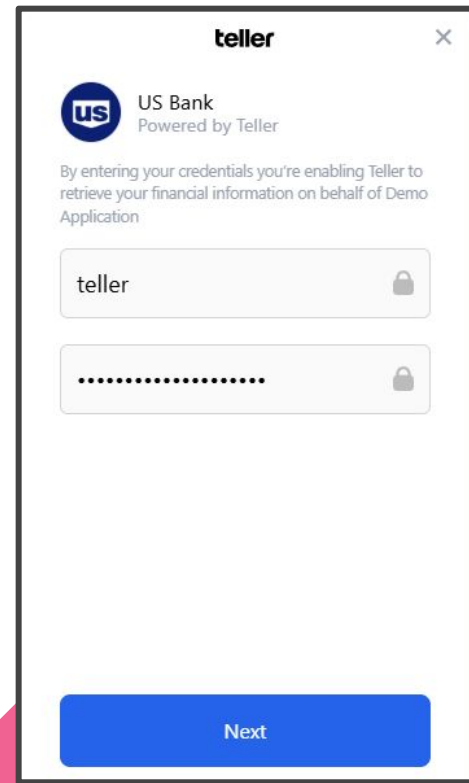


Sprint Week 2 - Trackify App

Sadiq Al-Humood, Joseph Lee

Teller Connect

- Client-side UI component
 - Users will utilize with it to link bank accounts
 - Handles credential validation
 - Multi- factor authentication
 - Account selection
 - Generates access token - used to access user's account (Teller API)
- Current mode: Environment.SANDBOX
 - Mockup data that we want to pull for testing
- For the future: Environment.PRODUCTION
 - Link actual bank account data



The image shows a mockup of a web application window titled "teller" with a close button (X) in the top right corner. Inside the window, there is a logo for "US Bank" with the text "Powered by Teller" next to it. Below the logo, a message states: "By entering your credentials you're enabling Teller to retrieve your financial information on behalf of Demo Application". There are two input fields: the first is labeled "teller" and has a lock icon on the right; the second is a password field represented by dots and also has a lock icon on the right. At the bottom of the window is a blue button labeled "Next".

mTLS Authentication for Teller API

- Firebase Function - bridge for API request from Teller
- Firebase config: store private key and certificate
 - Given by Teller API
- TransactionViewModel file in Jetpack Compose
 - Handle proper calling of Firebase Function
 - Error: API request failed; never reached Firebase Function
 - Cannot fetch transaction data

Function	Trigger	Version	Requests (24 hrs)	Min / Max Instances	Timeout
transactions us-central1	HTTP Request https://transactions-annb46pt9q-uc.a.run.app	v2	0	0 / 100	1m

Items per page: 25 1 - 1 of 1

```
const functions = require("firebase-functions");
const fs = require("fs");
const https = require("https");
const axios = require("axios");

exports.transactions = functions.https.onRequest(async (req, res) => {
  try {
    const certificate = functions.config().teller.certificate;
    const privateKey = functions.config().teller.private_key;

    // Write secrets to temporary files
    const certPath = "/tmp/certificate.pem";
    const keyPath = "/tmp/private_key.pem";
    fs.writeFileSync(certPath, certificate);
    fs.writeFileSync(keyPath, privateKey);

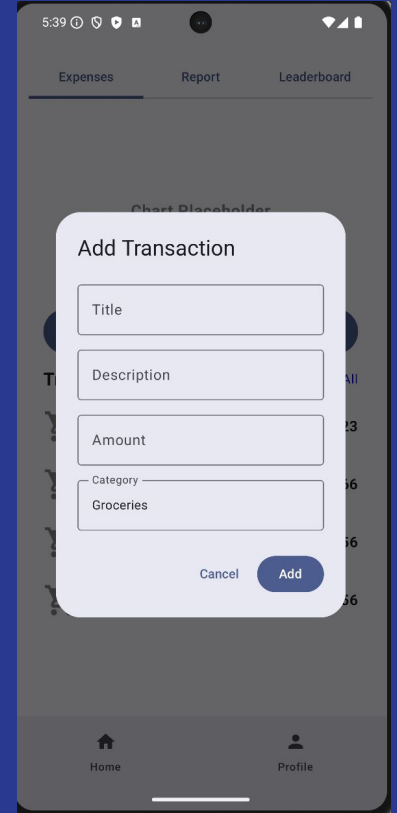
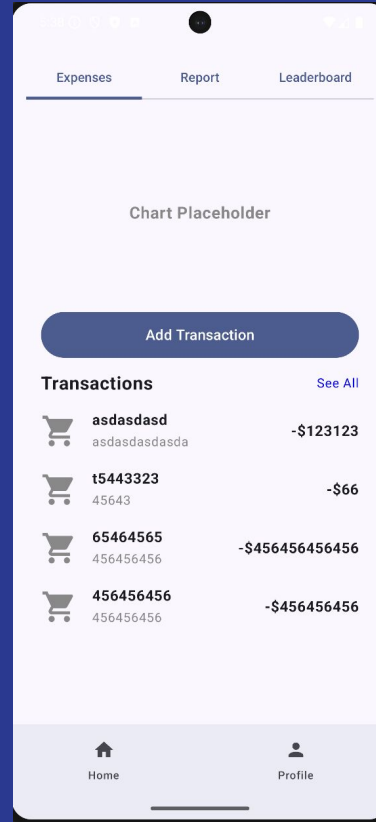
    // Configure mTLS
    const agent = new https.Agent({
      cert: fs.readFileSync(certPath),
      key: fs.readFileSync(keyPath),
    });

    // Fetch transactions from Teller API
    const accessToken = req.headers["authorization"];
    if (!accessToken) {
      res.status(400).send("Missing Authorization header");
      return;
    }

    const response = await axios.get("https://api.teller.io/transactions", {
      headers: {Authorization: `Basic ${accessToken}`},
      httpsAgent: agent,
    });

    res.json(response.data);
  } catch (error) {
    console.error(error);
    res.status(500).send(error.message);
  }
});
```

UI Elements



For the next 2 weeks...

- Push notifications
- Finalize API integration
- Leaderboard Firebase
- Compose Chart



Demo!