Auth Service have 2 databases: mongo and redis

//mongo structure

type Credentials struct {

ID primitive.ObjectID `bson:"\_id" json:"id"`

Username string `bson:"username" json:"username"`

Password string `bson:"password" json:"password"`

UserType UserType `bson:"userType" json:"userType"`

Verified bool `bson:"verified" json:"verified"`

}

//redis structure

key string,

value string

For key we store username,userId or userToken, and for value we generate new UUID for token value for verification of account.

User service use mongo as database

//mongo structure

type User struct {

ID primitive.ObjectID `bson:"\_id" json:"id"`

Firstname string `bson:"firstName,omitempty" json:"firstName,omitempty"`

Lastname string `bson:"lastName,omitempty" json:"lastName,omitempty"`

Gender Gender `bson:"gender,omitempty" json:"gender,omitempty"`

Age int `bson:"age,omitempty" json:"age,omitempty"`

Residence string `bson:"residence,omitempty" json:"residence,omitempty"`

Email string `bson:"email" json:"email"`

Username string `bson:"username" json:"username"`

UserType UserType `bson:"userType" json:"userType"`

}

Accommodation service use mongo as database

//mongo structure

type Accommodation struct {

ID primitive.ObjectID `bson:"\_id,omitempty" json:"id"`

Name string `bson:"name,omitempty" json:"name"`

Description string `bson:"description,omitempty" json:"description"`

Images string `bson:"images,omitempty" json:"images"`

Location Location `bson:"location,omitempty" json:"location"`

Benefits string `bson:"benefits,omitempty" json:"benefits"`

MinGuest int `bson:"minGuest,omitempty" json:"minGuest"`

MaxGuest int `bson:"maxGuest,omitempty" json:"maxGuest"`

OwnerId string `bson:"ownerId,omitempty" json:"ownerId"`

}

type Location struct {

Country string `bson:"country,omitempty" json:"country"`

City string `bson:"city,omitempty" json:"city"`

Street string `bson:"street,omitempty" json:"street"`

Number int `bson:"number,omitempty" json:"number"`

}

type Rate struct {

ID gocql.UUID `json:"id" db:"reservation\_id"`

ByGuestId string `json:"byGuestId" db:"by\_guestId"`

ForHostId string `json:"forHostId" db:"for\_hostId"`

ForAccommodationId string `json:"forAccommodationId" db:"for\_accommodationId"`

CreatedAt time.Time `json:"createdAt" db:"created\_at"`

UpdatedAt time.Time `json:"updatedAt" db:"updated\_at"` }

Notification service use mongo database

// mongo structure

Type Notification struct {

ID gocql.UUID `json:"id" db:"reservation\_id"`

ByGuestId string `json:"byGuestId" db:"by\_guestId"`

ForHostId string `json:"forHostId" db:"for\_hostId"`

Description string `bson:"description,omitempty" json:"description"`

CreatedAt time.Time `json:"createdAt" db:"created\_at"`

}

Reservation service combine mongo and cassandra databases

// mongo

type Appointment struct {

ID primitive.ObjectID `bson:"\_id,omitempty" json:"id"`

Available []time.Time `bson:"available,omitempty" json:"available"`

AccommodationId string `bson:"accommodationId,omitempty" json:"accommodationId"`

PricePerGuest int `bson:"pricePerGuest,omitempty" json:"pricePerGuest"`

PricePerAccommodation int `bson:"pricePerAccommodation,omitempty" json:"pricePerAccommodation"`

Rates []Rate `bson:"rates,omitempty" json:"rates"`

}

// cassandra

type Reservation struct {

ID gocql.UUID `json:"id" db:"reservation\_id"`

Period []time.Time `json:"period" db:"periodd"`

ByUserId string `json:"byUserId" db:"by\_userId"`

AccommodationId string `json:"accommodationId" db:"accommodation\_id"`

Price int `json:"price" db:"price"` }

Reccommondation service use neo4j database

// User Node model

type UserNode struct {

ID string `json:"id"`

FirstName string `json:"firstName"`

LastName string `json:"lastName"`

Email string `json:"email"`

Username string `json:"username"`

UserType string `json:"userType"`

}

// Accommodation Node model

type AccommodationNode struct {

ID string `json:"id"`

Name string `json:"name"`

Description string `json:"description"`

Images string `json:"images"`

Ratings []RatingNode `json:"ratings"`

}

// Rating Node model

type RatingNode struct {

ID string `json:"id"`

Value int `json:"value"`

DateTime time.Time`json:"date"`

ByUserId string `json:"byUserId"`

}

// Reservation Node model

type ReservationNode struct {

ID string `json:"id"`

Period string `json:"period"`

ByUserId string `json:"byUserId"`

AccommodationId string `json:"accommodationId"`

Price int `json:"price"`

}

// Recommendation Relationship model

type RecommendationRelationship struct {

RecommendedAccommodationID string `json:"recommendedAccommodationId"`

}