O que eu fiz:

Etapa 1

Tratativa do bug 1 – Mensagem -1 quando clica em Remover Comanda e Adicionar Item sem selecionar uma comanda.

Solução:

Tratativa na FrmBoteco.java

Alteração feita:

Coloquei um Catch (exception ex) ao invés de (Throwable ex) e coloquei uma mensagem de dialogo Solicitando para o usuário adicionar uma comanda antes de tentar remover alguma!

Inserir Item: "Favor insira uma comanda antes de tentar inserir um item! "

Tratativa do bug 2 – Mesas repetidas em comandas

Solução:

Tratativa na classe DlgAdicionarComanda.java

Alteração feita:

Foi inserida uma nova função (testaMesa) que manda como parâmetros (textMesa/mesa/facade.getComandas)

textMesa = Campo na label que está indo para o código

mesa = valor do campo

facade.getComandas = é um get que traz todas as comandas da classe BotecoFacade onde tem as comandas.

Na função testaMesa foi feito um “for” correndo toda as comandas com o comanda.size() e se ele encontra um número de mesa igual ele fala que já está em uso para o usuário.

Tratativa do bug 3 – Mesa -1 (menor que 0 ou igual a 0) e Qtde Pessoas (menor que 0 ou igual a 0)

Solução:

Tratativa na classe DlgAdicionarComanda.java

Alteração feita:

Foi feita uma tratativa na função testaVazio, onde agora há um “if” verificando se é inteiro e se o campo digitado é <= 0, caso seja correta essa informação, estoura um erro para o usuário notificando que o número da label é inválido.

Tratativa do bug 4 – Mesa maior que 10

Solução:

Tratativa na classe DlgAdicionarComanda.java

Tratativa na função criada por eu mesmo, testaMesa tem um “if” que caso a mesa for maior que 10 ele não permite a inserção da mesma!

Tratativa do bug 5 – Atualização dos dados no cabeçalho do Layout

Solução:

Tratativa na classe botecoFacade.java

Foi feita a tratativa nas funções getQuantidadeItensConsumo/getQuantidadeClientes/getQuantidadeMesas, foi realizado um “For” dentro dessas funções onde existe um contador que se acresce a cada item inserido e manda para o cabeçalho do Layout atualizado.

Etapa 2

Tratativa do Bug 1 – Não busca valorTotal consumido pela mesa.

Solução:

Tratativa em addItem em Comanda.java

Alteração feita:

Aproveitei que os campos valorTotal, Item, Preço e Quantidade estavam vindo facilmente nesta função e efetuei o cálculo do valorTotal recebendo o valorTotal + Preço do Item \* Quantidade do Item.

Um grande problema foi que na função calculaValorComissao em BotecoFacade.java estava atribuindo o 0 em valorTotal, então fiquei bastante tempo procurando onde estava o problema, foi onde comentei o método (para tratar depois) e ele funcionou corretamente.

Tratativa do Bug 2 – calculaValorComissao

Solução:

Tratativa em calculaValorComissao

Alteração feita:

Efetuei um For each onde ele percorre o Item da comanda que está na comanda e dentro do For efetuei If’s e Else If’s para que a cada item ele testasse o a Descrição do Item e se dependendo de qual descrição fosse ele faria uma multiplicação da porc e no final usei a variável valor para fazer o cálculo do preço do item \* quantidade deste item \* porcentagem dele para atribuir ao valor comissão.

OBS: Tive que efetuar uma alteração no método tratado que trazia a quantidade de produtos consumidos da comanda que já havia sido tratada na etapa anterior, onde na função/método getQuantidadeItensConsumidos foi feito um for percorrendo todos os itens da comanda, onde um contador efetuava o acréscimo fazendo o cont receber o cont + quantidade do item consumido da comanda.

Etapa 3 –

Tratativa em Valor Por Pessoa – Novo botão

Foi criado uma nova classe (DlgValorPorPessoa) em br.com.botecoHaoba.gui.

Nesta classe efetuei uma cópia do que tinha na DlgAdicionarItem e retirei algumas funções.

Criei uma função rachaConta que recebe uma comanda e que pega o valor total da comanda e divide pela quantidade de pessoas que tem na mesa.

Tratativa em Mais Consumido – Novo Botão

Foi criada uma nova classe (DlgMaisConsumido) em br.com.botecoHaoba.gui.

Nesta classe efetuei uma cópia do que tinha na DlgAdicionarItem e retirei algumas funções.