Few Assumptions:

1. For now I have assumed 5 questions and 5 answers.
2. For the Market Research Agency use case I have assumed the question will be as follows: “Name”, “Age”, “Job”, “Income”, “Savings”

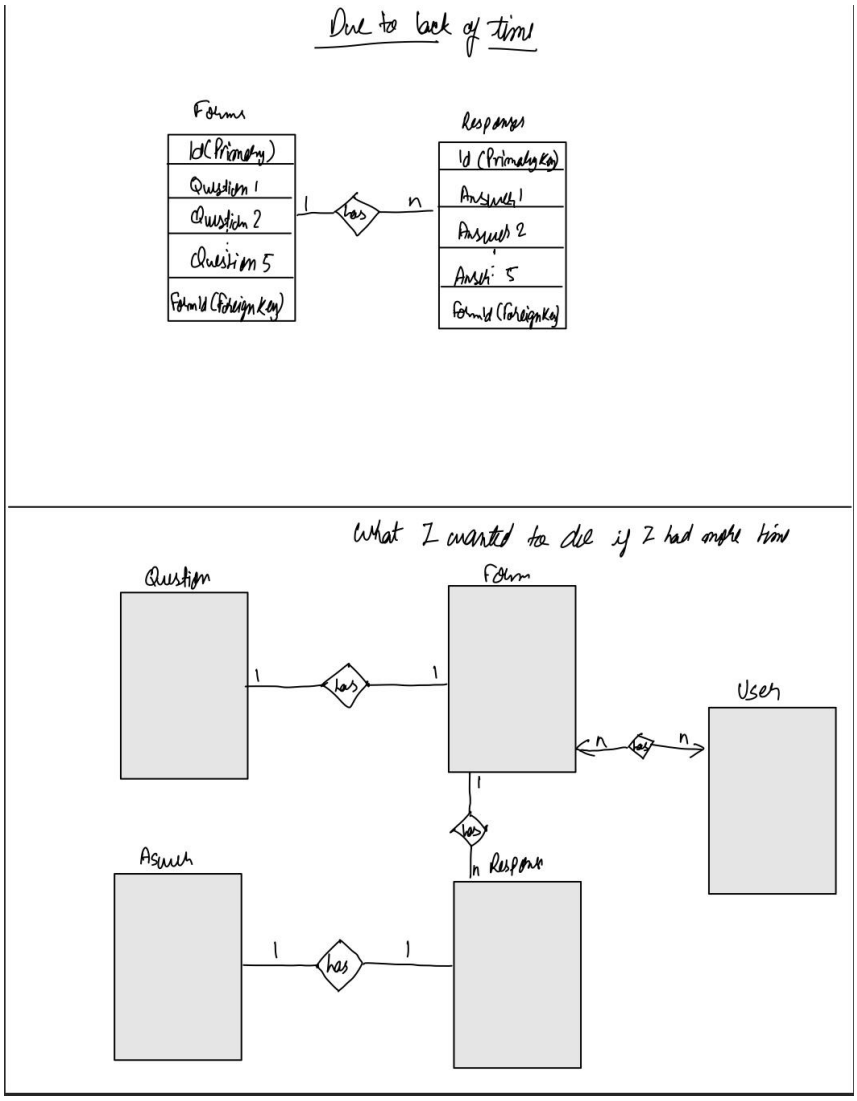
Google Sheets API Limitation: Sheets API has per-minute quotas, and they're refilled every minute. For example, there's a read request limit of 300 per minute per project. If your app sends 350 requests in one minute, the additional 50 requests exceed the quota and generates a 429: Too many requests HTTP status code response.

Solutions: The solution is very simple, while exporting the data to google sheets, this limitation can be handled by giving the endpoint a wait time of 10 seconds when reaching 300 hits. We can define a function such that if a 429 (Too Many Requests) error occurs, the function waits for the specified duration (extracted from the retry-after header) and then retries the API call. This should help you stay within the rate-limiting constraints of the Google Sheets API.

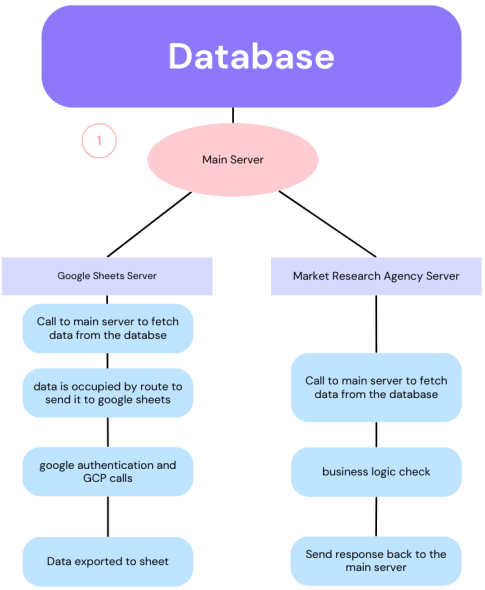
Database Choice: The database that I chose to go with was MySQL, although I use MongoDB but since it is a data platform and there has to be a relationship between the models which is well offered by Sequel DBs. Also Sequel DBs can be made more failsafe as compared to NoSQL as they have features such as transactions, triggers and SQL Database are very compatible to handle long data as they provide faster search, better indexing and various operations over it.

- ➢ Introduction
    - The goal of this project is to create a data collection platform which can be used by customers to power their most critical activities. There are various use cases available which should be implemented in a plug and play fashion.
- ➢ System Architecture
    - The technology stack that I have used is SERN which is Sql for database, ExpressJs and NodeJs for backend.
    - The architecture that I am following is Microservice architecture which allows me to achieve the plug and play goal, for more details refer to Project Approach pdf.
    - The architecture consists of three servers, main server, Google Sheets server and Market Research Agency Server.
- ➢ Database Design
    - Right now for the database I have two models: form and responses where the relationship between them is of one to many. As I was short on time I have included the question in the schema for form itself and the answer in the response schema. Also for now I have assumed and limited the number of questions to five, this goes for the answers as well.
    - If I had more time at my disposal I would have designed the schema with four models: Form, Questions, Responses and Answers where the relationship between Form-Question would be one-to-one as each form would correspond to one particular table of questions. The relationship between Form-Responses

would still remain one-to-many as each form can have multiple responses. The relationship between responses-answer would be one-to-one as each response would be associated with a particular answer table given by a user.



- Module/Component Design
    - The modules/component design is such that each server has an app.js, routes and controllers. The main server has an additional database to it since its primary task is to communicate with the database.
    - Once again pertaining to the limited time that I had I would have preferred to have a middleware to the main server to handle authentication and authorization which yes, also means that I wanted to have different users on the data platform as well.
- Data Flow Diagram



4

➢ Backend Design