Data Analytics Workflow Development on DBLP Dataset

- Sponsor: Pro. Jia Zhang
- Point of Contact: Keen Wang (keen.wang@sv.cmu.edu)
- Faculty Advisor: Pro. Jia Zhang
- Team: Team 5

Roadmap

Introduction

Motivation

Related technology

System design

System implementation

Demo

Conclusions and future work

Introduction

- DBLP is a well-known service that provides open bibliographic information on major computer science journals and proceedings.
- DBLP indexes more than 3 million publications' metadata.

dblp
computer science bibliography

Motivation

- Create the data analytics service for DBLP:
  - Build a network between authors and publications
  - Develop a user-friendly interface

- Help users get what they want effectively, efficiently and accurately:
  - Improve functionality of queries
  - Visualize the answer

Related Technology

Graph Database:

- A database management system with CRUD options working on a graph data model
- Good at handling data relationships

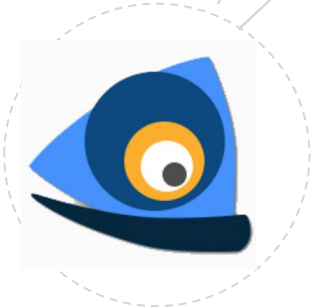
MySQL:

- Use relational database to store user information

Related Technology

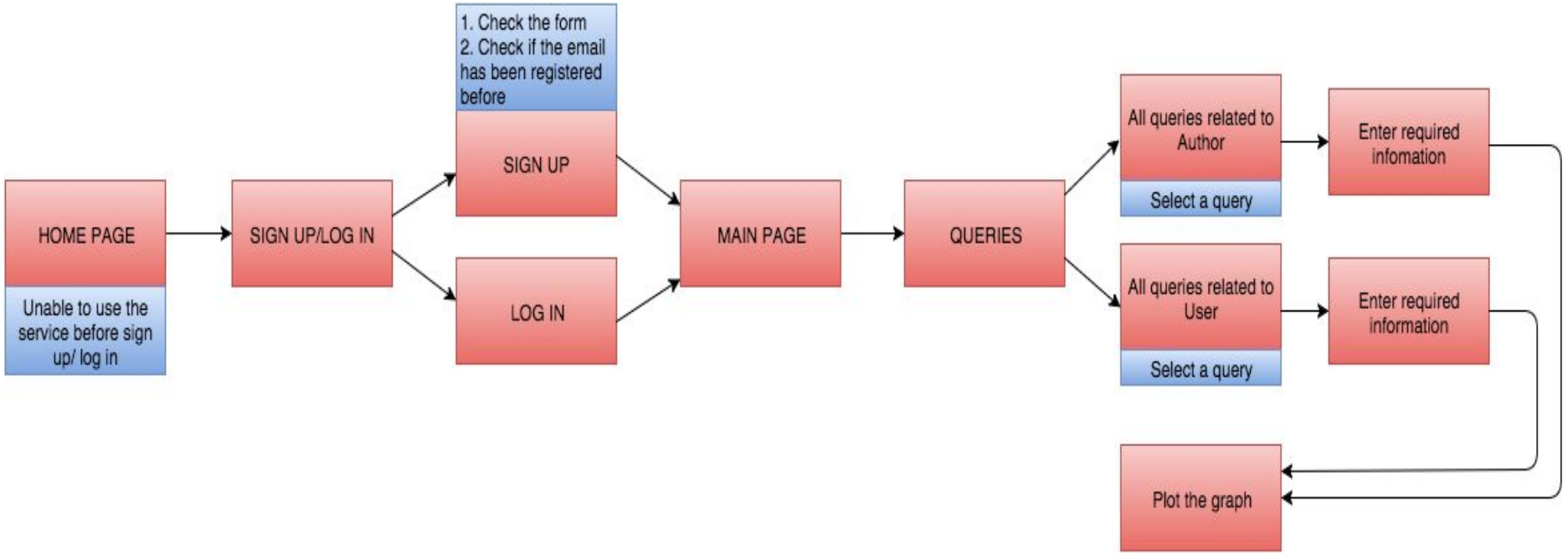
- Spring framework:
  - Provides a comprehensive programming and configuration model for modern Java-based applications

- vis.js:
  - Display dynamic, automatically organised, customizable network views.

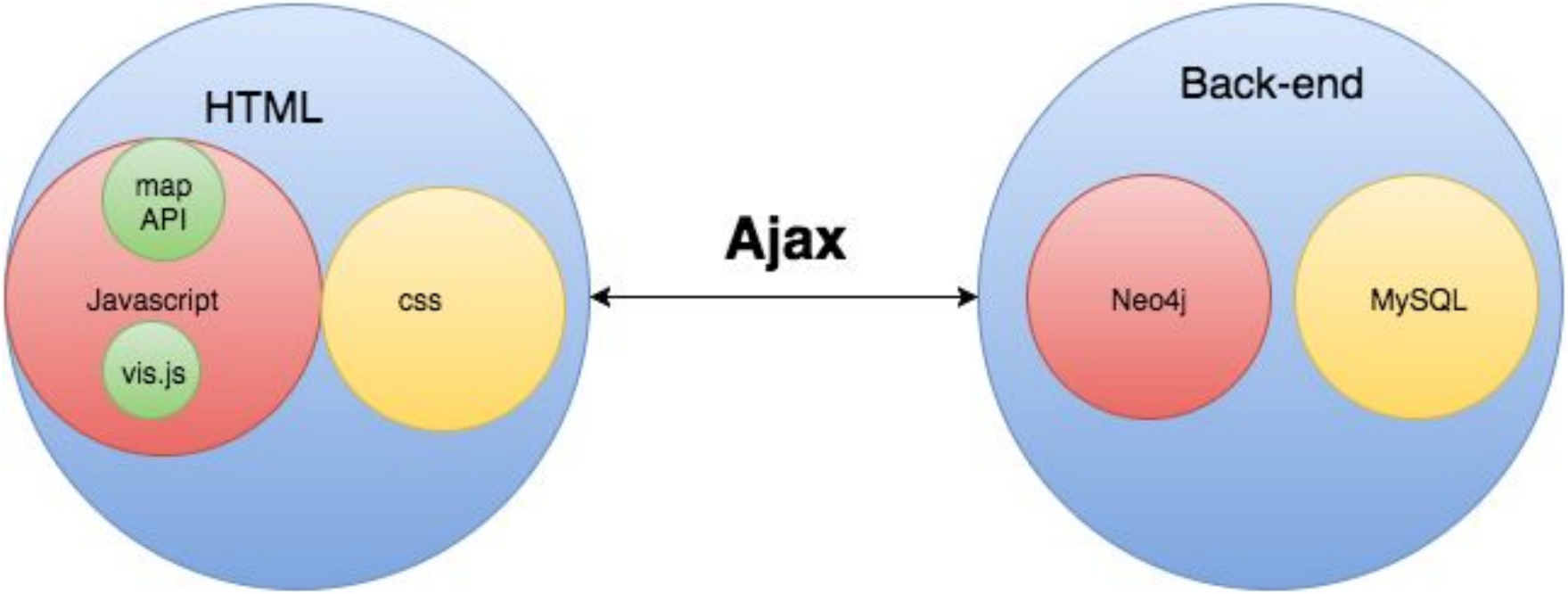


System Design (Front-end)

- Page flow:

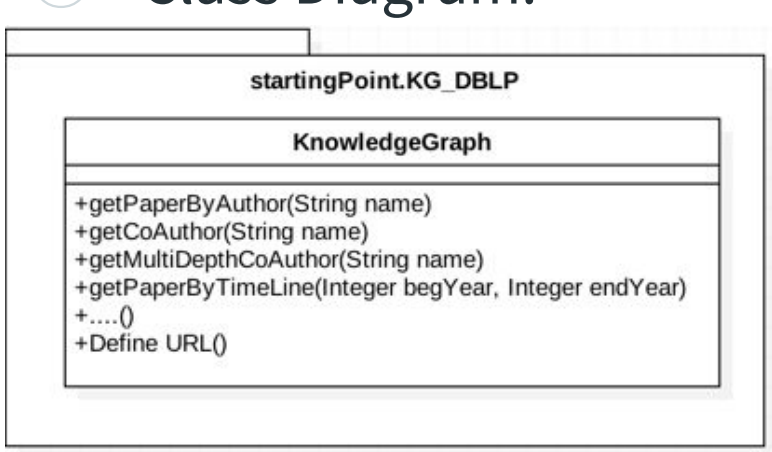


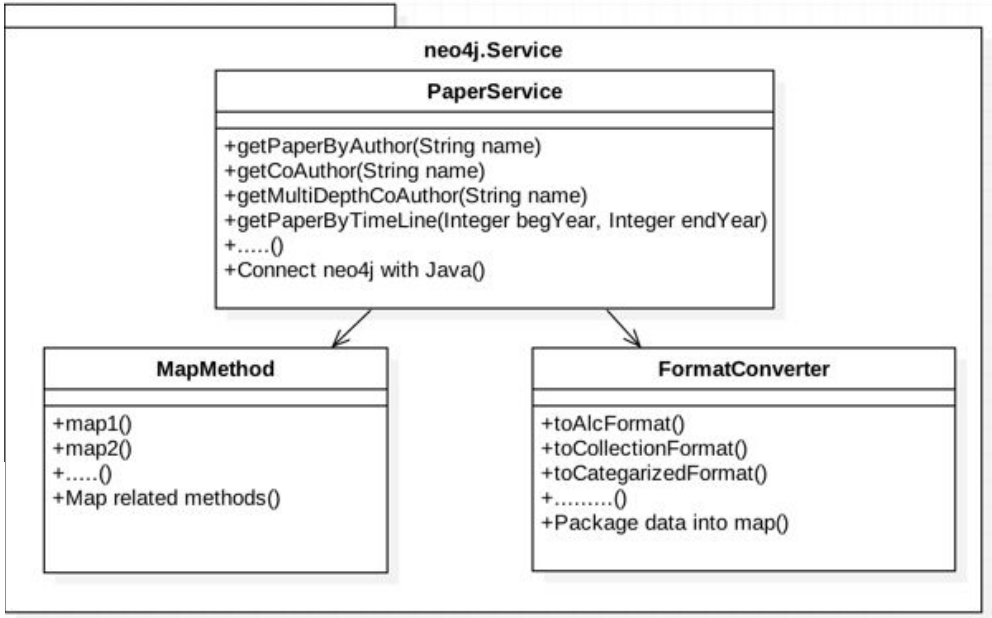
System Implementation (Front-end)



System Design (Back-end)

- Class Diagram:



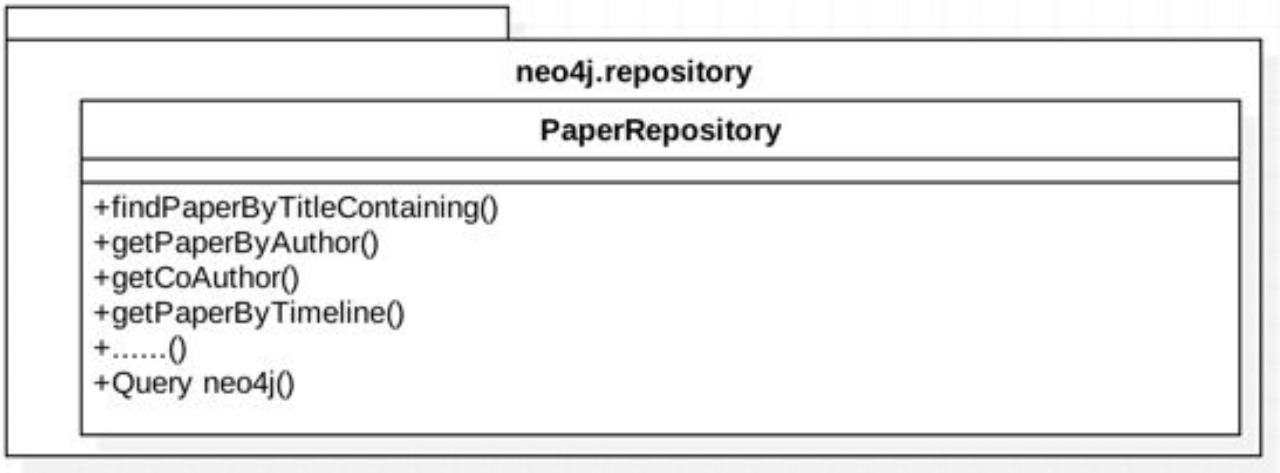



```
@RequestMapping("/findByTitleContaining")
public String findByTitleContaining(@RequestParam(value = "title",required = false) String title) {
    System.out.println("findByTitleContaining");
    title = title.replace('+', ' ');
    System.out.println("Title:     " + title);
    Map<String, Object> map = paperService.findByTitleContaining(title == null ? "" : title);
    String json = "";
    ObjectMapper mapper = new ObjectMapper();
    try {
        //convert map to JSON string
        json = mapper.writeValueAsString(map);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return json;
}
```

```
public Map<String, Object> findByTitleContaining(String s) {
    System.out.println("findByTitleContaining:     " + s);
    Iterator<Map<String, Object>> result = paperRepository.findByTitleContaining(s).iterator();
    return FormatConverter.toAlcFormat(result);
}
```

System Design (Back-end)

- Class Diagram:



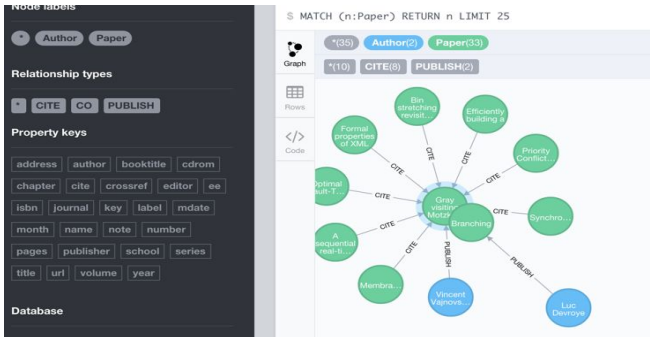

```
@Query("MATCH (p:Paper)<-[:PUBLISH]-(a:Author) WHERE a.name = {name} RETURN p.title as paper, collect(a.name) as cast LIMIT 50")
List<Map<String, Object>> getPaperByAuthor(@Param("name") String name);

@Query("MATCH (a1:Author)<-[:CO]-(a2:Author) WHERE a2.name = {name} RETURN a2.name as author, collect(a1.name) as cast LIMIT 50")
List<Map<String, Object>> getCoAuthor(@Param("name") String name);
```

System Implementation (Back-end)

- Parse and Load DBLP data:

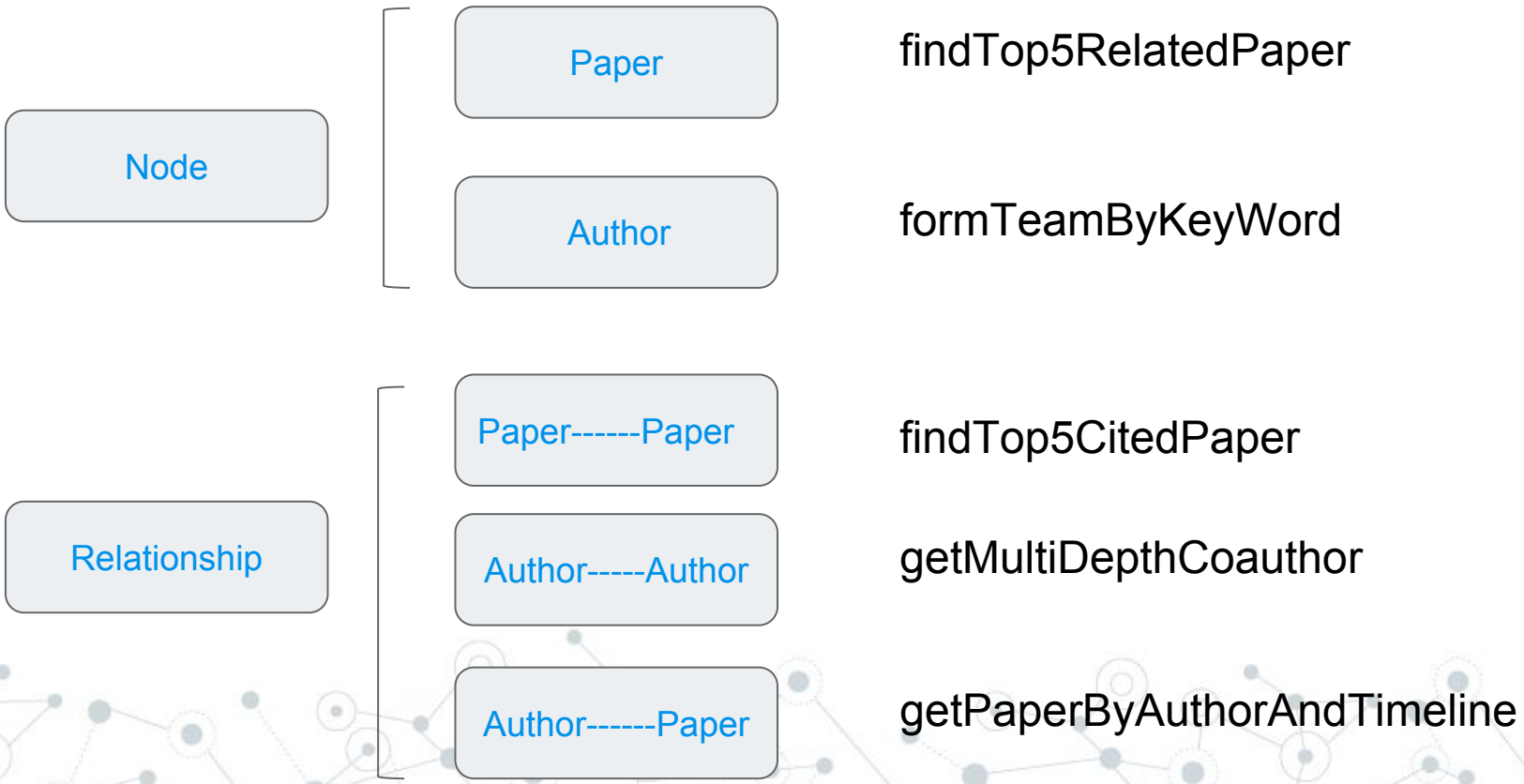

```
</article>
<article mdate="2011-01-11" key="journals/acta/Simon83">
<author>Hans-Ulrich Simon</author>
<title>Pattern Matching in Trees and Nets.</title>
<pages>227-248</pages>
<year>1983</year>
<volume>20</volume>
<journal>Acta Inf.</journal>
<url>db/journals/acta/acta20.html#Simon83</url>
<ee>http://dx.doi.org/10.1007/BF01257084</ee>
</article>
```



mdate	key	author	editor	title	pages	year	volume	journal	number	url
2011-01-11	journals/acta/Saxena	Sanjeev Saxena		Parallel Integer Sorti	607-619	1996	33	Acta Inf.	7	db/journals/acta/act
2011-01-11	journals/acta/Simon8	Hans-Ulrich Simon		Pattern Matching in T	227-248	1983	20	Acta Inf.		db/journals/acta/act
2011-01-11	journals/acta/Goodma	Nathan GoodmanlOded Shmueli		NP-complete Problen	171-178	1983	20	Acta Inf.		db/journals/acta/act
2011-01-11	journals/acta/Blum82	Norbert Blum		On the Power of Cha	425-433	1982	17	Acta Inf.		db/journals/acta/act
2013-11-28	journals/acta/Schonh	nhage		rpern der Charakteris	395-398	1977	7	Acta Inf.		db/journals/acta/act

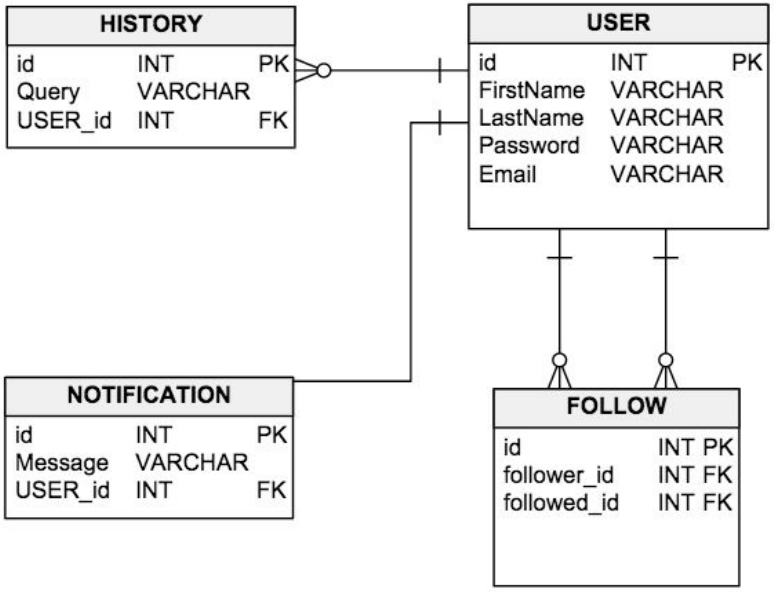
System Implementation (Back-end)

- Categorized Queries:

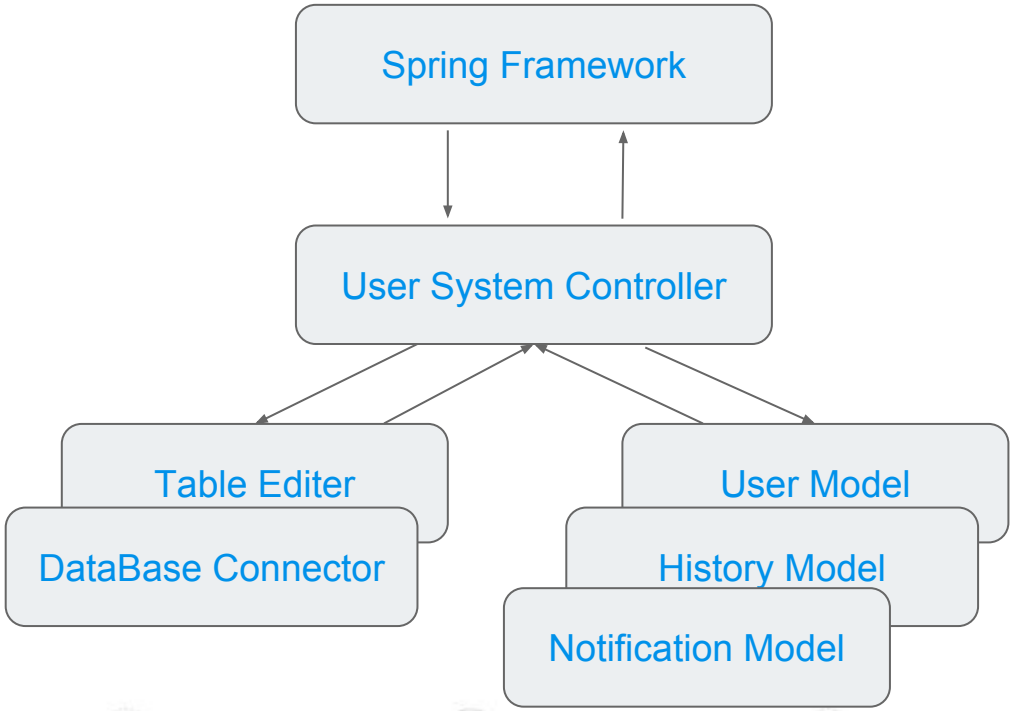


System Design (User-System)

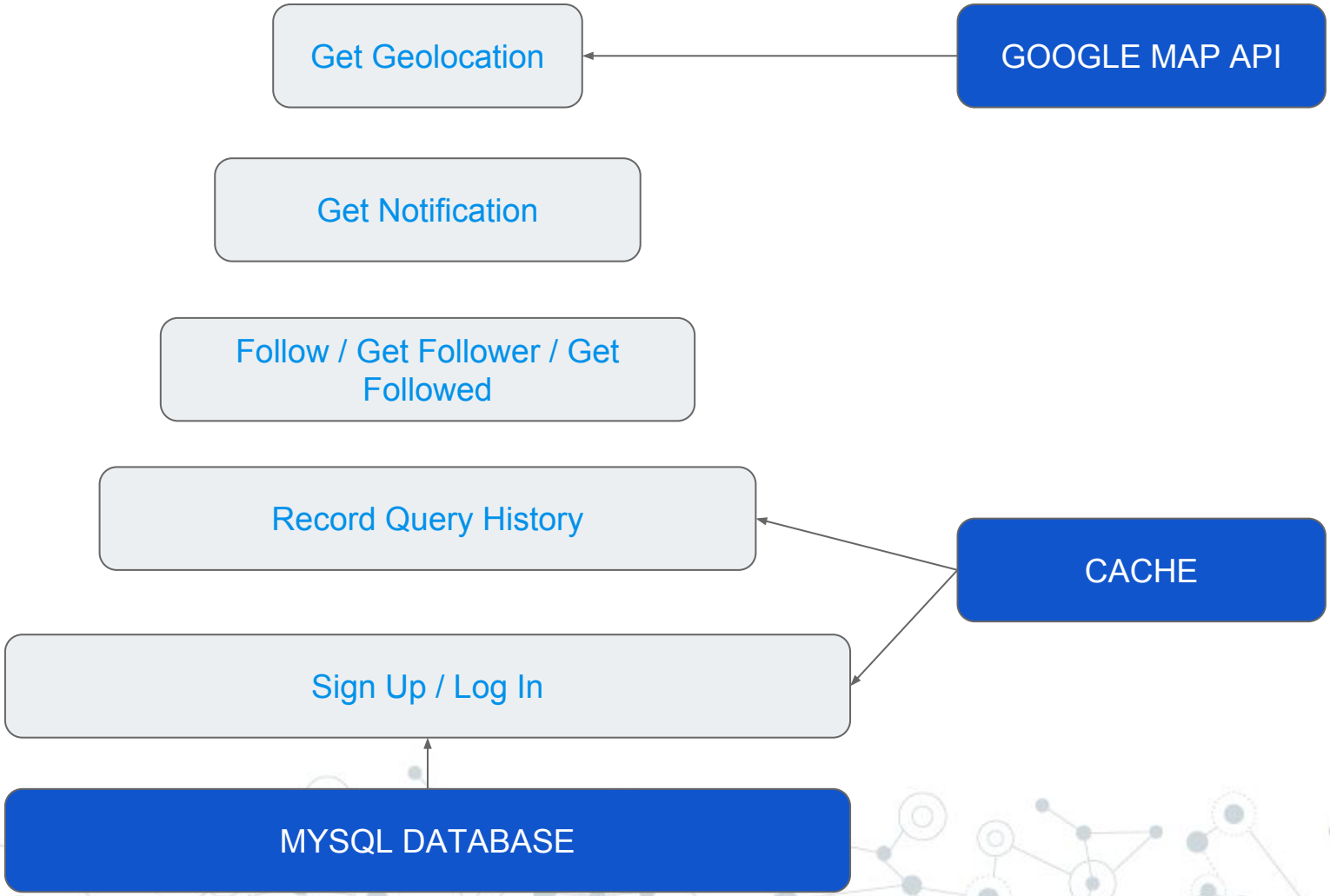
DataBase Schema



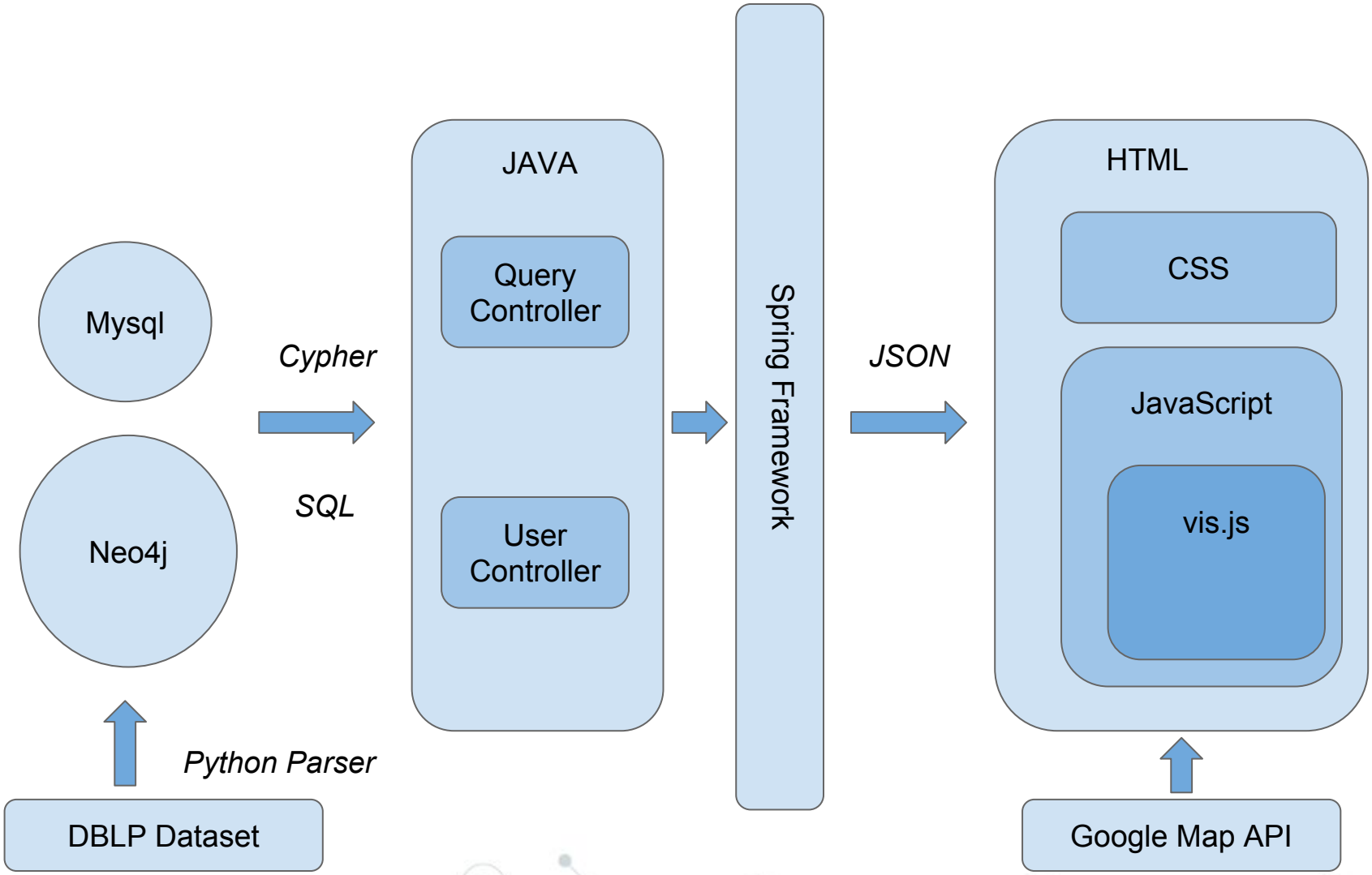
Backend Architecture



System Implementation (User-System)



System Architecture



Demo

Conclusion & Future Work

Problem:

Big data

Query time consuming

Future Work:

More friendly UI Design

Load all data

More efficient class design

Q&A

Thanks Jia & TAs!