

A decorative graphic on the left side of the slide, consisting of a network of white lines and small circles on a blue gradient background, resembling a circuit board or a neural network.

DEBT MANAGEMENT SERVICE



OUTLINE:

- Problem Statement
 - Challenges
 - Technologies
 - Diagrams and Explanations
 - Future goals
 - Demo
 - Q & A
- 
- 
- 

PROBLEM STATEMENT:

- Design and implement a service (microservice)
 - Be able to receive a rest service to receive user account records and be able to output to reporting system.
 - Be able to process via File from FTP
 - Be able to output record and upload to SFTP server.



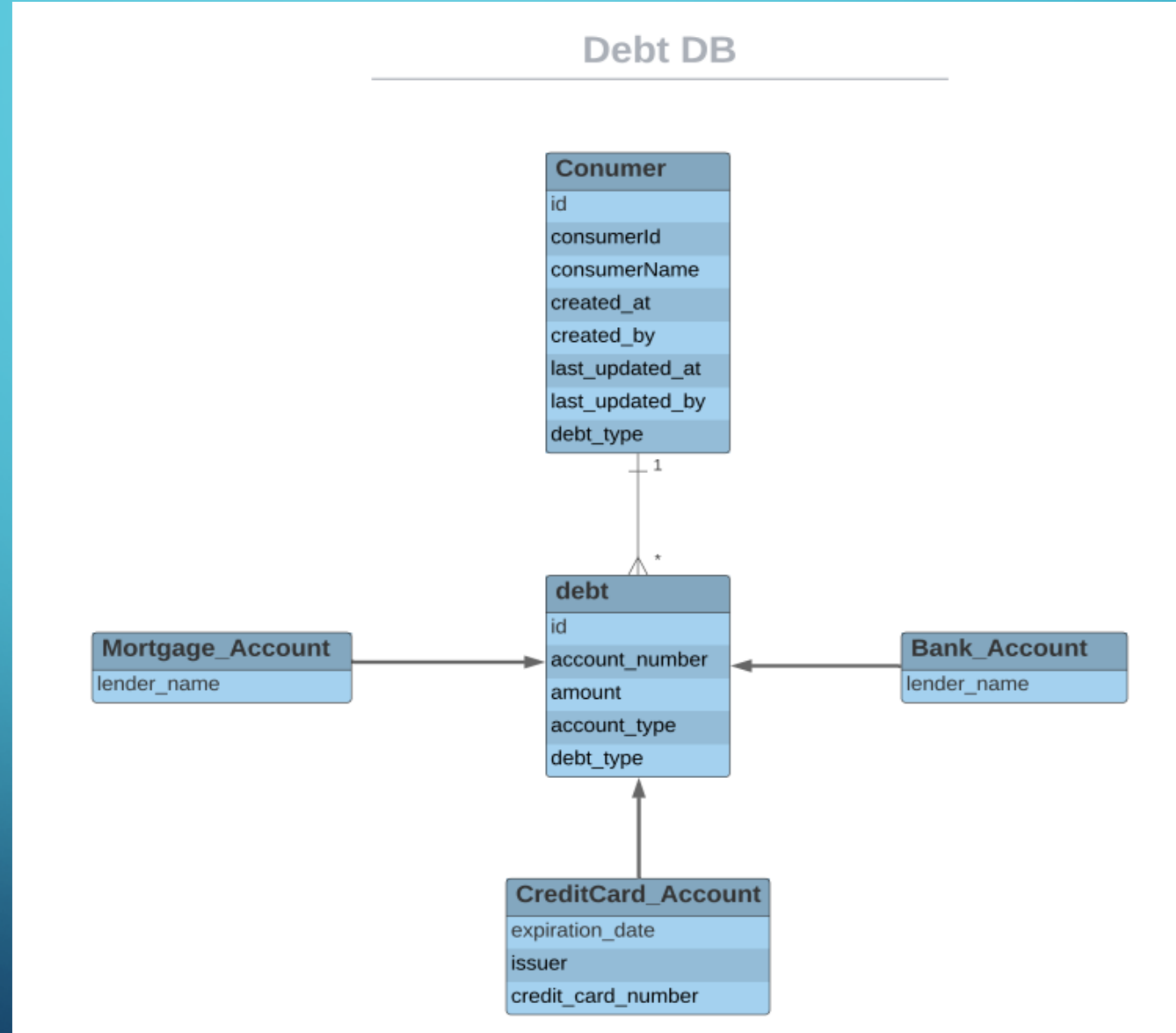
CHALLENGES:

- Structure of records
 - Time
 - Developing resource
- 
- 
- 

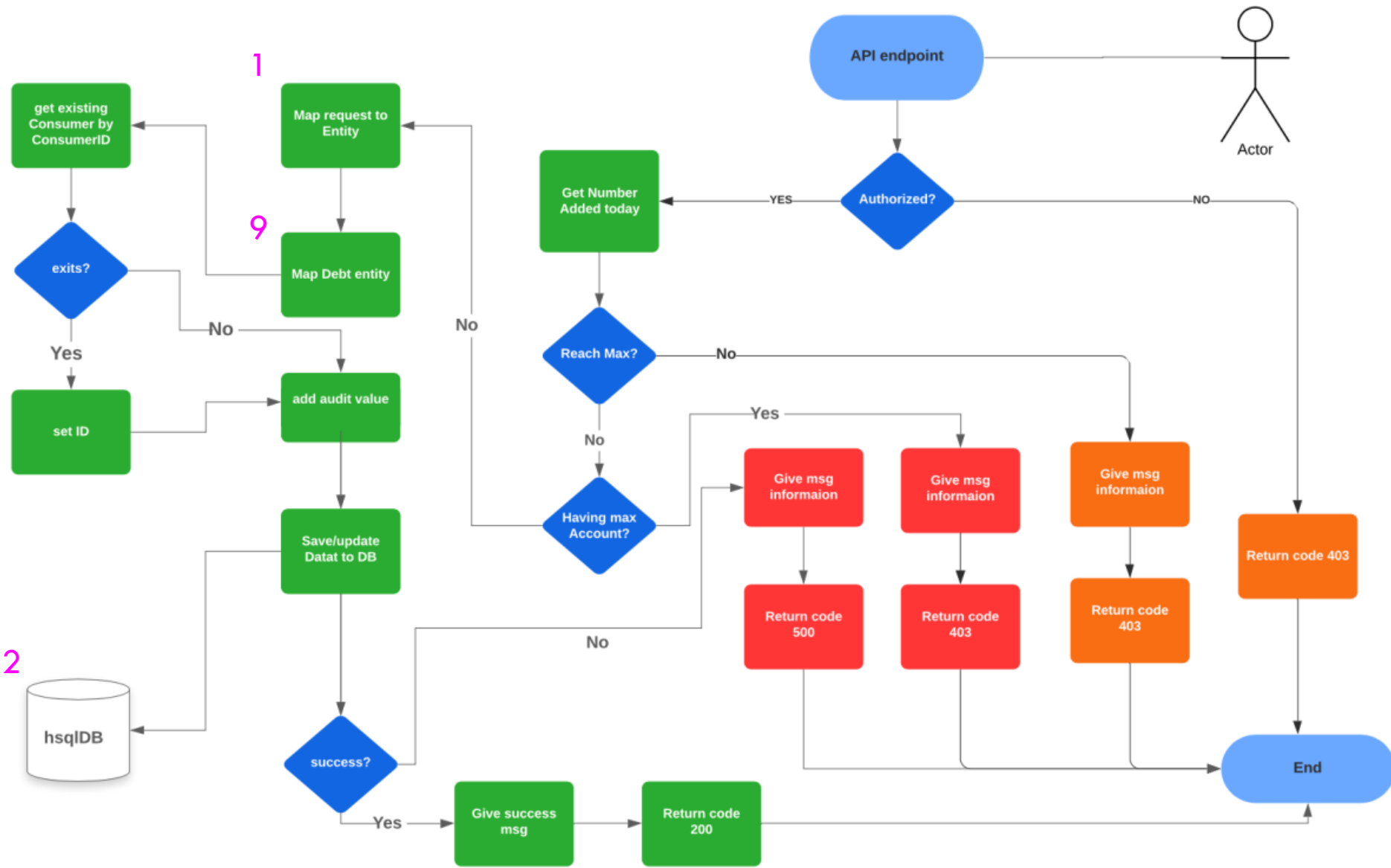
TECHNOLOGIES:

- SpringBoot
- Rest
- Spring-Scheduling
- hsqlDB
- Jaxb
- Orika
- Jsch
- Commont-net
- GoF
- Karate Automation test
- etc...

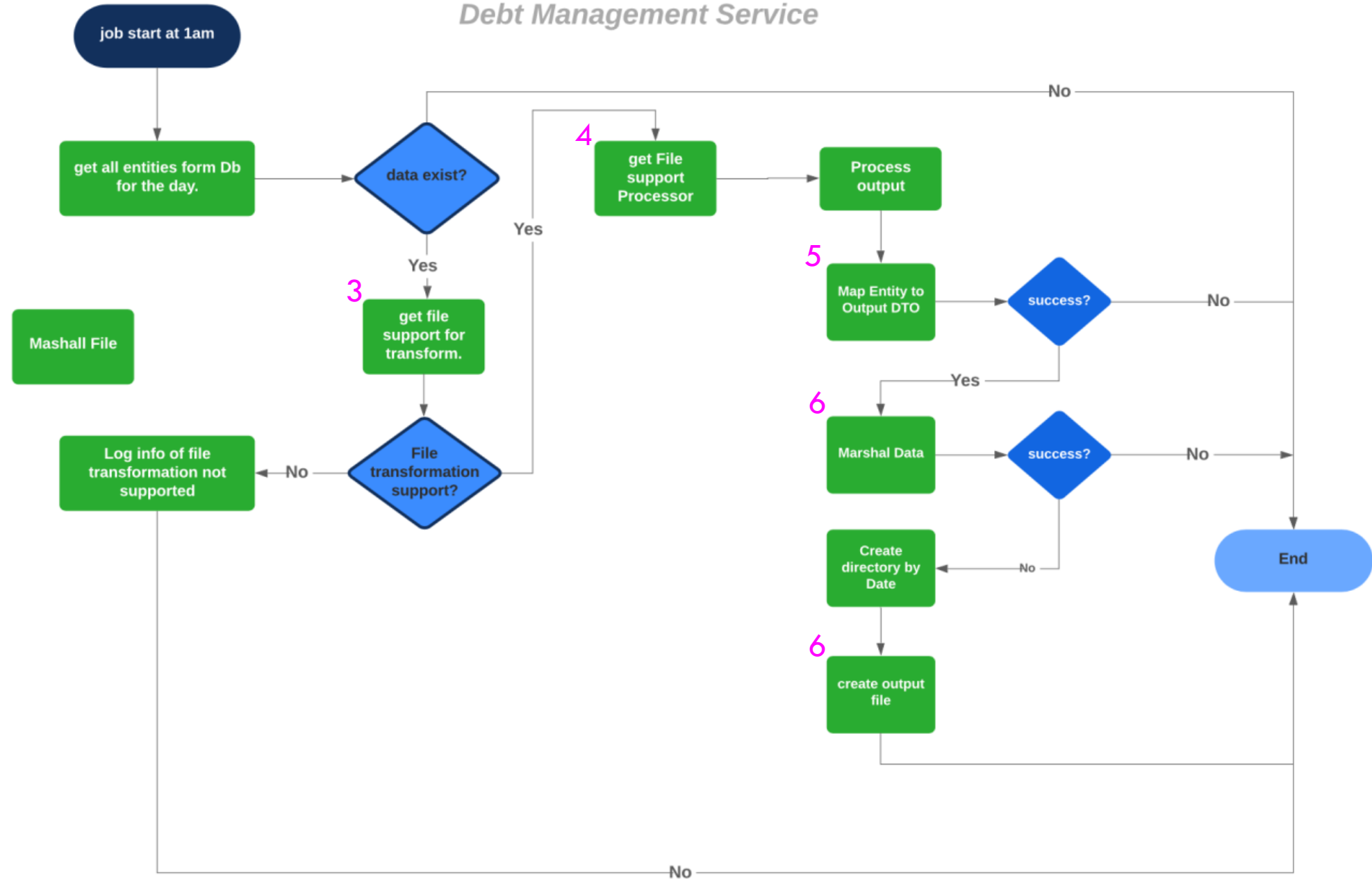
DIAGRAMS



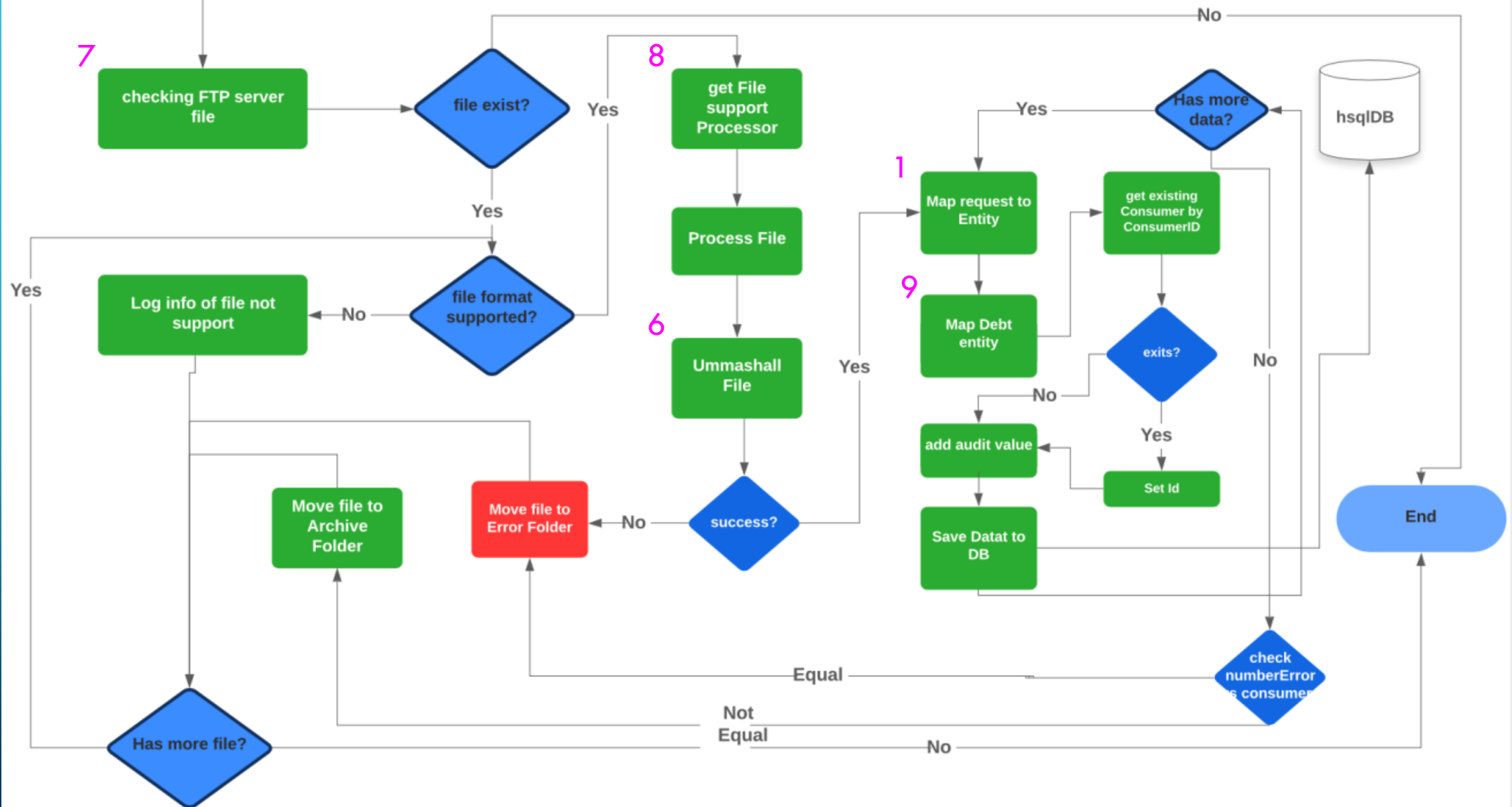
Debt Management Service



Debt Management Service



Debt Management Service



EXPLANATIONS

- Debt DB: As the debt can be BankAccount, CreditCardAccount, MortgageAccount and others(in the future); So, I am in “**debt**” I am using discriminatorColumn to distinguish those types.
- To avoid another join I decided to put all the fields of Description tags from input into “debt” as fields of Description will be dynamic according to Debt Type.

EXPLANATIONS (CONT)

- (1) : For mapping I am using "Orika" for data Mapping from Entity to DTO and vice versa.
- (2) : Since, I don't have any Database server installed, I am using hsqldb which is a memory DB
- (3) , (4) & (8) : I am Strategy Processor, where checking the supported file that is configurable in the properties. In the future, if another file is needed we can always update supported file and created new processor to process the output which is flexible enough.
- (5) : Similar to (1) using "Orika" for data mapping from DTO back to Entity.

EXPLANATIONS (CONT)

- (6) : marshalling and unmarshalling according to file support. For now it's XML so I am using JAXB for transformation. We can use Gson for example for JSON file in the future. After creating file I am using "JSCH" for uploading to SFTP server.
- (7) : Reading files from FTP using "common-net" library.
- (9) : For mapping debt entity to DTO and vice versa, Builder pattern to build the DebtType (BankAccount, CreditCardAccount, MortgageAccount) according to the DebtType of the data. So, in case there another debt type coming soon we can always easily build that object.

FUTURE GOALS:

- Add more unit tests to cover all the code at least 85% in order to pass the maven build using JACOCO
- Find another better way to to check the type of debt from input (RequestBody as well as input files.)
- Update DB using DB server

The background is a blue gradient. In the corners, there are white line-art illustrations of circuit boards. The top-left and bottom-left corners feature more complex, branching circuit patterns. The top-right and bottom-right corners have simpler, more linear circuit patterns.

DEMO

The background is a blue gradient. In the corners, there are white line-art illustrations of circuit boards or neural networks, with lines and small circles representing nodes.

QUESTION AND ANSWER

&

THANKS FOR YOUR TIME