

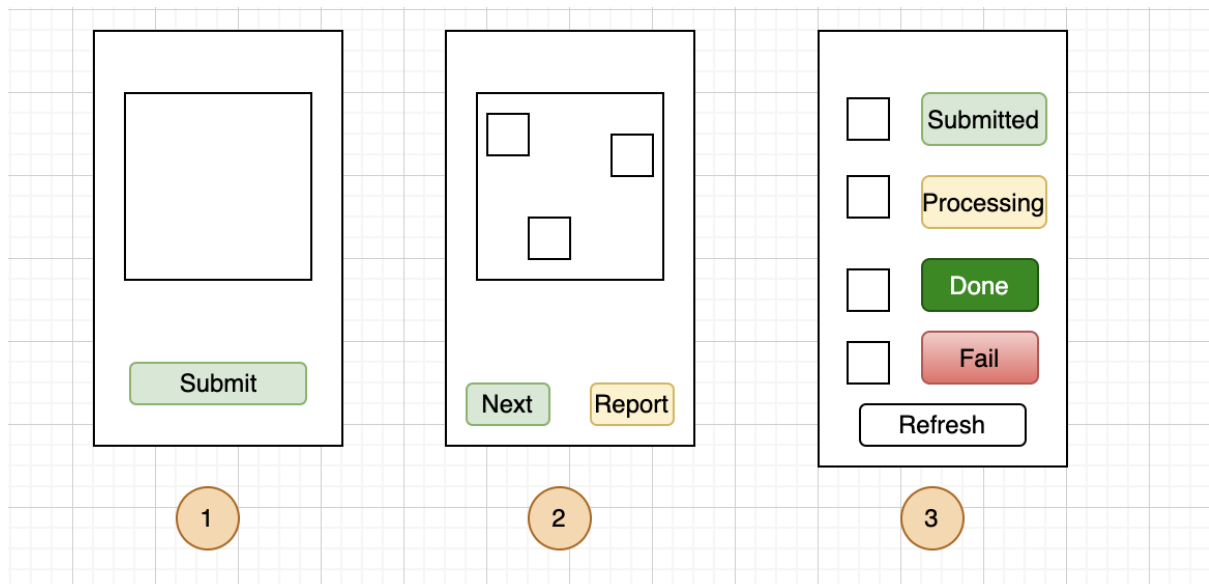
Summary

Design an image recognition app (using camera to recognize the objects - table, dog, cat, human, etc).

Assumption

System is designed for 1000 concurrent users and 100 req/s upload images.

UI/UX

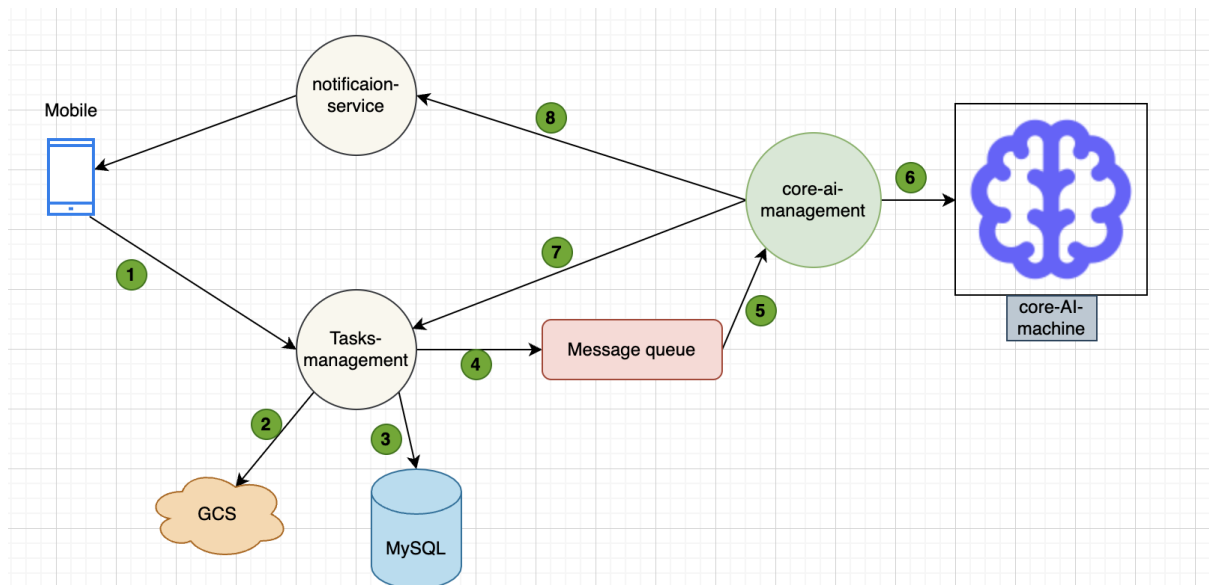


Screen 1: Users use camera and take a picture, after that click button submit

Screen 2: Users view result a task, users can report bug or loop by click next

Screen 3: User can list their tasks and see status

System design architecture



1: Users submit images to task-management services. Users also call API to list task and task_status

2. Upload image to Google cloud storage
3. Create a task, save to db, response task_id to mobile
4. Push task information to queue, after that close context of request
5. Core AI management service receive a task
6. Core AI management service get image from GCS, integrate with core-AI machine and wait response
7. After has response, Core AI management service send request to update task status
8. Core AI management service send notification to user

Why do i chose?

MySQL: Task model is object with specific field, we don't need change the number of field usually, so i think SQL is good. MySQL has phpMyAdmin, easy to use and free.

Google cloud storage: 200 Gbps per project, \$0.020/GB/month, if we get average 5 req/s and 1 image size 3M. We should pay $5 \times 3600 \times 24 \times 30 \times 0.003 \times 0.02 = 777.6\$$ per month

Message queue: Google PubSub, i am so familiar with Google ecosystem so i chose that. I used to use kafka but i don't like that.

Bottleneck

The bottleneck is (6), core AI machine. We get 100 req upload image per second, never bottleneck at GCS. With MySQL we can use Cloud SQL for MySQL for scalability.

Database design

profiles

Field	Data type	Comment
id	INT(20)	id of user
email	VARCHAR(128)	email of user, index
phone	VARCHAR(16)	phone number of user
name	VARCHAR(255)	user name
meta_data	LONGTEXT	store everything
created_at	datetime	
updated_at	datetime	
KEY	`idx_email`(`email`)	key for optimizer filter

tasks:

Field	Data type	Comment
id	INT(20)	id of task
user_id	INT(20)	this task belong to
status	INT(10)	enum submitted, processing, succeed, fail
retry_cnt	INT(10)	retry times
image	VARCHAR(255)	image url in GCS
meta_data	LONGTEXT	store everything in json object
created_at	datetime	
updated_at	datetime	
deleted_at	datetime	
KEY	`idx_user_id`(`user_id`)	key for optimizer filter
KEY	idx_status(`status`)	key for optimizer filter

notification:

Field	Data type	Comment
id	INT(20)	id of noti
noti_type	INT(100)	use for mobile, each type has each format
deep_link	VARCHAR(255)	what happens when user click
target	VARCHAR(64)	phone or email of user
title	VARCHAR(255)	title of noti
data	LONGTEXT	noti data, like image, message, format, text color
status	INT(100)	enum (fail, sent, received, opend)
meta_data	LONGTEXT	store everything
created_at	datetime	
updated_at	datetime	
opened_at	datetime	user open noti, callback from webhook

KEY	<code>`idx_target`(`target`)</code>	key for optimizer filter
-----	-------------------------------------	--------------------------