**System requirements**

Functional requirements**:**

* Usershould be able to store/update their marketing preferences (sms/email/post)
* Usershould be able retrieve his/her marketing preferences (sms/email/post)

**Use case diagram**

Actors of system:

* **User** can store, update and retrieve his/her marketing preferences

Use cases:

* **Add** marketing preference
* **Update** marketing preference
* **Retrieve** marketing preferences

Diagram

Description automatically generated

Class diagram:

* **MarketingPreference** Keeps whole data structure
* **IdentityData-**  Any information that allows individual to be identified uniquely (Name , address, phonNumber, ..)
* **DescriptiveData-** Any additional information about a customer that goes beyond identity to better understand their habits (ex: Family details, lifestyle details, career details, ..)
* **BehavioralData** General patterns to understand how customers feel about product and services
* **PreferenceType** The way that they want to be informed by system (email, SMS , post)

Text

Description automatically generated

We can also create activity , sequence diagram for detailed explanation , just lack of time.

**API design**

**1.Create preference**

@PostMapping("/preference/create")  
public ResponseEntity<MarketingPreference> createMarketingPreference(@RequestBody MarketingPreferenceDTO preferenceDTO)

**2.Edit preference**

@PutMapping("/preference/edit")  
public ResponseEntity<MarketingPreference> editMarketingPreference(@RequestBody MarketingPreferenceDTO preferenceDTO,@RequestParam String preferenceId)

**3.Fetch preferences**

@GetMapping("/preferences")  
public ResponseEntity<Page<MarketingPreference>> retrievePreferences(Pageable pageable)

**API documentation :** [**http://localhost:9090/swagger-ui.html#/**](http://localhost:9090/swagger-ui.html#/)

Postman collection can be found in **documentation** directory

**Application Architecture**

**Diagram

Description automatically generated**