**Technical Documentation**

**Technologies Used**

* **Spring Boot** (Java framework for building REST APIs)
* **Google Maps API** (for fetching bank details)
* **RESTTemplate**
* **Postman** (testing)

**Components**

| **Component** | **Description** |
| --- | --- |
| BankController | Accepts API calls like /banks/nearby?zipcode=98390 |
| BankService | Contains logic to get coordinates and call MapsService |
| MapsService | Calls Google Maps API with the location and gets bank data |
| RestTemplate | Makes HTTP GET requests to the external API |
| application.yml | Stores API key and configurations like the port numbers |

**Data Flow**

1. **User sends request** with zipcode to BankController
2. BankController calls the bankService’s getNearbyBanks
3. **BankService** calls maps.service.url (maps/banksIn10Miles) and appends the zipcode
4. The maps/banksIn10Miles inturn calls the mapsContoller
5. The mapsContoller calls mapsService.findNearbyBanks(zipcode);
6. MapsService hits **Google geolocationAPI** to get latitude and longitude from zipcode
7. Using lattitue and logiture and 10 miles radius (converted to meters) and **Googles Place nearbysearch API** fetches nearby banks.
8. Response is parsed and returned to user as JSON.

**Input:**

**Zipcode**

**http://localhost:8080/banks/nearby?zipcode=98390**

**Output :**

["Umpqua Bank","Chase Bank","Umpqua Bank","Umpqua Bank","Umpqua Bank","1st Security Bank","Umpqua Bank","Bank of America (with Drive-thru ATM)","KeyBank","Timberland Bank","Umpqua Bank","U.S. Bank Branch","U.S. Bank Branch","Wells Fargo Bank","KeyBank","Banner Bank","Wells Fargo Bank","Umpqua Bank","Umpqua Bank","KeyBank"]

**Response**

json

[

{

"name": "Chase Bank",

"address": "1100 North Meridian Avenue, Puyallup",

"latitude": 47.2008661,

"longitude": -122.2960788

},

...

]