System Design Document

Friday October 8, 2021

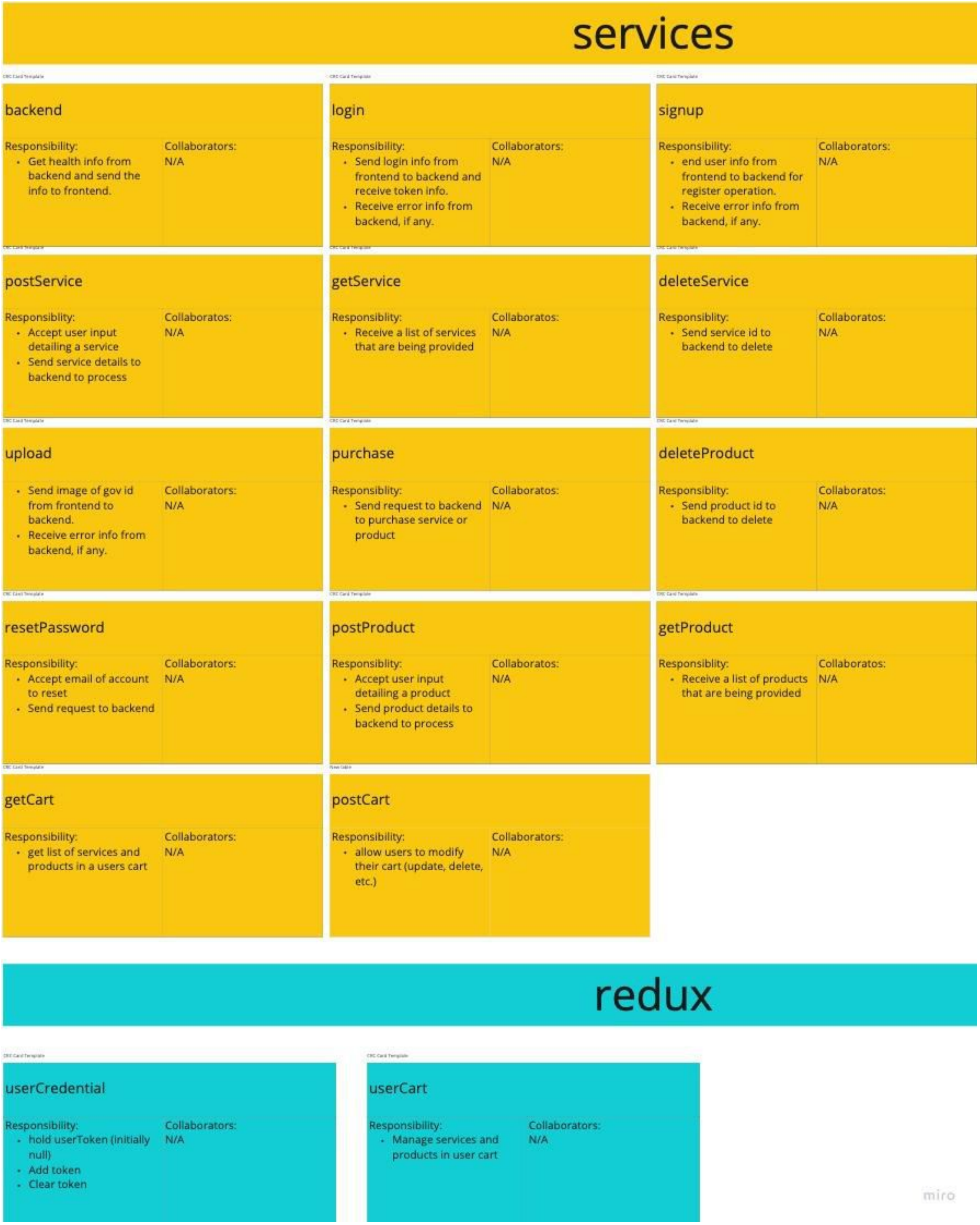
Sprint 1

Krutik Tejalkumar Shah
Mark (LiangShu) Chen
Steven Hans Limantoro
Tanzim Ahmed
Tony Chen
Pasa Aslan
Payam Yektamaram

Table of Contents

CRC Cards

Frontend



services

backend

Responsibility:
- Get health info from backend and send the info to frontend.

Collaborators:
N/A

login

Responsibility:
- Send login info from frontend to backend and receive token info.
- Receive error info from backend, if any.

Collaborators:
N/A

signup

Responsibility:
- end user info from frontend to backend for register operation.
- Receive error info from backend, if any.

Collaborators:
N/A

postService

Responsiblity:
- Accept user input detailing a service
- Send service details to backend to process

Collaboratos:
N/A

getService

Responsiblity:
- Receive a list of services that are being provided

Collaboratos:
N/A

deleteService

Responsiblity:
- Send service id to backend to delete

Collaboratos:
N/A

upload

- Send image of gov id from frontend to backend.
- Receive error info from backend, if any.

Collaborators:
N/A

purchase

Responsiblity:
- Send request to backend to purchase service or product

Collaboratos:
N/A

deleteProduct

Responsiblity:
- Send product id to backend to delete

Collaboratos:
N/A

resetPassword

Responsibility:
- Accept email of account to reset
- Send request to backend

Collaborators:
N/A

postProduct

Responsibility:
- Accept user input detailing a product
- Send product details to backend to process

Collaboratos:
N/A

getProduct

Responsiblity:
- Receive a list of products that are being provided

Collaboratos:
N/A

getCart

Responsibility:
- get list of services and products in a users cart

Collaborators:
N/A

postCart

Responsibility:
- allow users to modify their cart (update, delete, etc.)

Collaborators:
N/A

redux

userCredential

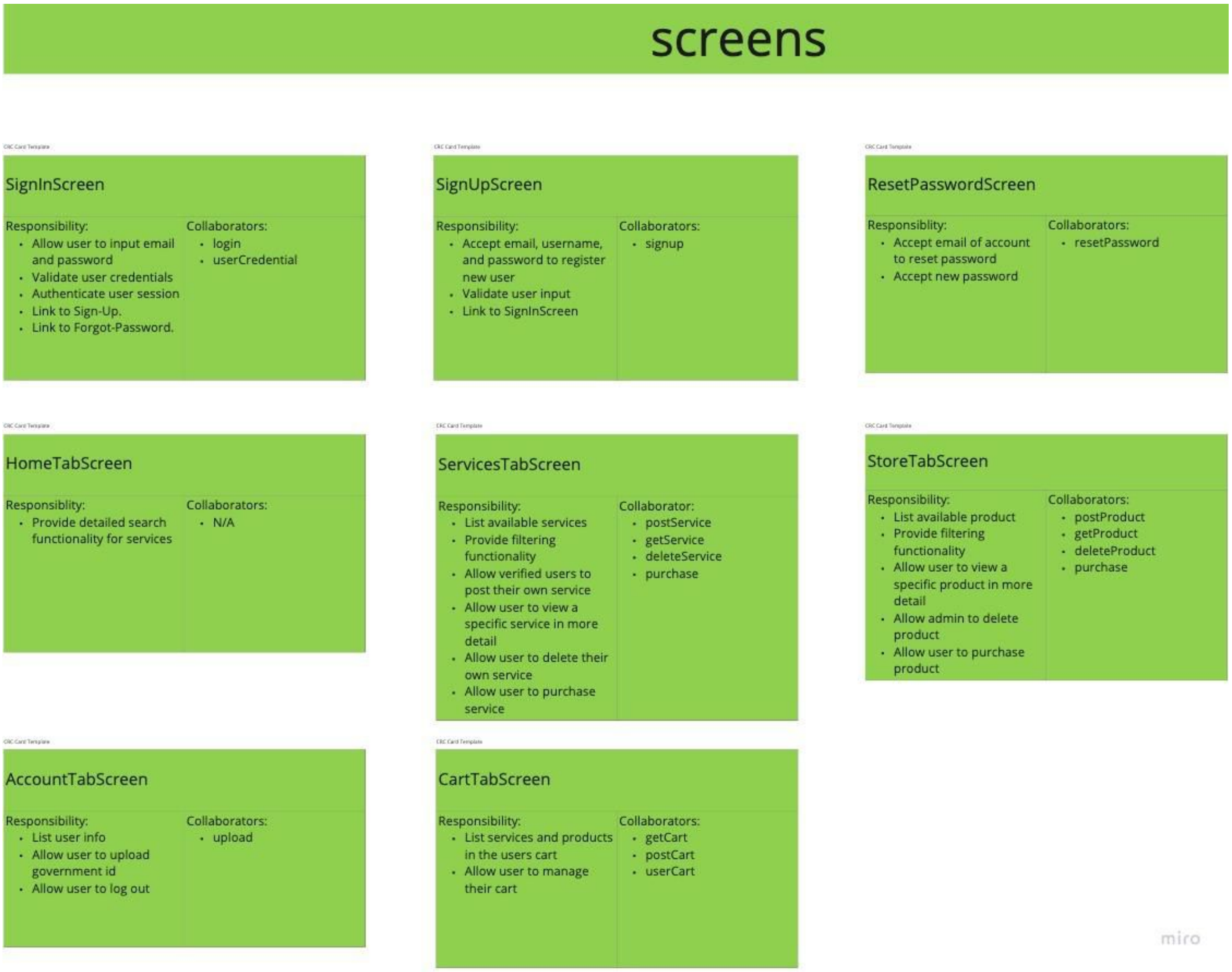
Responsibility:
- hold userToken (initially null)
- Add token
- Clear token

Collaborators:
N/A

userCart

Responsibility:
- Manage services and products in user cart

Collaborators:
N/A



screens

CRC Card Template

SignInScreen

Responsibility:
- Allow user to input email and password
- Validate user credentials
- Authenticate user session
- Link to Sign-Up.
- Link to Forgot-Password.

Collaborators:
- login
- userCredential

CRC Card Template

SignUpScreen

Responsibility:
- Accept email, username, and password to register new user
- Validate user input
- Link to SignInScreen

Collaborators:
- signup

CRC Card Template

ResetPasswordScreen

Responsiblity:
- Accept email of account to reset password
- Accept new password

Collaborators:
- resetPassword

CRC Card Template

HomeTabScreen

Responsiblity:
- Provide detailed search functionality for services

Collaborators:
- N/A

CRC Card Template

ServicesTabScreen

Responsibility:
- List available services
- Provide filtering functionality
- Allow verified users to post their own service
- Allow user to view a specific service in more detail
- Allow user to delete their own service
- Allow user to purchase service

Collaborator:
- postService
- getService
- deleteService
- purchase

CRC Card Template

StoreTabScreen

Responsibility:
- List available product
- Provide filtering functionality
- Allow user to view a specific product in more detail
- Allow admin to delete product
- Allow user to purchase product

Collaborators:
- postProduct
- getProduct
- deleteProduct
- purchase

CRC Card Template

AccountTabScreen

Responsibility:
- List user info
- Allow user to upload government id
- Allow user to log out

Collaborators:
- upload

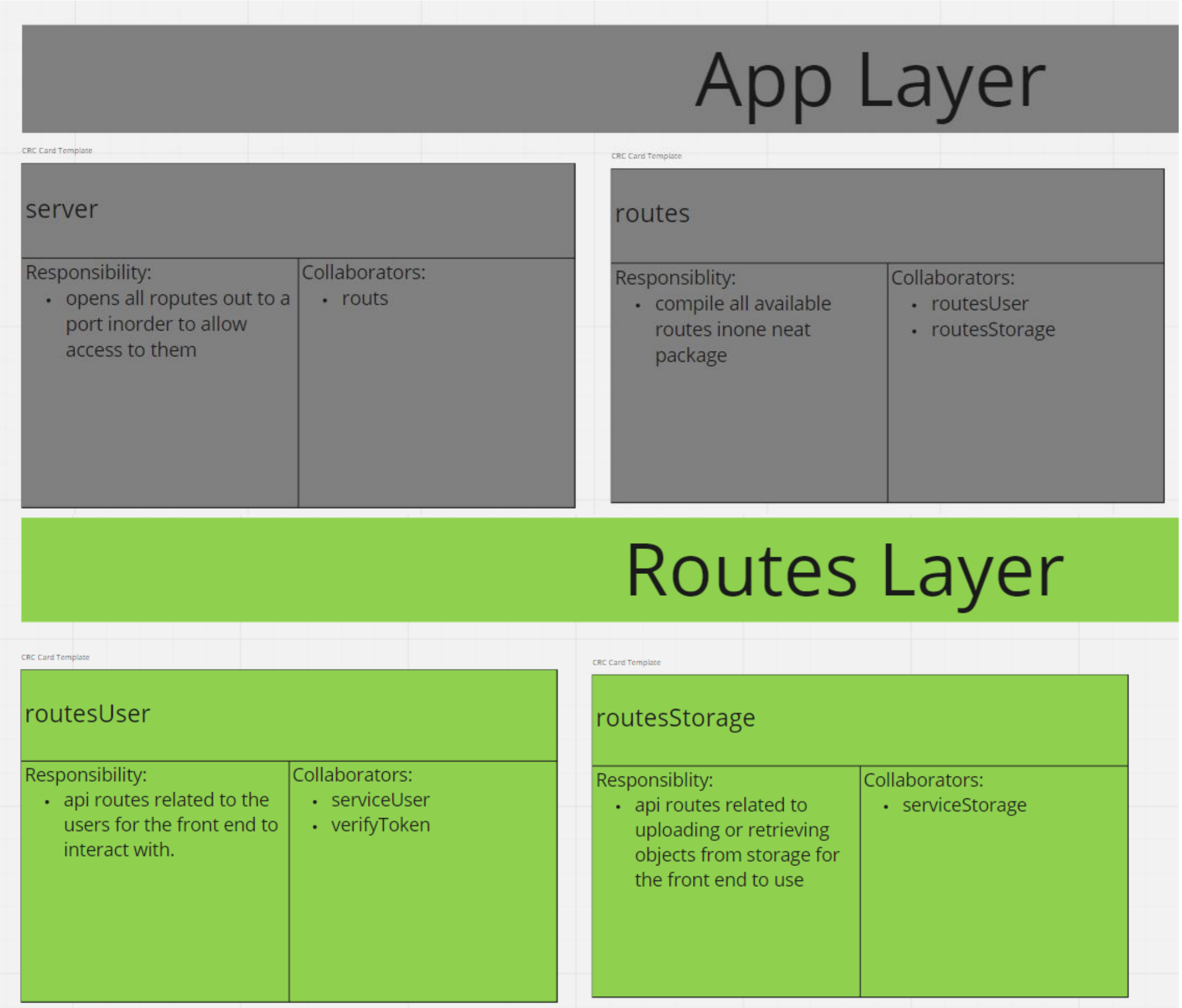
CRC Card Template

CartTabScreen

Responsibility:
- List services and products in the users cart
- Allow user to manage their cart

Collaborators:
- getCart
- postCart
- userCart

Backend



App Layer

CRC Card Template

server

Responsibility:	Collaborators:
• opens all roputes out to a port inorder to allow access to them	• routs

CRC Card Template

routes

Responsiblity:	Collaborators:
• compile all available routes inone neat package	• routesUser • routesStorage

Routes Layer

CRC Card Template

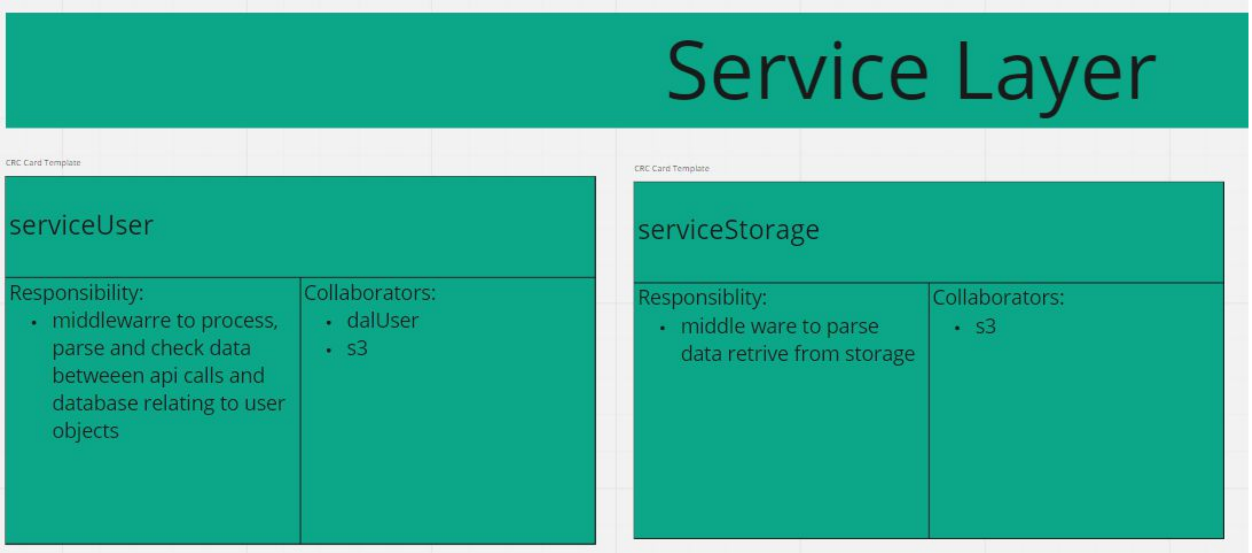
routesUser

Responsibility:	Collaborators:
• api routes related to the users for the front end to interact with.	• serviceUser • verifyToken

CRC Card Template

routesStorage

Responsiblity:	Collaborators:
• api routes related to uploading or retrieving objects from storage for the front end to use	• serviceStorage



Service Layer

CRC Card Template

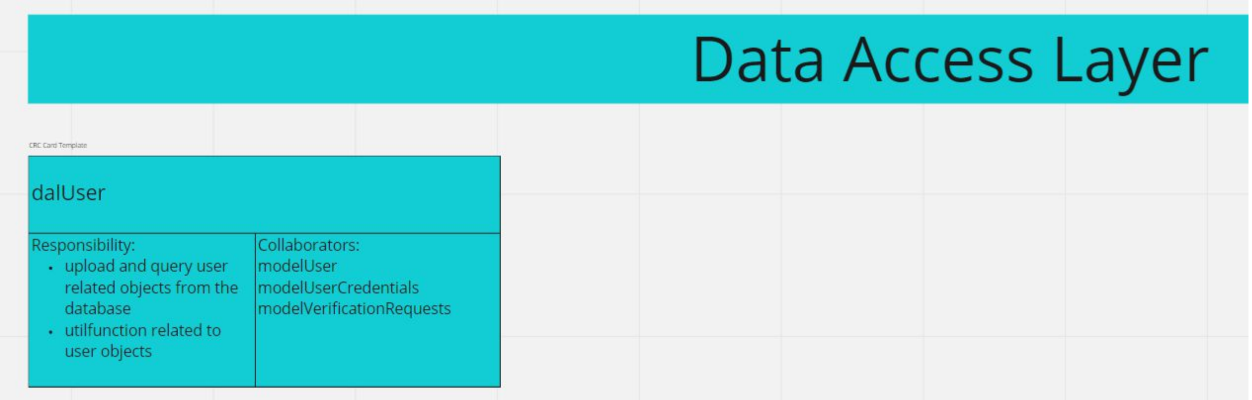
serviceUser

Responsibility:	Collaborators:
• middlewarre to process, parse and check data betweeen api calls and database relating to user objects	• dalUser • s3

CRC Card Template

serviceStorage

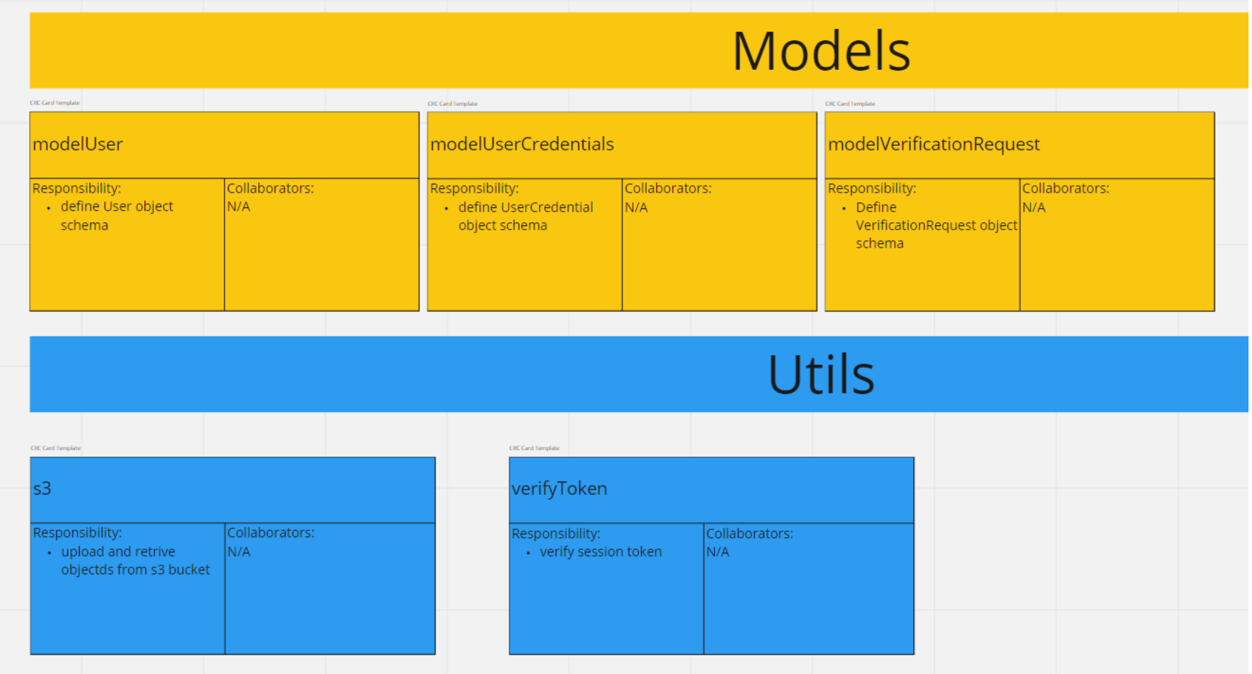
Responsiblity:	Collaborators:
• middle ware to parse data retrive from storage	• s3



Data Access Layer

CRC Card Template

dalUser	
Responsibility: • upload and query user related objects from the database • utilfunction related to user objects	Collaborators: modelUser modelUserCredentials modelVerificationRequests



Models

CRC Card Template

modelUser	
Responsibility: • define User object schema	Collaborators: N/A

CRC Card Template

modelUserCredentials	
Responsibility: • define UserCredential object schema	Collaborators: N/A

CRC Card Template

modelVerificationRequest	
Responsibility: • Define VerificationRequest object schema	Collaborators: N/A

Utils

CRC Card Template

s3	
Responsibility: • upload and retrive objectds from s3 bucket	Collaborators: N/A

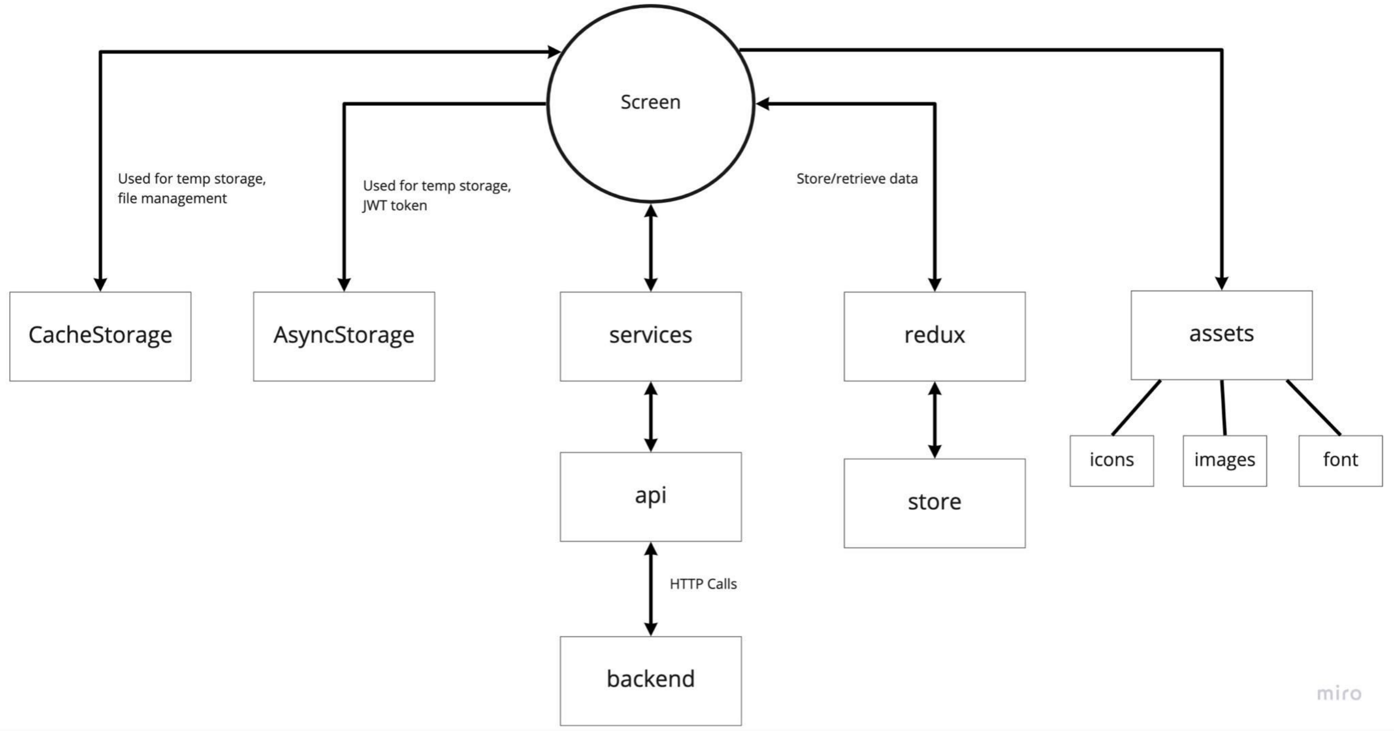
CRC Card Template

verifyToken	
Responsibility: • verify session token	Collaborators: N/A

System Architecture Diagram (*Node.js Layered Arch*)

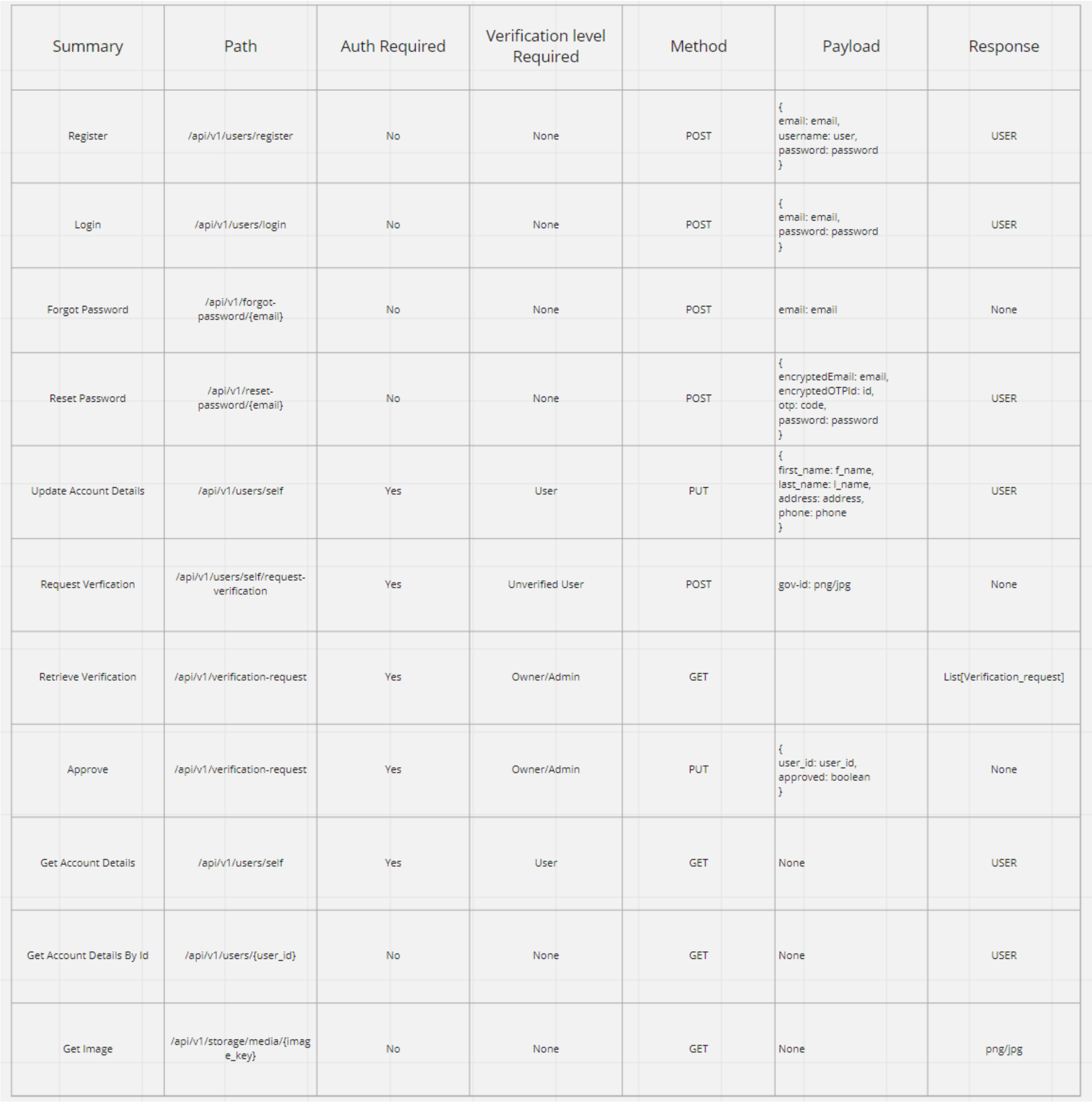
Frontend

Architecture



Backend

Server Endpoints & Descriptions



Summary	Path	Auth Required	Verification level Required	Method	Payload	Response

Register	/api/v1/users/register	No	None	POST	{
					email: email,
					username: user,
					password: password
					}
						USER

Login	/api/v1/users/login	No	None	POST	{
					email: email,
					password: password
					}
						USER

Forgot Password	/api/v1/forgot-password/{email}	No	None	POST	email: email	None

Reset Password	/api/v1/reset-password/{email}	No	None	POST	{
					encryptedEmail: email,
					encryptedOTPId: id,
					otp: code,
					password: password
					}
						USER

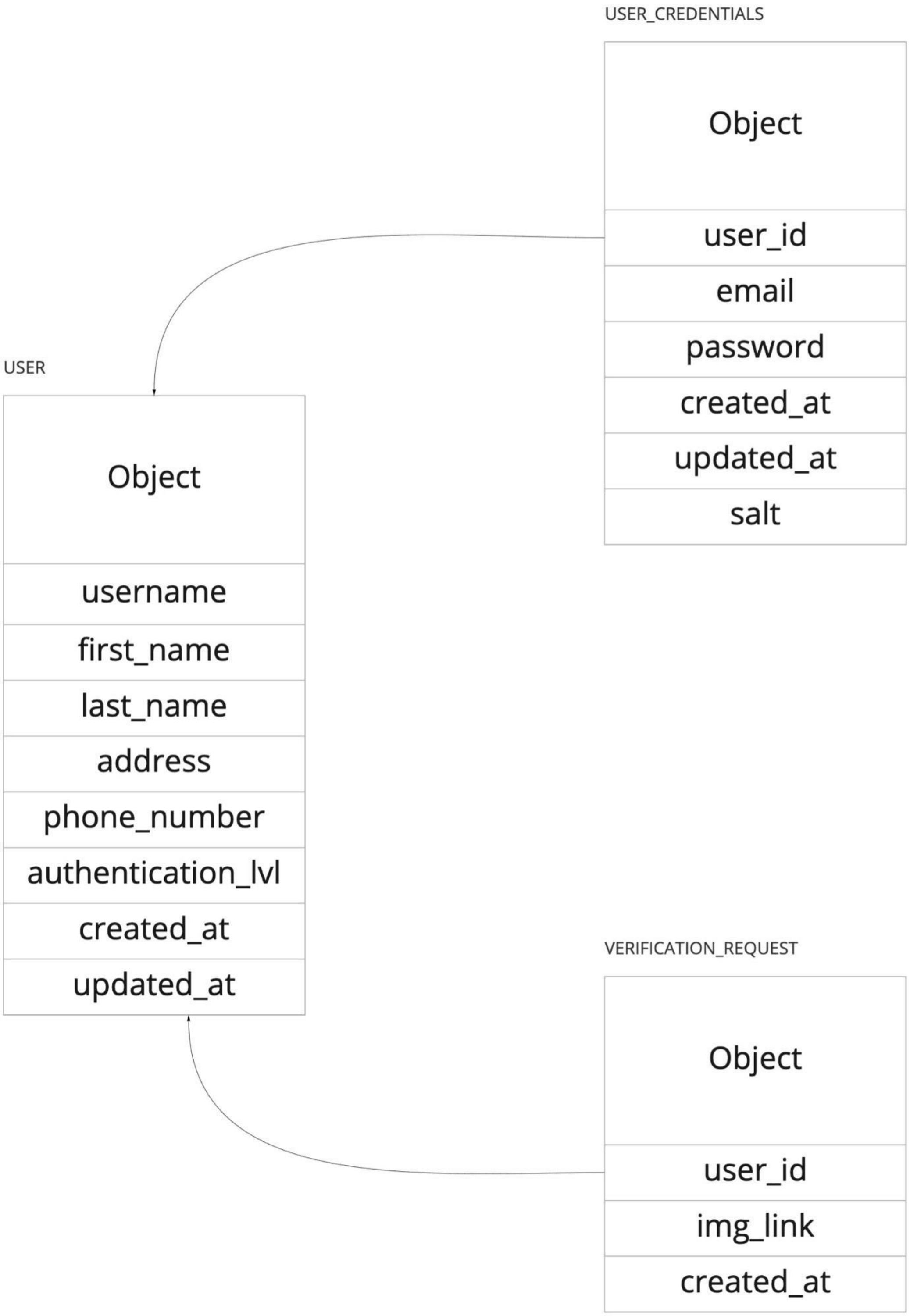
Update Account Details	/api/v1/users/self	Yes	User	PUT	{
					first_name: f_name,
					last_name: l_name,
					address: address,
					phone: phone
					}
						USER

Request Verfication	/api/v1/users/self/request-verification	Yes	Unverified User	POST	gov-id: png/jpg	None
Retrieve Verification	/api/v1/verification-request	Yes	Owner/Admin	GET		List[Verification_request]

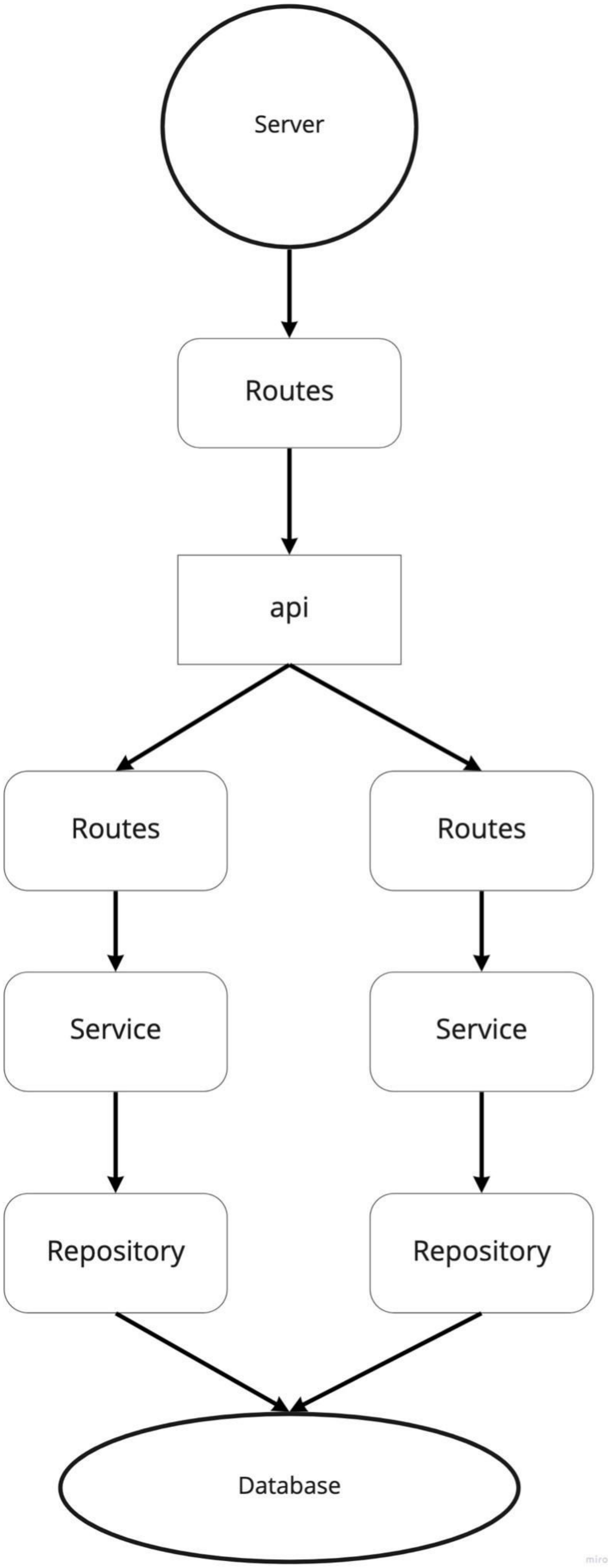
Approve	/api/v1/verification-request	Yes	Owner/Admin	PUT	{
					user_id: user_id,
					approved: boolean
					}
						None

Get Account Details	/api/v1/users/self	Yes	User	GET	None	USER
Get Account Details By Id	/api/v1/users/{user_id}	No	None	GET	None	USER
Get Image	/api/v1/storage/media/{image_key}	No	None	GET	None	png/jpg

Database Schema Models



Architecture


Server
Routes
api
Routes
Routes
Service
Service
Repository
Repository
Database
miro