Bârsan Alexandru
Stafie Ștefan

Cloud computing
3 June 2021

Case study

This application provides businesses that don't have an e-commerce website a platform to display their discount coupons to clients, which they can later use. Such services are already known as coupons which are specific to a single business. The problem with the analogue coupons is that you can lose them, have them stolen or have them deteriorate. Furthermore, most businesses make their own coupon system and clients have a hard time keeping track of all the offers.

We want to improve the client's experience by making them accessible from a digital space. Our app will help display all the coupons from your phone in a user-friendly way. Every store that is logged in and has bought an API key could add and remove discounts. Users will be able to see the stores that have created an account and if they subscribe to a specific store, they will be able to view and use the discounts.

Existing solutions

Shopify Inc. is a Canadian multinational e-commerce company headquartered in Ontario. It is also the name of its proprietary e-commerce platform for online stores and retail point-of-sale. Shopify offers online retailers a suite of services including payments, marketing, shipping and customer engagement tools. The drawback with a big e-commerce like Shopify is the fact that if a store wants to send out discounts to users, they would have to create a website on their platform and pay for a full subscription.

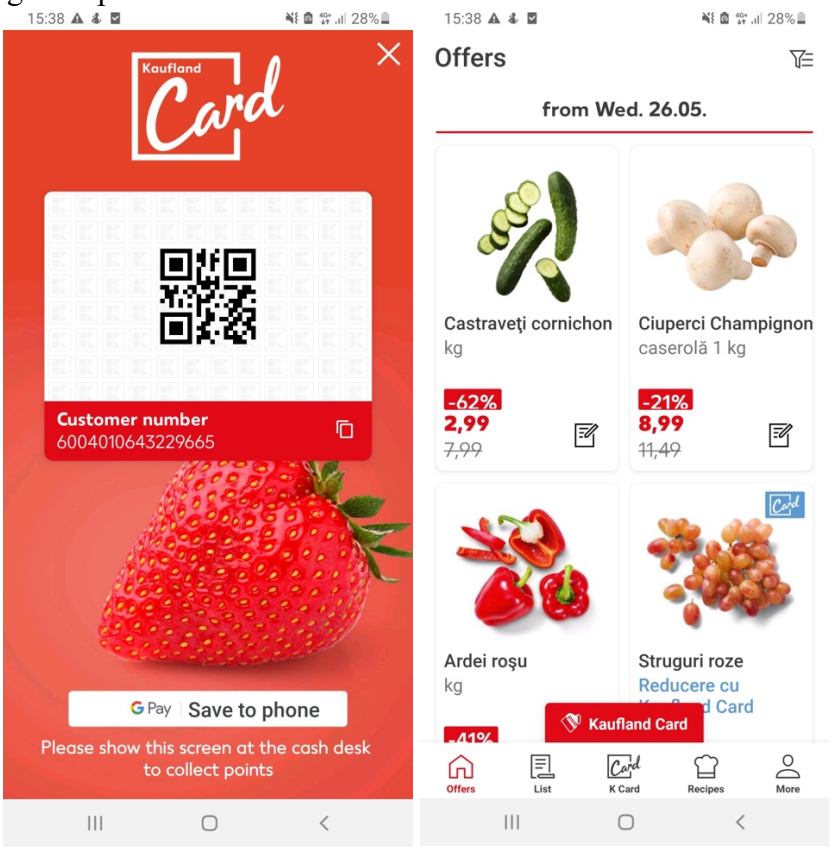
Many businesses have already made their own couponing app. This kind of application requires maintenance over a sustained period of time, funds to pay the programmers and to host the app. One of them is Kaufland. Their app allows the client to view details about their card (points, QR code) and current promotions and coupons. Many businesses follow this model, and it's hard to keep track of. Our application allows the users to view all of them in a single place.

This allows businesses to use this application as a platform to display their offers to customers without making their own application.

Existing solutions also include physical coupons. They are easy to lose, scratch, rip or tear. Nobody likes that.

Kaufland

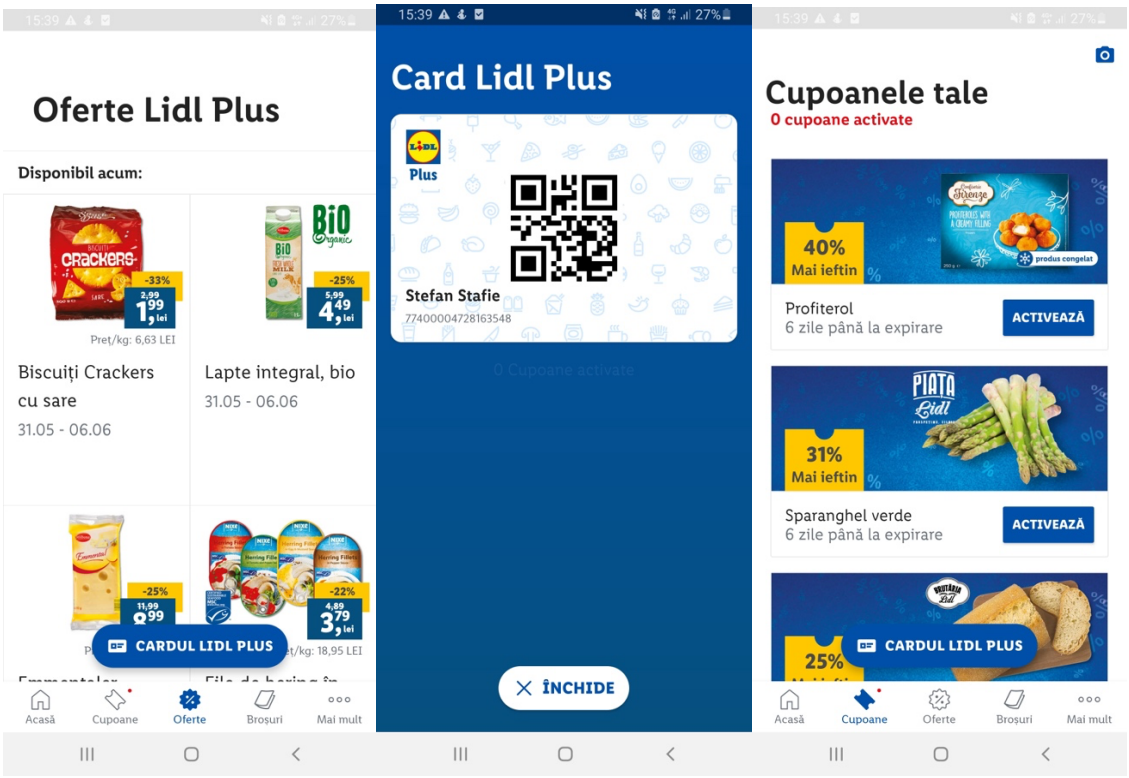
Platform: Android & iOS
Features: Allows the client to view details about their card (points, QR code) and current promotions and coupons.
Requires: Selecting a shop



Lidl Plus

Platform: Android & iOS
Features: Allows the client to view details about their card (points, QR code) and current promotions and coupons. Also, the posibility to see receipts online

Requires: Authentication (including personal data), selecting a shop (out of lidl shops).



Overview of technologies

Backend

Google cloud - we use this platform because it provides a reliable and highly scalable cloud computing services to its users. These services help us compute and store data, and help developers build, test, and deploy apps. Google cloud platform covers application, storage, and cloud computing services for backend, mobile, and web solutions using the internet.

MongoDB is an agile and scalable NoSQL database. MongoDB is based on the NoSQL document store model, in which data objects are stored as separate documents inside a collection instead of in the traditional columns and rows of a relational database. The documents are stored as binary JSON objects. The motivation of the MongoDB language is to implement a data store that provides high performance, high availability, and automatic scaling. MongoDB offers great website back-end storage for high-traffic websites that need to store data such as user credentials, high volume of discount documents, or other items because it is fast, scalable, and easy to implement.

Flask is a web framework for Python, which means it provides functionality to building web applications, including managing HTTP requests. We need this feature to enable communication with our application. There are several Flask extensions for integrating MongoDB and we'll be using the Flask-PyMongo extension.

Frontend

React is a library for building composable user interfaces. It encourages the creation of reusable UI components, which present data that changes over time. React can also render on the server using Node. We use this for our interface that the user interacts with.

Node Js is a JavaScript runtime environment which allows the infrastructure to build and run an application. It’s a light, scalable, and cross-platform way to execute code. It uses an event-driven I/O model which makes it extremely efficient and makes scalable network application possible.

Cloud services & API's

Google Maps API: Provides location capabilities and useful geographical information (geolocation, store addresses, user address), also it makes it easy to render maps and markers.

App engine - it is a service for building scalable applications. We use this to host our API for both frontend and backend of the app.

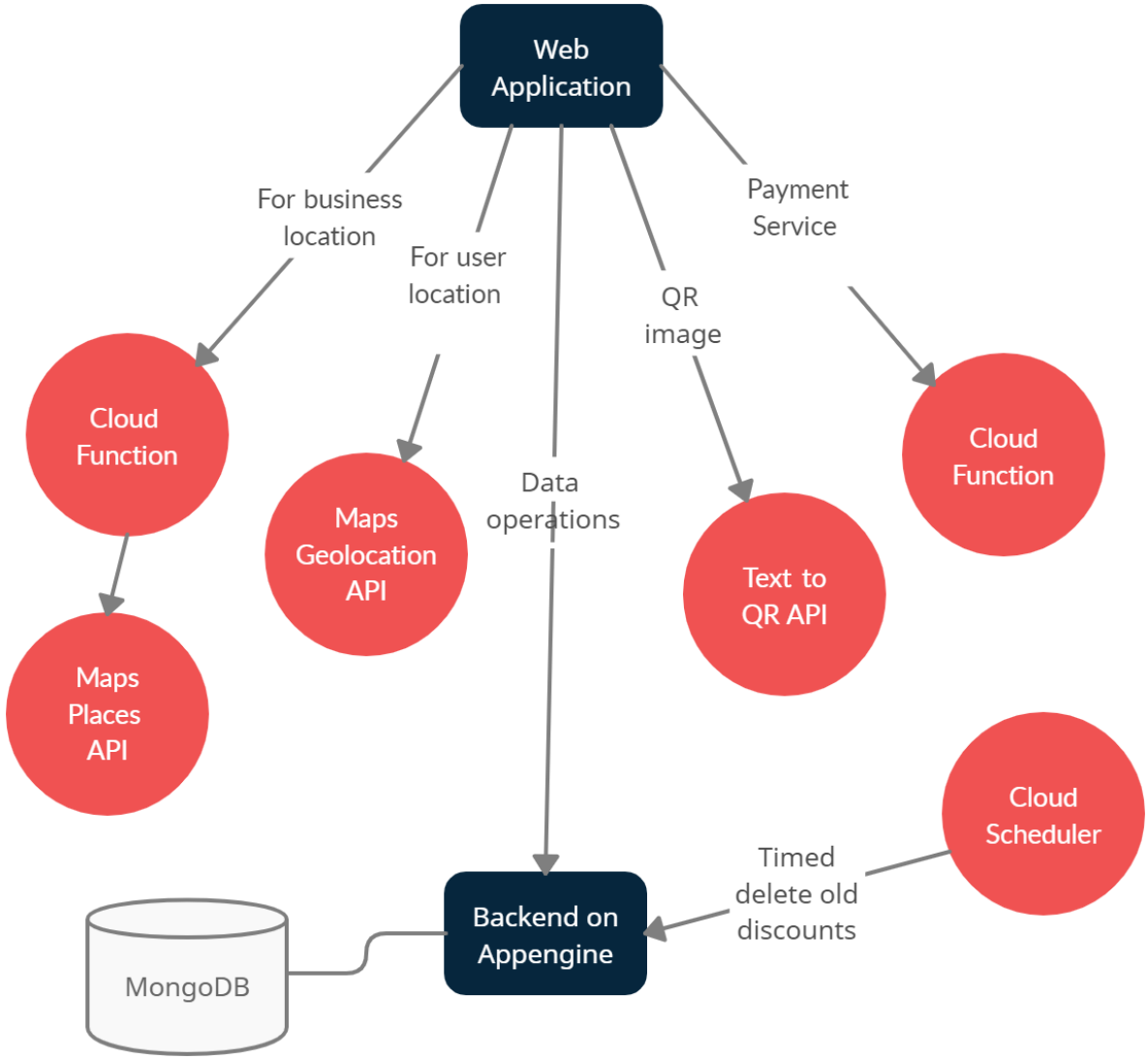
Maps Embed API & Maps JavaScript API - Used for rendering a map inside the site, along with all the useful commands a default google app has.

Maps Places API - Used for searching after a text input. We used it for finding the location of a business.

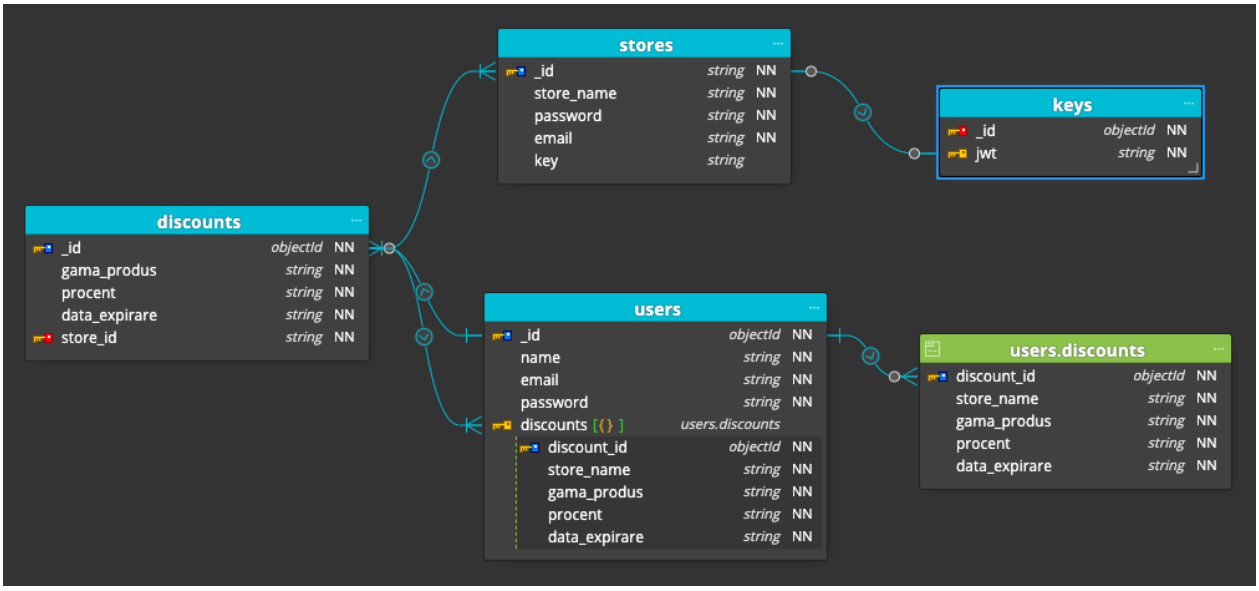
Maps Geolocation API - Used for finding the location of the user

Google Cloud Scheduler – Used for calling the *AutoDelete* API endpoint once a day, to remove all the discounts that expired.

Architecture



Database design



This is the database model that is used to store all the information that our app requires. It is composed of 4 collections, *discounts*, *users, stores, keys.* Credentials in both *users* and *stores* collections are encrypted using the *Bcrypt* library from Flask.

The *users* collection contains an array of discounts that belong to a store, which the user is subscribed to. The motivation behind this design was the easy access to all of the discounts that the user is subscribed to. In case of a discount being deleted or updated, users will have their discounts updated every time they log in.

The *discounts* collection will contain metadata of a specific discount, such as discount percentage, product range, the discount's expiration date and the unique id of the store that created the discount.

The *stores* collection contains the credentials and the API key they will have acquired, after which it will be stored in the *keys* collection.

Business canvas

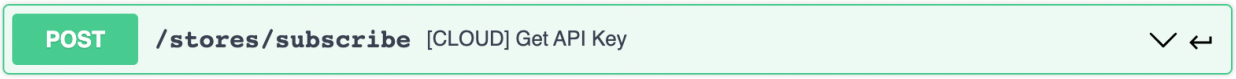
Functionality flow and app structure

Backend API

https://app.swaggerhub.com/apis/alexbarsan944/promo-app/1.0

User authentication required

Store authentication required



Appkey required

Delete key required