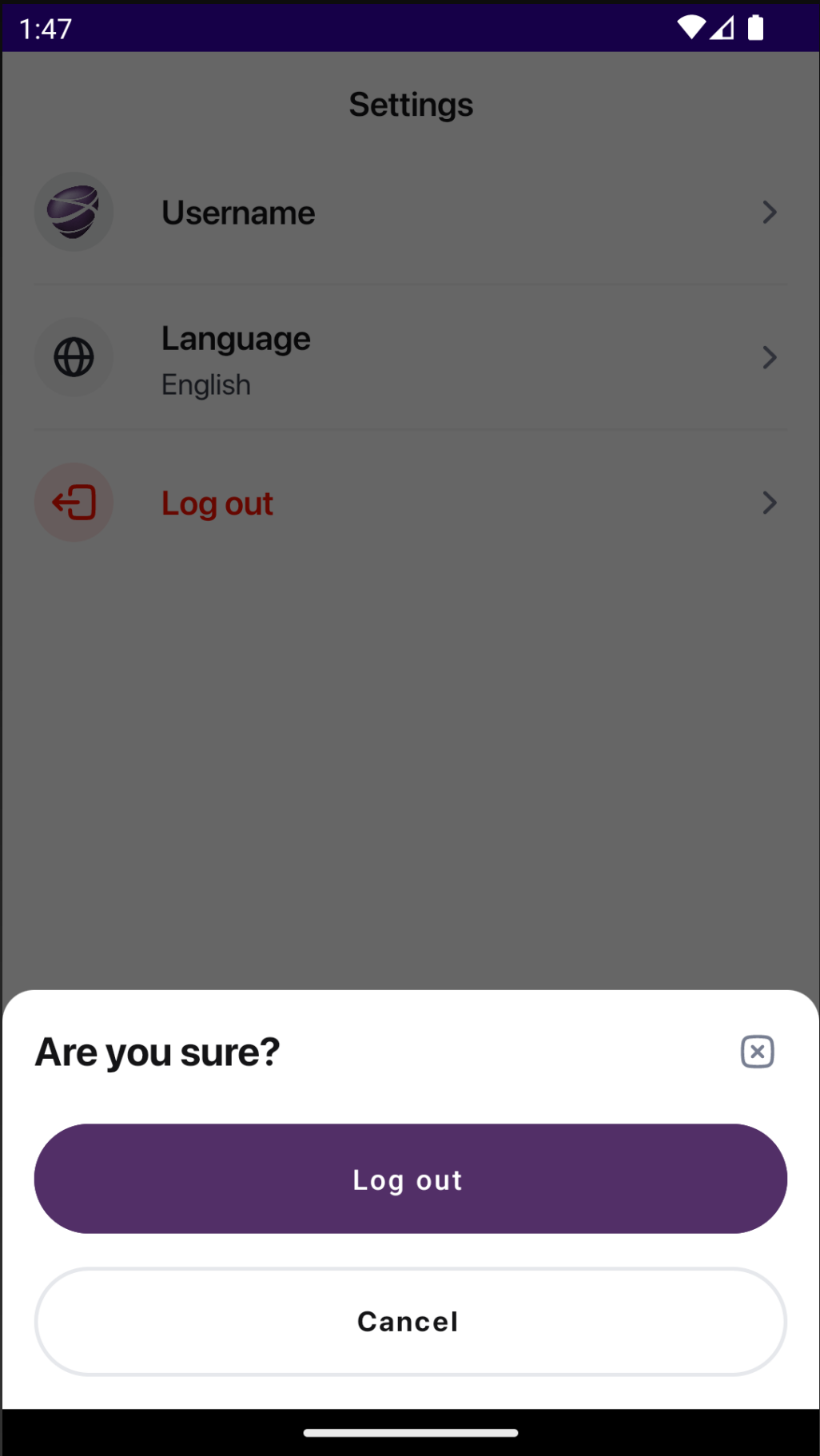
Add card front side

1:48

< Add card ⬡ Azercell

Farid Adashov

4169 0000 0000

EXPIRE

MM / YY *VISA*

Next

1	2	3	—
4	5	6	⎵
7	8	9	⌫
,	0	.	→|

Add card back side

1:48

< Add card ⬡ Azercell

VISA

CVV 111

Add card

Cards screen

1:47

Cards ⬡ Azercell

ghjghjg 👁

4169 566* ** 5333**
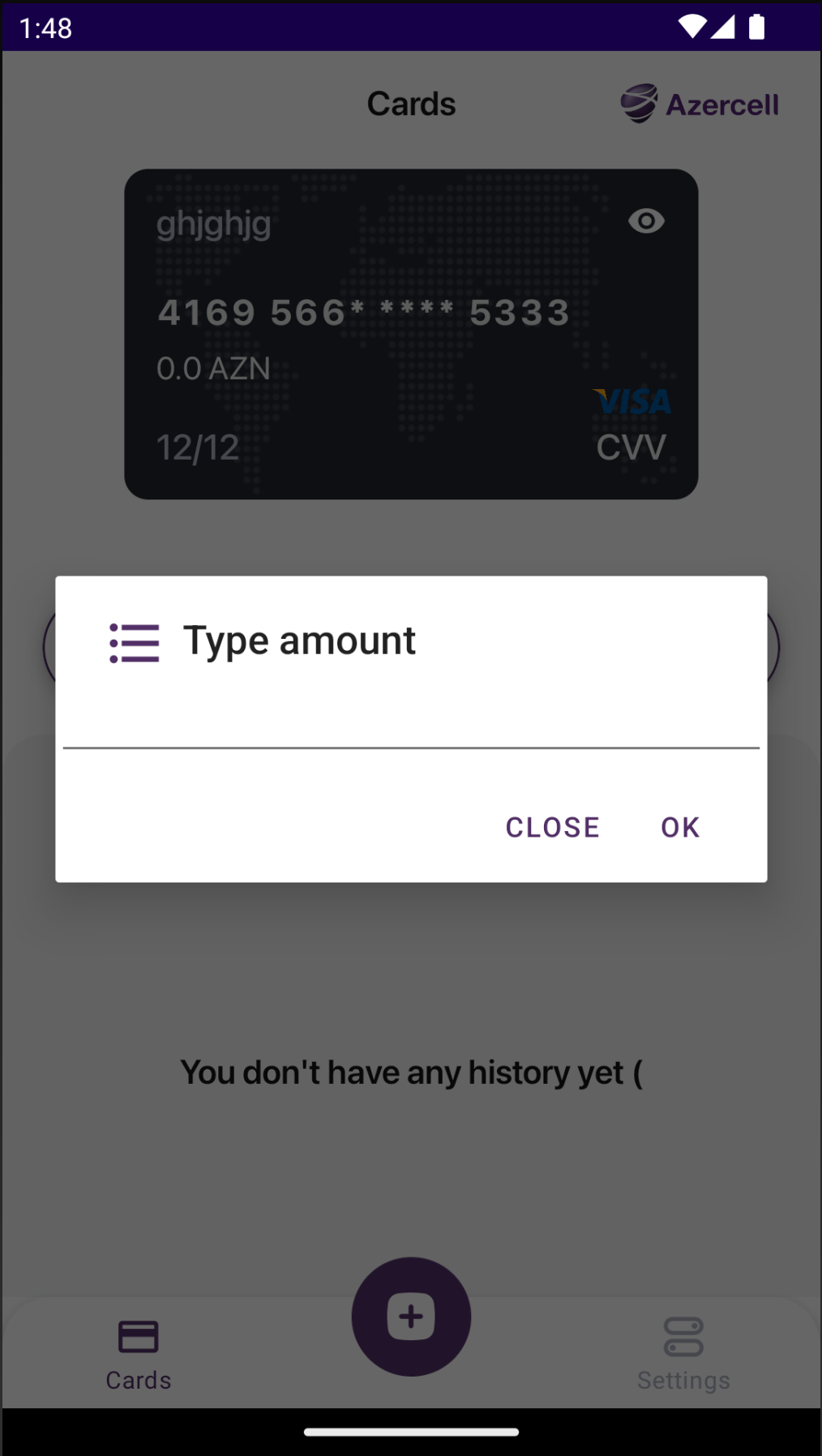
0.0 AZN

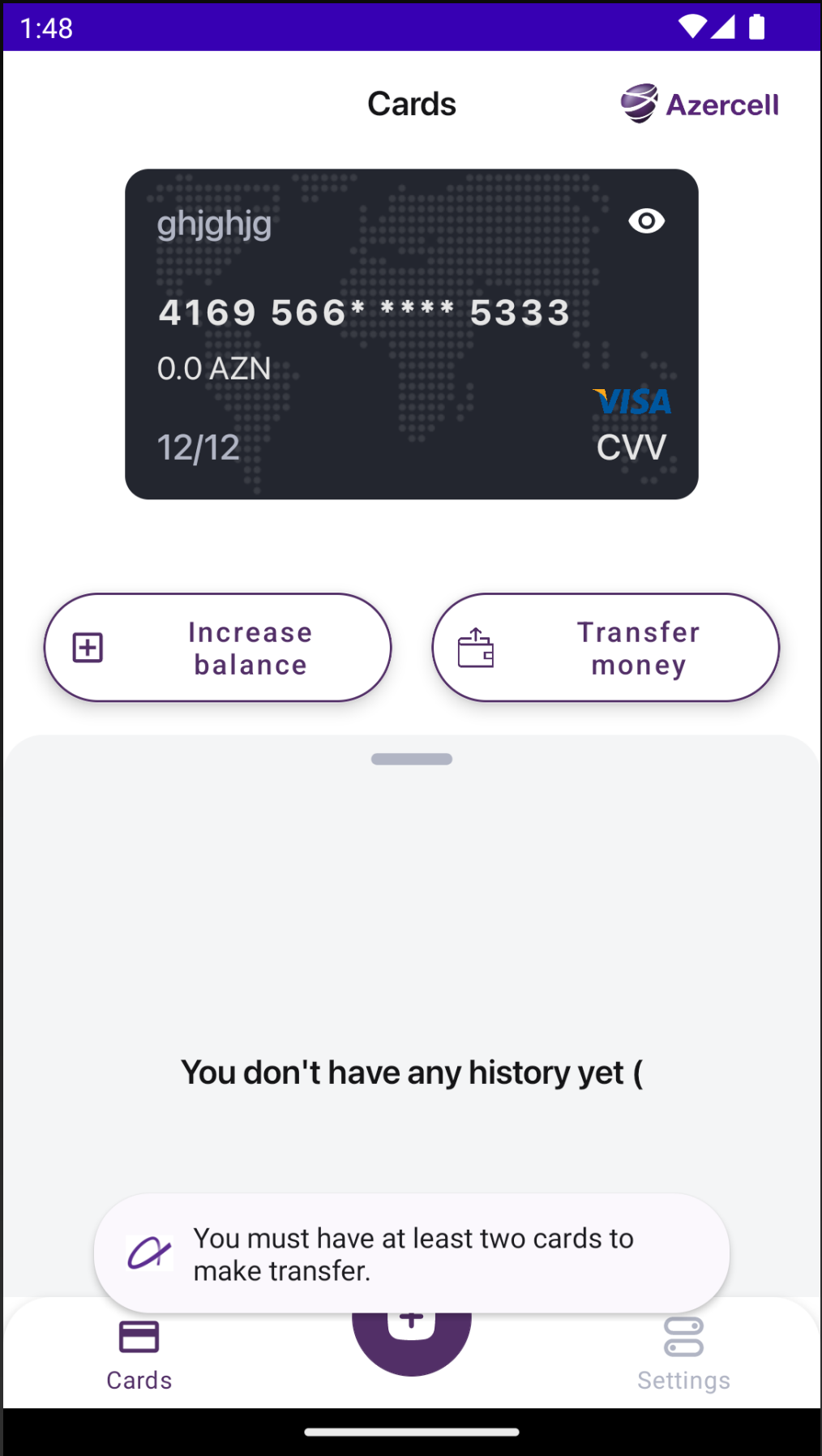12/12 *VISA* CVV

⊞ Increase balance

⊟ Transfer money

You don't have any history yet (
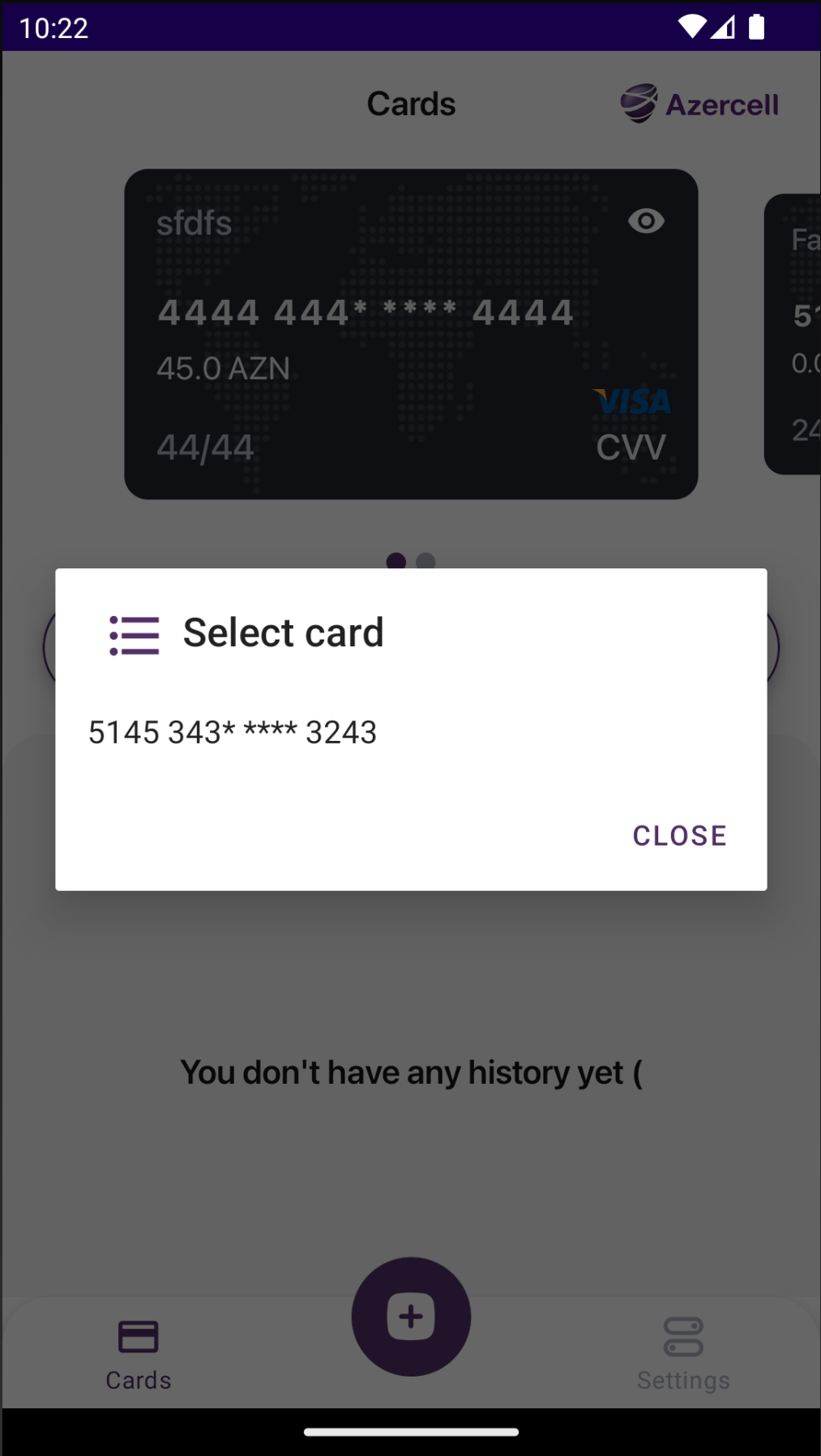
⊞ Cards ⊕ Settings
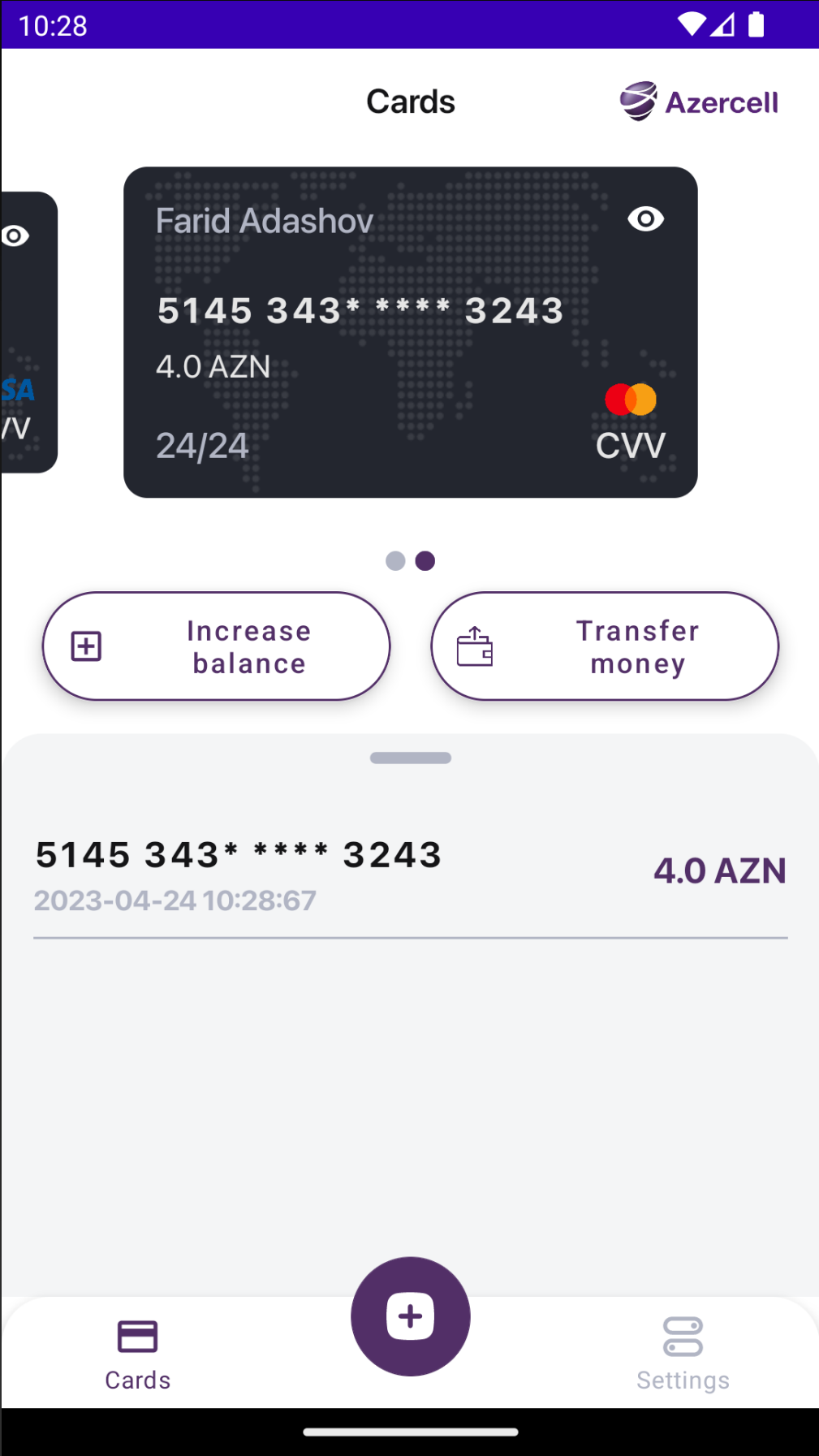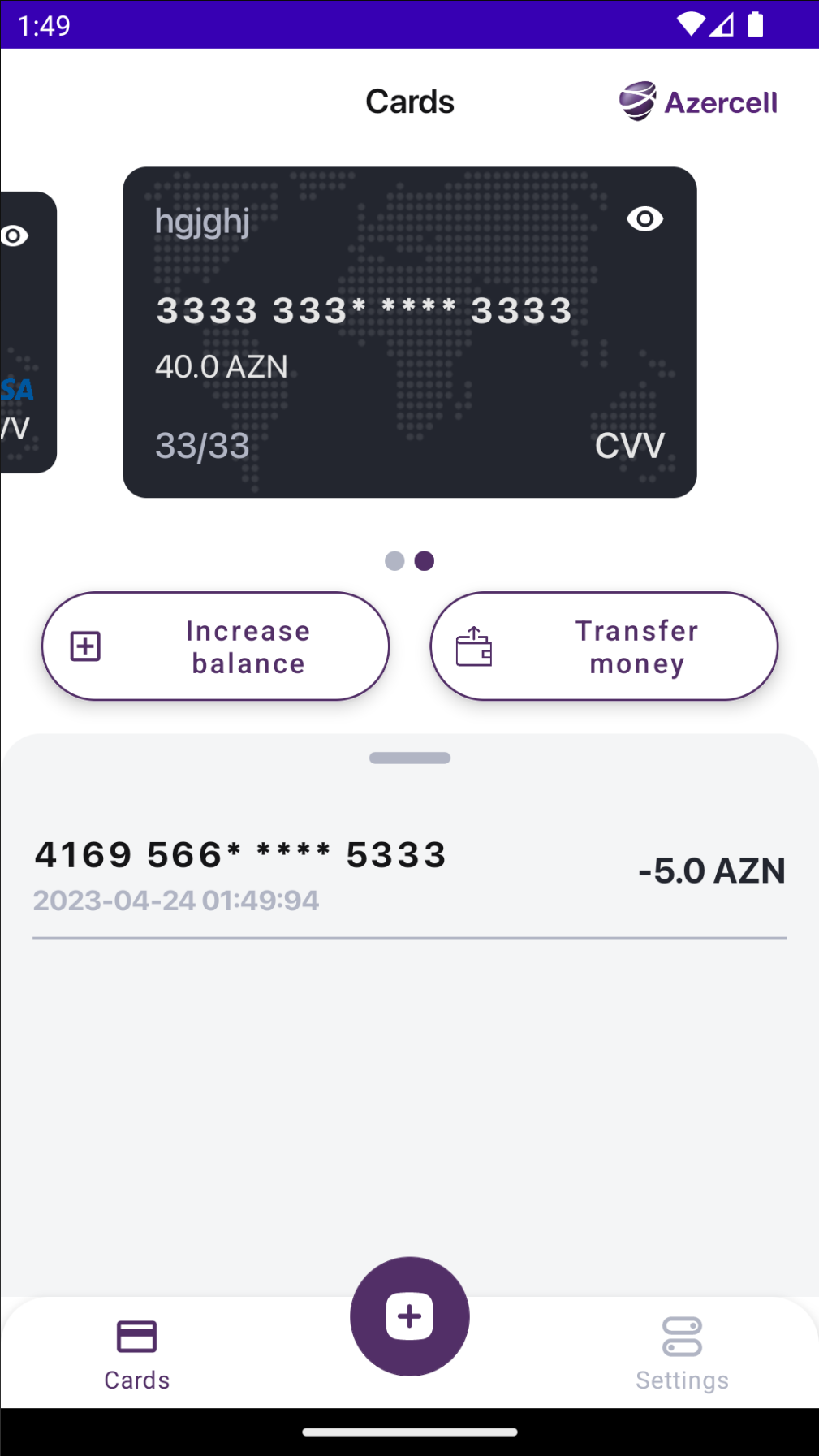
Get amount dialog

Not enough cards

Select card

You must have at least two cards to transfer money

Select card dialog to transfer money

Transfer history

App Structure

- App structure is based on MVVM

- Modules

    - App

    - Domain

    - Data

    - Common

App

- App module is the top part and it keeps UI system itself
  - Dependency injection
  - ViewModels
  - Navigation
  - UI - fragments
  - Adapters
  - Listeners

Domain

- Domain side is used to transfer data between DATA->APP modules
  - Entities
  - Repositories
  - Interactors

Data

- Data side is used to get data from outside. Here i used fake data holder class for this aim.

  - Api - it can be any other data source

  - DTO - data transfer objects. Response classes

  - Mappers - for mapping data to our ui entities

  - Repository implement - Implementation of DOMAIN side repositories

Common

- As the name suggests, here you can add common purpose objects

  - Extensions

  - Utils

  - You can create common purpose Views (Widgets)

Thanks for your time