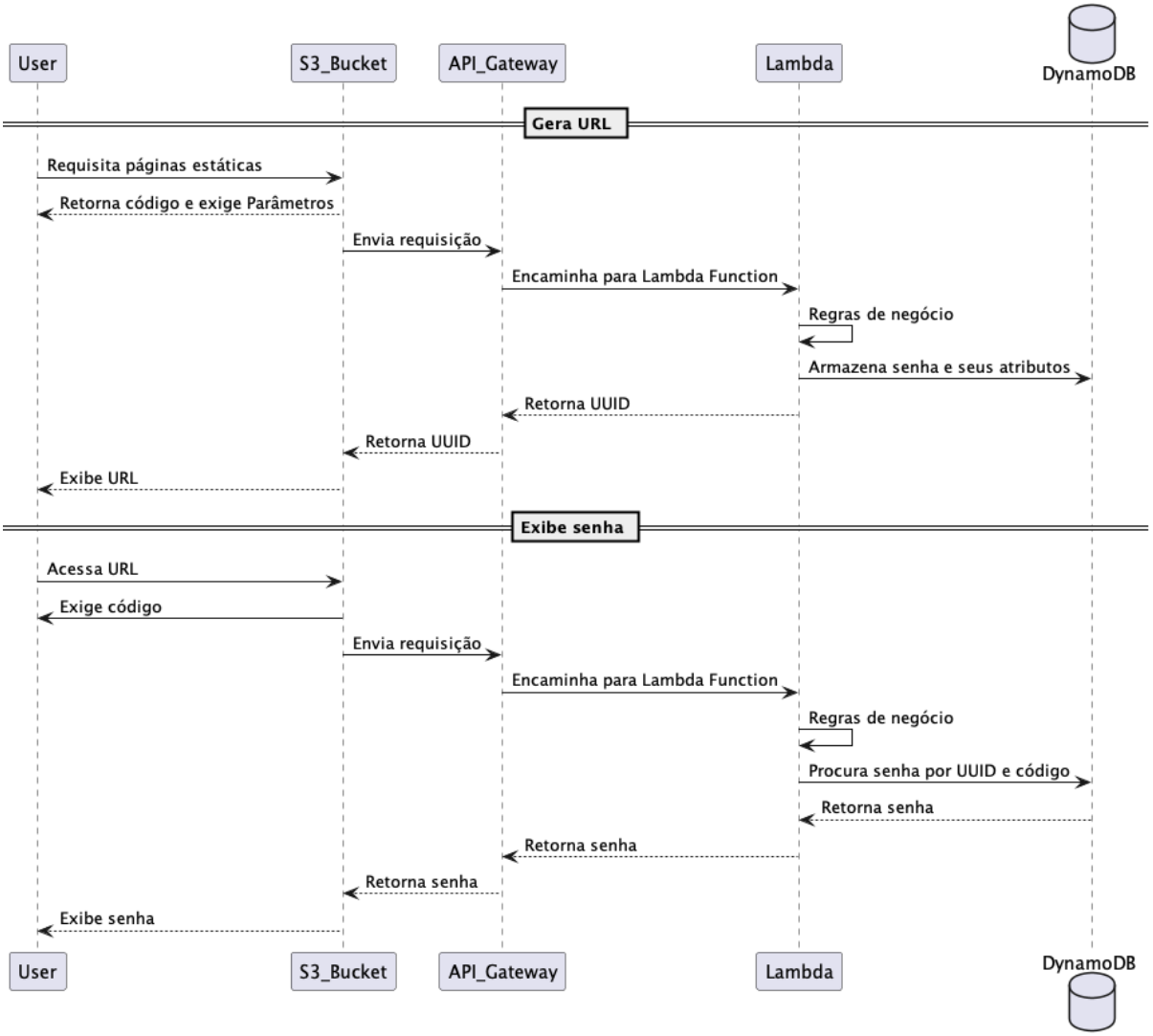

Resumo

Primeiramente, fiz uma alteração no fluxo do projeto para adicionar um código de 5 dígitos gerado aleatoriamente quando acessamos a página de gerar link. Este código é requisitado ao acessar o link para ter acesso a senha, desta forma se algum usuário mal intencionado tiver acesso ao link também teria que ter acesso ao código. Seria mais performático e seguro se implementássemos dois itens:

- *"Schedule"* responsável por limpar todos os objetos do banco de dados que já estão inválidos.
- Limite de tentativas de erro ao digitar código incorreto.

Mas não desenvolvi estes itens.

Tecnologias

Considerando os requisitos e se tratando de um Serverless Application Model, aqui estão as minhas escolhas:

- HTML + JavaScript + Bootstrap: para desenvolver o Front-End estático.
- Amazon S3: para armazenar os arquivos estáticos e disponibilizar o consumo do mesmo.
- Amazon API Gateway: para gerenciar e monitorar nosso Back-End (API HTTP) através de eventos.
- AWS Lambda: para executar nosso código Python *"Serverless"* através de eventos recebidos pelo API Gateway.
- Amazon DynamoDB: para armazenar os objetos criados pelo Back-End de forma rápida. Escolhi este banco de dados por ser NoSQL, já que não temos a necessidade de relações entre tabelas ou objetos complexos, e sim uma demanda de alta escalabilidade.
- AWS Identity and Access Management: responsável por controlar o acesso das funções lambda, API Gateway e DynamoDB através de "policies".
- AWS Systems Manager Parameter Store: Para armazenar os pares de chave RSA com mais segurança.

Desenvolvimento

Basicamente, o usuário recebe um código e escolhe se quer uma senha customizada ou por padrão. Depois decide por quantas horas poderá visualizar e quantas vezes também. Ao clicar em "gerar link" irá receber um link para visualização de senha, que ao ser acessado terá que digitar o código para visualizar a senha.

Para o usuário escolher gerar a senha através de parâmetros, terá que decidir entre 3 opções: Senha simples (Apenas letras e números),Senha média (Letras maiúsculas e minúsculas e números) ou Senha forte (Letras maiúsculas e minúsculas, números e caracteres especiais). A opção "Customizada" permite que digite sua própria senha.

A criação da URL ficou como responsabilidade do front-end, concatenando com a resposta do back-end após envio do formulário.

As lógicas de controle de visualizações, expiração da senha e geração de senha automática foram implementadas no back-end, pois é uma tarefa que exige mais segurança e controle do sistema. Além de evitar que o fluxo para geração de senha seja exposto, manter no back-end permite fácil manutenção do sistema.

Para a criptografia da senha optei pela assimétrica, já que possui mais segurança e a senha será armazenada no banco de dados. Se optássemos por simétrica, a mesma chave seria utilizada para criptografar e descriptografar a informação, o que pode gerar vulnerabilidades se a chave for comprometida. Neste caso as chaves podem ser RSA OAEP 256 bits.