



UFPB

CENTRO DE INFORMÁTICA

PROJETO: Bills Division

DOCUMENTO DE REQUISITOS

Gabriel Souza Cruz Araújo - 20190020539

Kelvin Brenand - 20180005270

Leandeson P. Santos de Araújo - 20200121362

Dário Ribeiro Maracajá - 20180115826

Histórico de Revisões

Data	Versão	Descrição	Autor(es)
31/03/2022	1.00	<ul style="list-style-type: none">Glossário;Descrição;Elicitação de requisitos;	Gabriel Souza Cruz Araújo Dário Ribeiro Maracajá Kelvin Brenand Leandeson P. Santos de Araújo
06/04/2022	2.00	<ul style="list-style-type: none">Elicitação de requisitos;Análise de requisitos funcionais;Análise de requisitos não-funcionais;	Gabriel Souza Cruz Araújo Dário Ribeiro Maracajá Kelvin Brenand Leandeson P. Santos de Araújo
23/04/2022	3.00	<ul style="list-style-type: none">Casos de uso e diagrama;	Kelvin Brenand
20/06/2022	4.00	<ul style="list-style-type: none">Alteração de Linguagem de Programação (Requisito não funcional - portabilidade);Remoção de restrições;	Gabriel Souza Cruz Araujo
27/06/2022	5.0	<ul style="list-style-type: none">Descrição da interface de usuário;	Kelvin Brenand
27/06/2022	6.0	<ul style="list-style-type: none">Diagramas de Arquitetura	Dário Ribeiro Maracajá
28/06/2022	7.0	<ul style="list-style-type: none">diagrama de classes de análise	Gabriel Souza Cruz Araújo Kelvin Brenand

Sumário :

Sumário :	3
1. Introdução	6
2. Descrição geral	8
3. Glossário	9
4. Elicitação de Requisitos	10
5. Análise de Requisitos	11
5.1. Requisitos funcionais	12
5.2. Requisitos não funcionais	14
6. Especificação de Requisitos	16
7. Análise de casos de uso (diagrama de classes de análise)	18
8. Descrição da interface com o usuário	19
9. Diagramas de Arquitetura	20

1. Introdução

O app Bill Division veio para solucionar os problemas na hora de dividir as despesas com amigos e familiares. Visando uma solução rápida e prática na hora de pagar as contas.

1.1. Propósito do documento

Apresentar o planejamento do produto , suas partes envolvidas e quais as funcionalidades do aplicativo.

1.2. Visão geral do documento

Esta introdução fornece as informações necessárias para fazer um bom uso deste documento, explicitando seus objetivos e as convenções que foram adotadas no texto, além de conter uma lista de referências para outros documentos relacionados. As demais seções apresentam a especificação do sistema Bill Division e estão organizadas como descrito abaixo.

- Seção 2 – Descrição geral do sistema: apresenta uma visão geral do sistema, caracterizando qual é o seu escopo e descrevendo seus usuários.
- Seção 3 – Requisitos funcionais (casos de uso): especifica todos os requisitos funcionais do sistema, descrevendo os fluxos de eventos, prioridades, atores, entradas e saídas de cada caso de uso a ser implementado.
- Seção 4 – Requisitos não funcionais: especifica todos os requisitos não funcionais do sistema, divididos em requisitos de usabilidade, confiabilidade, desempenho, segurança, distribuição, adequação a padrões e requisitos de hardware e software.
- Seção 5 – Descrição da interface com o usuário: apresenta desenhos, figuras

ou rascunhos de telas do sistema.

1.3. Documentos relacionados

1. Sem documentos relacionados.

2. Descrição geral

O app Bill Division veio para solucionar os problemas na hora de dividir as despesas com amigos e familiares. Visando uma solução rápida e prática na hora de pagar as contas.

2.1. Motivação

Facilitar a divisão do pagamento de uma conta entre pessoas.

2.2. Problemas identificados

Durante a compra ou consumo em um estabelecimento entre um grupo de pessoas, no final é necessário dividir o quanto cada um vai pagar. Esse sistema busca agilizar esse processo facilitando na hora de saber qual pessoa consumiu e quanto ela consumiu.

2.3. Visão geral do sistema

O sistema é independente e auto-contido, sendo o seu único objetivo gerenciar os custos compartilhados de um grupo de pessoas. Ele irá permitir que o usuário adicione itens/produtos de consumo, e pessoas; que ele modifique os itens e perfis cadastrados; e que ele verifique o total da conta e o total para cada pessoa. Também será possível que uma pessoa se retire da conta, pagando a sua parte, que será registrado em um histórico.

O sistema não irá permitir que o usuário pague a conta e nem faça pedidos ao estabelecimento através do aplicativo, pois ele não tem como objetivo ser um intermediário entre o cliente e o estabelecimento.

2.4. Usuários do sistema

Uma única pessoa do grupo que irá cadastrar os amigos/familiares para realizar a inserção de itens consumidos/comprados por cada um.

2.5. Suposições e restrições gerais

Sem restrições

3. Glossário

Dados do Item: o nome, preço, quantidade e pessoas que irão compartilhar o pagamento deste item.

Dados da pessoa: O nome e os itens que ela irá pagar.

4. Elicitação de Requisitos

4.1. Brainstorming

Esta técnica consiste de reuniões que permitem que ideias surjam e que sejam exploradas. Basicamente é uma técnica para geração de ideias. Ainda acrescenta que Brainstorming é uma reunião de várias pessoas que fazem sugestões de ideias sem que sejam criticadas ou julgadas, isto é, a técnica sugere a exploração de toda e qualquer ideia, fruto da reunião, seja explorada livremente.

4.2. Cenários

Para os usuários, compreender e criticar um cenário pré-definido de suas possíveis interações com o sistema é mais fácil do que defini-las em palavras. Por esse motivo, desenvolve-se um conjunto de interação dos cenários e eles são usados para elicitar e clarear os requisitos do sistema. O cenário começa com um esboço da interação e, durante a elicitação, os detalhes são adicionados para criar uma descrição completa dessa interação.

4.3 Casos de Uso

Os casos de uso são formas de representar possíveis utilizações do sistema por atores utilizando algum de seus serviços. São narradas interações entre o sistema e os atores envolvidos, para atingir um ou mais objetivos. Um caso de uso cria um contrato entre stakeholders de um sistema sobre seu comportamento. É descrito o comportamento do sistema sob diversas condições enquanto o sistema responde a uma requisição de um ator primário, podendo ser uma pessoa, dispositivo físico, mecanismo ou subsistema. São definidas pré-condições e fluxo de eventos primários, onde o ator primário inicia uma interação com o sistema para alcançar determinado objetivo e o sistema responde de acordo com os requisitos dos stakeholders, gerando as pós condições.

5. Análise de Requisitos

5.1. Requisitos funcionais

[RF 01] Adicionar item/Produto

Descrição: O usuário deve ser capaz de adicionar um item. Registrando os dados do item.

Casos de uso relacionados: cadastro de item/produto.

Prioridade: Essencial

[RF 02] Cadastrar outras pessoas

Descrição: O usuário poderá cadastrar no aplicativo as demais pessoas que irão dividir a conta pelo app.

Casos de uso relacionados: Cadastro de usuários

Prioridade: Essencial

[RF 03] Exibir itens/produtos

Descrição: O usuário terá acesso a uma lista com todos itens cadastrados, mostrando também o preço total.

Caso de uso relacionado: Exibição de itens/produtos, Exibição do total da conta.

Prioridade: Essencial

[RF 04] Exibir itens/produtos de uma pessoa

Descrição: O usuário terá acesso a uma lista de produtos específica de cada pessoa, sendo mostrado o custo total para aquela pessoa.

Caso de uso relacionado: Exibição de itens/produtos de um pessoa, Exibição do custo total de um pessoa.

Prioridade: Essencial

[RF 05] Modificar as informações de um item

Descrição: O usuário deve conseguir modificar as informações cadastradas para um item específico. Essas informações são: preço, quantidade, nome, pessoas que estão incluídas na divisão do custo.

Casos de uso relacionados:

Prioridade: Desejável

[RF 06] Retirar pessoa

Descrição: O usuário deverá ser capaz de remover uma pessoa da conta, abatendo o total do perfil do total da conta.

Casos de uso relacionados:

Prioridade: Desejável

[RF 07] Encerrar conta

Descrição: O usuário deverá ser capaz de encerrar a conta, exibindo o total da conta e o total a pagar para cada integrante da conta.

Casos de uso relacionados:

Prioridade: Essencial

[RF 08] Criar conta

Descrição: O usuário deverá ser capaz de criar a conta, para que itens possam ser adicionados.

Casos de uso relacionados:

Prioridade: Essencial

5.2. Requisitos não funcionais

Performance/Eficiência

[NF 01] Comportamento em relação aos recursos

Descrição: Limite de uma (1) conta por aplicativo.

Casos de uso relacionados:

Prioridade: Essencial

[NF 02] Comportamento em relação aos recursos

Descrição: Limite de cinquenta (50) pessoas por conta.

Casos de uso relacionados:

Prioridade: Essencial

[NF 03] Comportamento em relação aos recursos

Descrição: Limite de mil (1000) unidades por item.

Casos de uso relacionados:

Prioridade: Essencial

[NF 03] Comportamento em relação aos recursos

Descrição: Limite de mil (1000) itens por conta.

Casos de uso relacionados:

Prioridade: Essencial

Confiabilidade

[NF 01] Exatidão/Precisão

Descrição: O preço total da conta, e total para cada pessoa deve ser aproximado em duas casas decimais depois da vírgula.

Casos de uso relacionados:

Prioridade: Importante

Restrições de desenho/projeto

[NF 01] Banco de dados

Descrição: O sistema deve usar o banco de dados NoSQL para manter os dados durante o uso, e caso o sistema seja encerrado.

Casos de uso relacionados:

Prioridade: Desejável

[NF 02] Restrições de Recursos

Descrição: O sistema deve funcionar sem acesso à internet

Casos de uso relacionados:

Prioridade: Essencial

6. Especificação de Requisitos

Adição de elementos

UC001 - Adicionar item

O usuário consegue adicionar itens à conta e especificar as pessoas que estarão ligadas ao item.

Ator: Usuário (Humano)

Prioridade: Essencial

Entradas e pré-condições: O usuário passa os dados do Item.

Saídas e pós-condições:

É adicionado um item à conta geral, e a conta das pessoas ligadas ao item.

Fluxo de Eventos Principal:

- 1 - Esse caso de uso se inicia quando o usuário seleciona a opção: “Adicionar item a conta”
- 2 - O sistema apresenta um formulário a onde o usuário deve ser capaz de inserir os dados do item.
- 3- O caso de uso termina quando o usuário seleciona a opção: “Finalizar”.

UC002 - Adicionar pessoa

Ator: Usuário (Humano)

Prioridade: Essencial

Entradas e pré condições: É informado o nome da pessoa.

Saídas e pós condições:

Adiciona uma pessoa a conta, que poderá dividir os preços dos itens.

Fluxo de Eventos Principal:

- 1- Esse caso de uso começa quando o usuário escolhe a opção: “Adicionar pessoa a conta”
- 2- O sistema apresenta um formulário onde o usuário será capaz de informar o nome da pessoa.
- 3 - O caso de uso se encerra quando o usuário seleciona a opção: “Finalizar”.

Exibir Elementos

UC 001 - Exibir a conta geral.

O sistema exibe ao usuário uma lista de todos os itens adicionados na conta.

Ator: Usuário (Humano)

Prioridade: Essencial

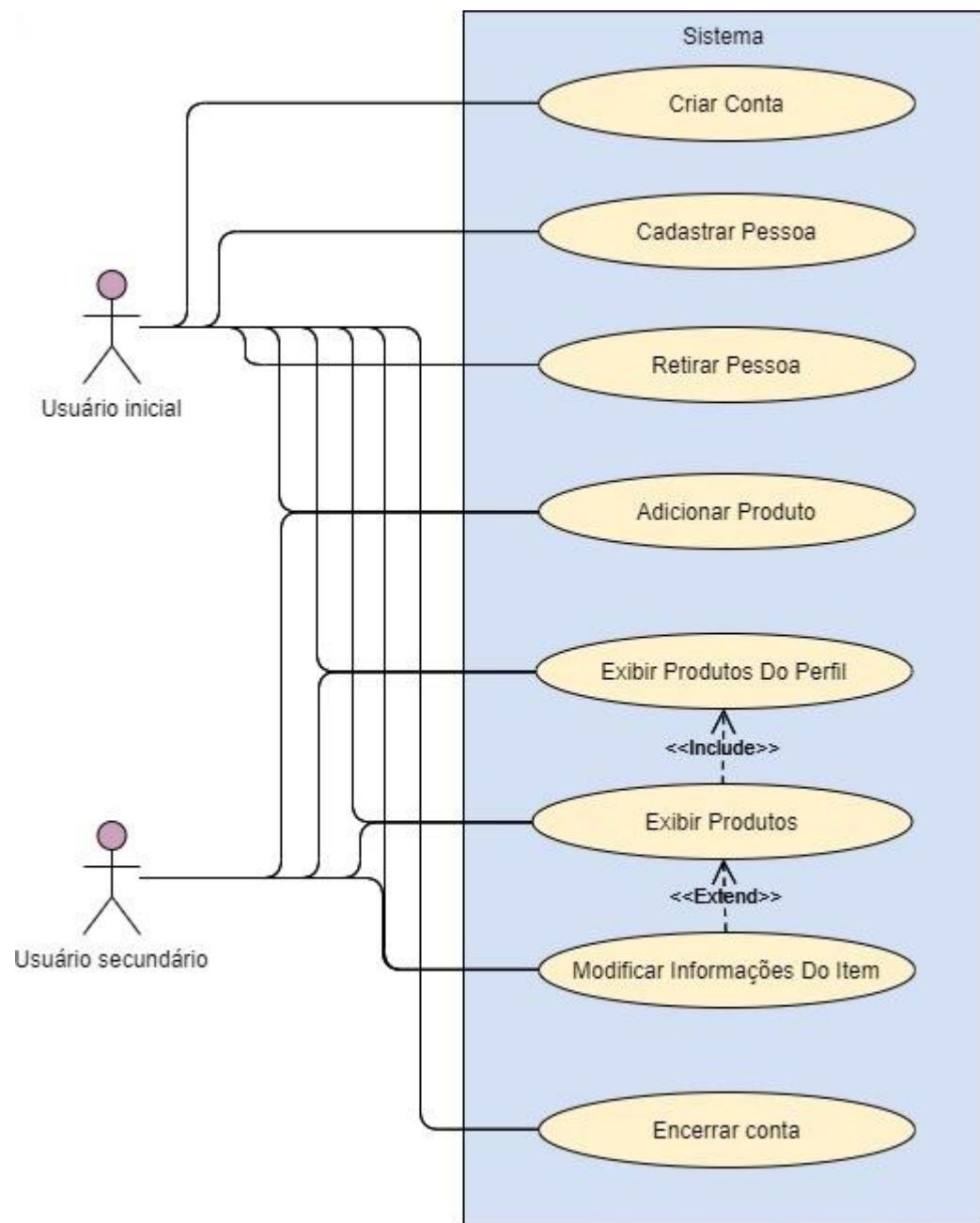
Entradas e pré condições: Deve haver itens adicionados

Saídas e pós condições:

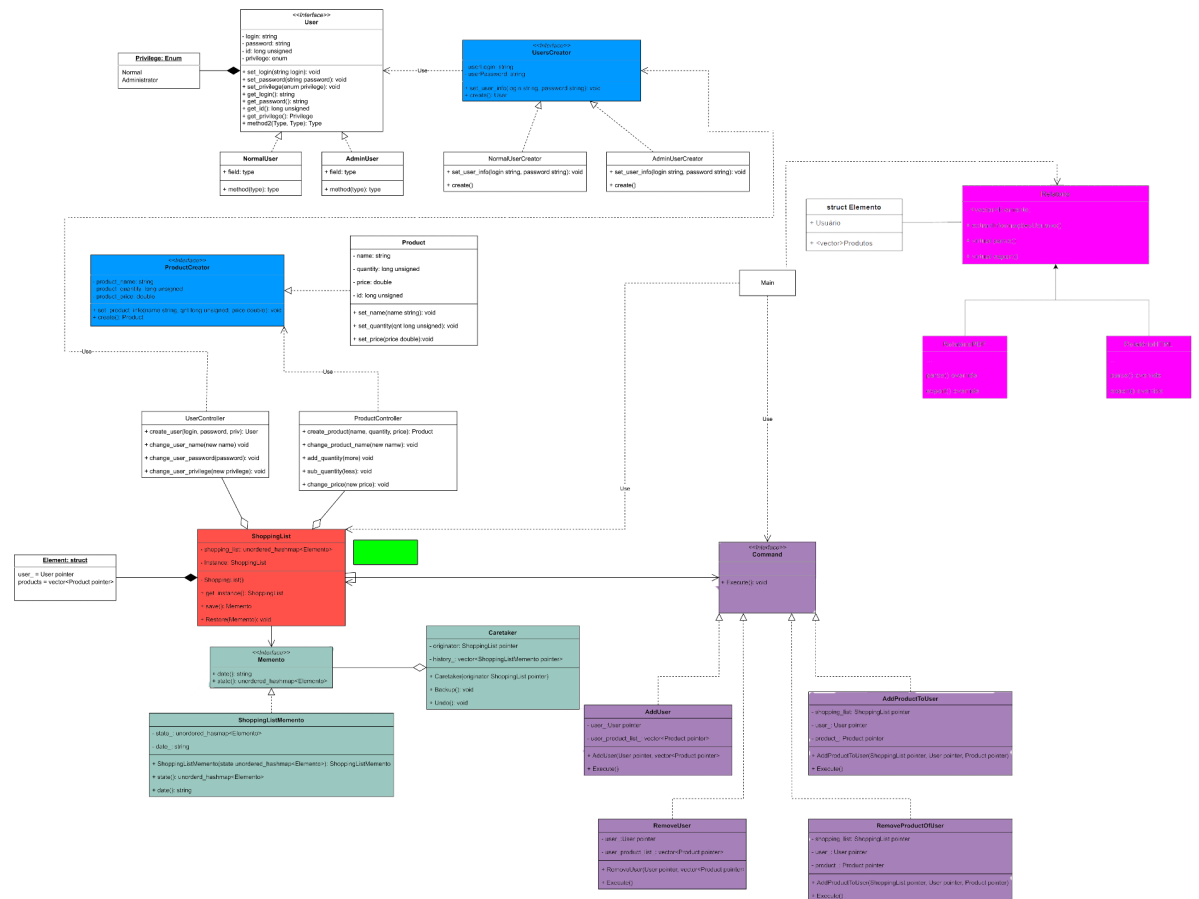
Fluxo de Eventos Principal:

- 1- Esse caso de uso se inicia quando o sistema é iniciado.
- 2- Esse caso de uso aparece depois de toda adição de item ou pessoa, com a lista atualizada.
- 3- Esse caso de uso é encerrado junto com o sistema.

6.1. Diagrama Casos de Uso



7. Análise de casos de uso (diagrama de classes de análise)



- Padrões *Facade* e *Singleton*
 - Classe: *ShoppingList*
 - Objetivo: Criar uma classe que abstrai e simplifica a interação do cliente com as entidades *User* e *Product*.

- Padrão Factory method
 - Classes: *ProductCreator* e *UserCreator*
 - Objetivo: Definir uma interface comum para criar objetos, sem que o cliente precise conhecer uma classe concreta, e com isso isolar quem usa e quem cria.

- Padrão Template method
 - Classes:
 - Classe abstrata: Relatorio
 - Classes concretas: RelatorioTXT e RelatorioHTML
 - Objetivo: Definir o esqueleto de um algoritmo para gerar um relatório de acessos do sistema, deixando para as subclasses apenas a tarefa diferente que é gerar o relatório num formato específico, porém todos terão as mesmas informações que por sua vez foram definidas no template.

- Padrão Command
 - Classes:
 - Classe Comando: Command
 - Classes ComandoConcreto: AddUser, RemoveUser, AddProductToUser e RemoveProductOfUser
 - Objetivo: Implementar uma fachada de comandos, permitindo parametrizar diferentes requisições e facilitando futuras adições de novos tipos de comandos.

- Padrão Memento
 - Classes:
 - Classe Fonte: Estrutura de dados (coleção) de Pagamento mantida pela classe ShoppingList
 - Classe Memento: Memento, ShoppingListMemento
 - Classe Zelador: CareTaker
 - Objetivo: Desfazer uma ação de atualização (restaurar um estado anterior).

8. Descrição da interface com o usuário

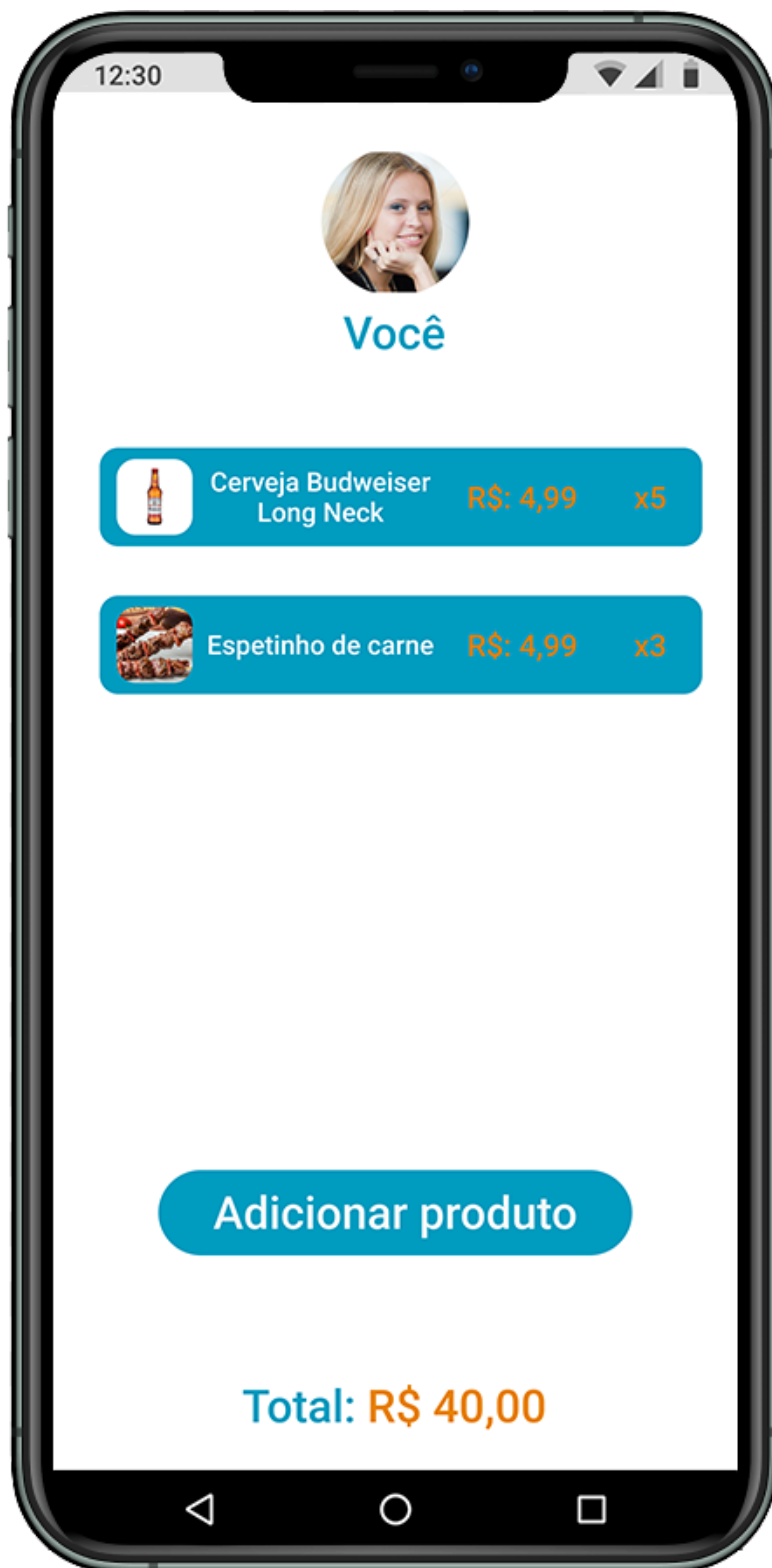
Tela de login:



Tela principal com todos os participantes de uma conta:



Tela individual de cada participante de uma conta:



9. Diagramas de Arquitetura

