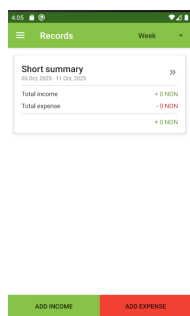


Trabalho Final - Automação de Teste

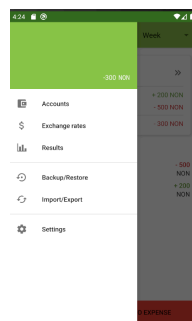
De acordo com os requisitos fornecidos da aplicação MoneyTracker, criar os casos de teste e o projeto de automação de teste com Appium utilizando o padrão Page Objects em python.

Requisitos:

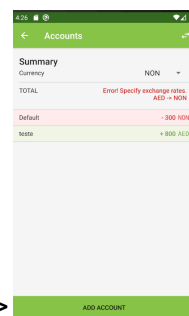
Tela Principal



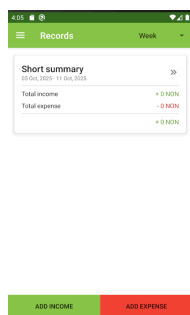
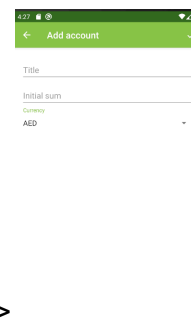
Menu



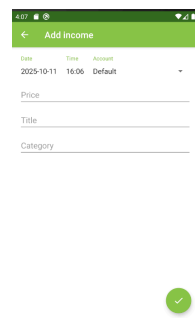
Accounts



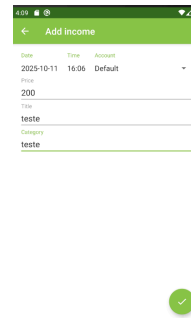
Add Account



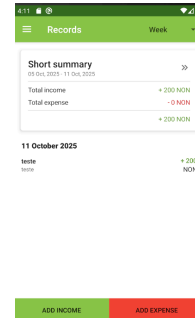
-->



-->



-->

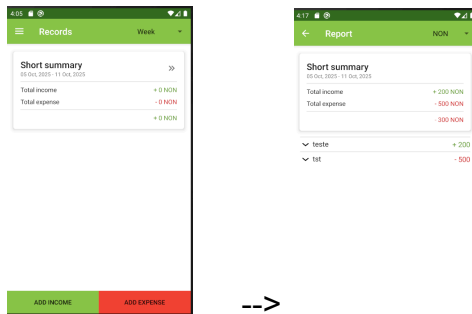


Tela Principal

Add Income

1. Adicionar uma conta "Account"
2. No formulário Account o título pode ter somente letras até 20 caracteres e o campo Inicial sum deve ter somente números até 13 caracteres. Os campos não podem ser vazios
3. Na tela principal escolher um formulário (Add Income/Expense) para testar
 - a. O campo Price possui limite de 1 a 13 caracteres, só aceita números e não pode ser vazio
 - b. Campo Category não pode ser vazio e deve aceitar apenas letras de limite de 1 a 30 caracteres

- c. Campo Title não pode ser vazio e deve aceitar de 1 a 20 caracteres (apenas letras)
4. Pode salvar income/expense de datas no passado e no presente somente (nunca no futuro)
5. Lista de incomes e expenses salvos devem ser exibidos na tela Records. Para exibir registros de datas anteriores basta clicar no seletor Week (parte superior da tela à direita) e selecionar "All time"
6. Incomes/expenses podem ser editados e removidos
7. Não pode cadastrar Incomes/expenses com campos vazios
8. Duplicidade de registros de incomes/expenses não deve ser permitido
9. Adicione pelo menos um registro de cada (income e expense) e na tela Records ao clicar em Short Summary deve aparecer a tela de Report com a lista de registros realizados (incomes ou expenses)



Link da apk MoneyTracker: [Trabalho Appium - Page Objects](#)

A apk é instrumentalizada, depois de instalar no emulador, dê um long clique para abrir o menu *App Info* -> na tela de App Info clique em *Permissions* -> clique em *Files and Media* -> Selecione a opção *Allow management of all files*.

Depois disso a aplicação poder ser executada, testada e ir gerar um arquivo de cobertura no sdcard chamado coverage.ec

Gere o arquivo de cobertura de código

Link do código fonte da aplicação

<https://1drv.ms/f/s!Avu7qFzImegriooiBvzLTRRc34HxKw?e=87LFNg>

Conforme mostrado em aula, as instruções para gerar o relatório de cobertura:

Abrir o projeto moneytracker no Android Studio -> aguardar carregar as bibliotecas -> verificar se o seu java está correto no projeto -> No Android studio abra o arquivo build.gradle que fica na pasta app -> clique no play para executar a task

jacocoFullReport-> se aparecer Build successful, o seu ambiente está Ok caso contrário corrija os erros de Ambiente.

Depois de executar seus testes automáticos, no terminal do Android Studio digite: adb pull sdcard coverage.ec

Na raiz do projeto ir aparecer a pasta coverage.ec, entrando na pasta você obter o arquivo coberage.ec, clique para copiar e cole o arquivo na pasta moneytracker/app/build/outputs/coverage

No Android Studio (build.gradle) execute a task jacocoFullReport e o relatório em html dever aparecer na pasta moneytracker/app/build/outputs

Entrega:

No diretório do projeto final [Trabalho Appium - Page Objects](#) crie uma pasta com seu nome para colocar os arquivos fonte do seu projeto final.

- Coloque o relatório de cobertura de código
- Crie um vídeo identificando você, identifique seu computador, e mostre os testes executando na sua máquina e no fim o relatório de cobertura, o vídeo deve estar na sua pasta do projeto final.

Prazo para entrega: 20/10/2025