

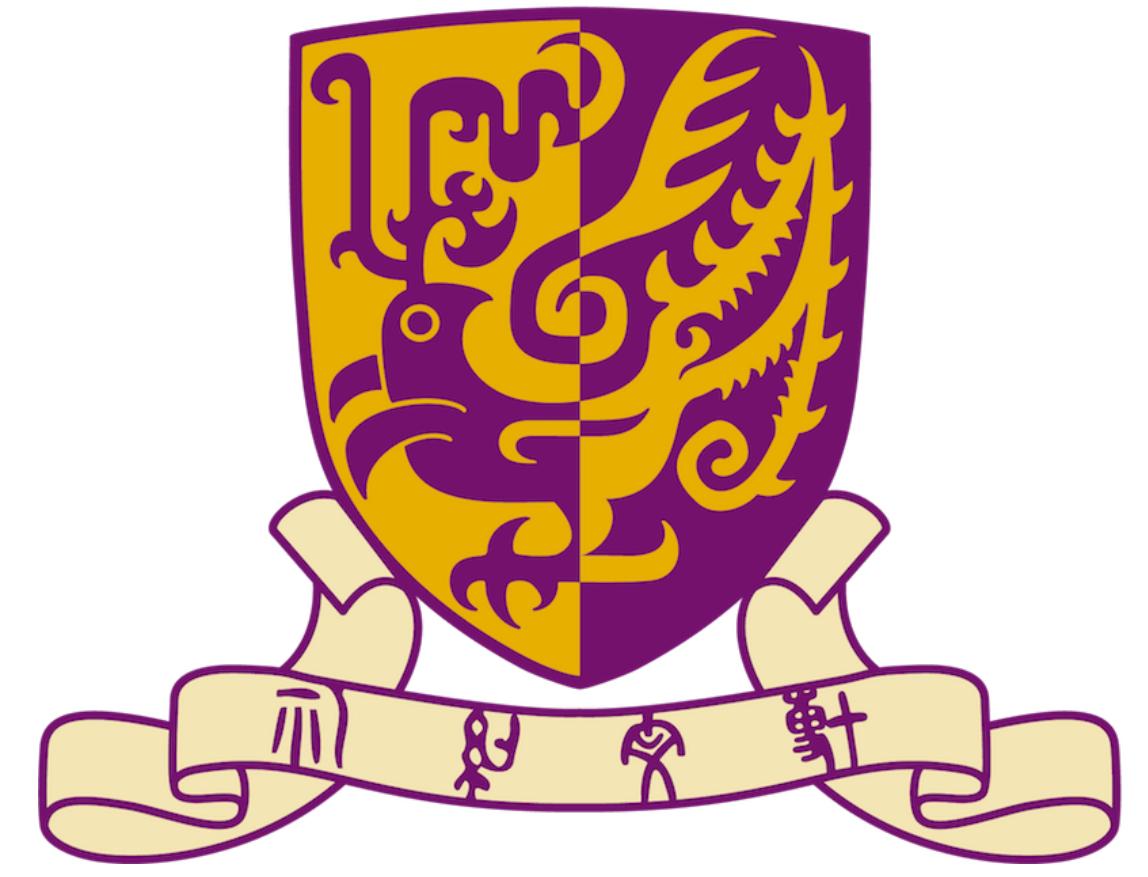


Cross Platform Application Development

Course: IERG4999 Student Name: Doria Tang

Supervisor: Prof. Kehuan Zhang

Dept. of Information Engineering, The Chinese University of Hong Kong



Objectives

- To create an online anonymous platform that allows users to express their opinions freely.
- To provide an informative platform that encourages users to explore different books.
- To develop a small community that users can meet people who share similar interests.

Introduction

Overview. The project is based on a cross-platform application framework that encourages students to exchange their books and ideas on an online platform.

Motivation. This project aimed to solve two major problems:

- Allow students to meet more friends through the use of social media application. [1, 2]
- Address the problem of wasted textbooks by encouraging them to exchange the used books.

Technical Background. The application is targeted to deploy on iOS and Android.

- Frontend: Flutter framework + Dart language [3]
 - Only a single codebase is used
 - Code is rendered as widgets in the element tree to represent the current state of user interface
 - The rendering tree is then used to generate a platform-agnostic Skia canvas
 - GPU renders the UI to the screen to achieve cross-platform effect

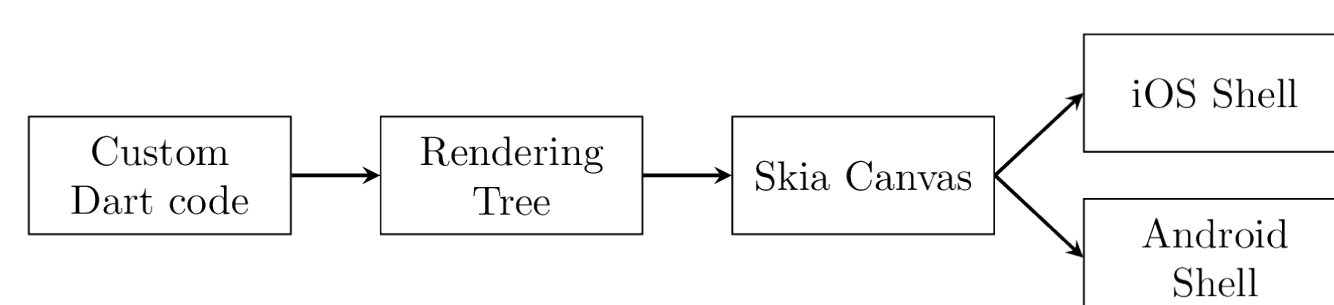


Figure: Flutter rendering process

- Backend: Node.js server to create various APIs
- Database: MongoDB to create non-relational documents [4]

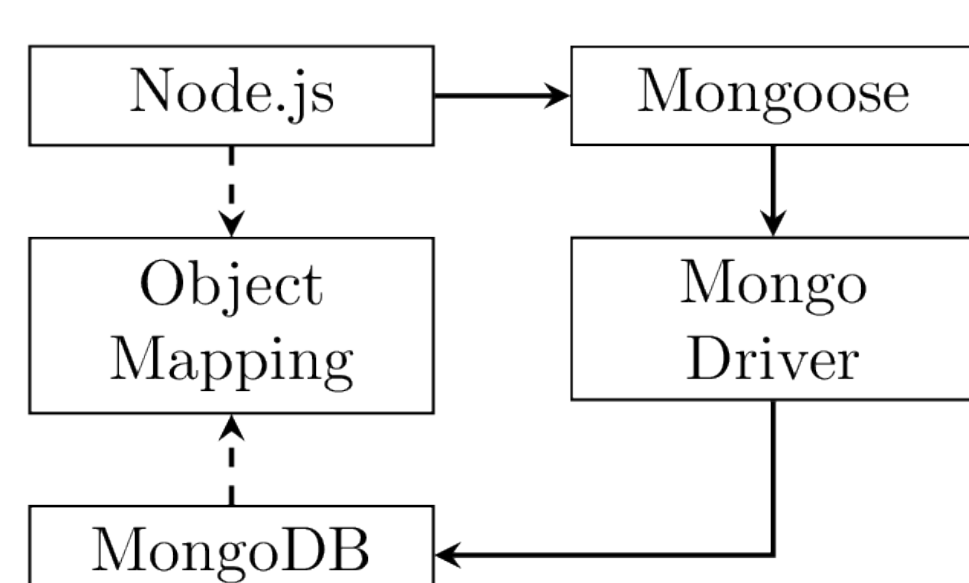


Figure: Nodejs and MongoDB relations

Node.js uses the Mongoose package as the driver to communicate with MongoDB and perform various actions. The Schema and models are defined using Mongoose API and send the queries back to the server in JSON-format.

Methodology

The application consists of three main parts: the frontend application, the backend server, and the database. The three-layer architecture allows for separations for the whole application, so that it allows for easier maintenance and performs independent updates on different components.

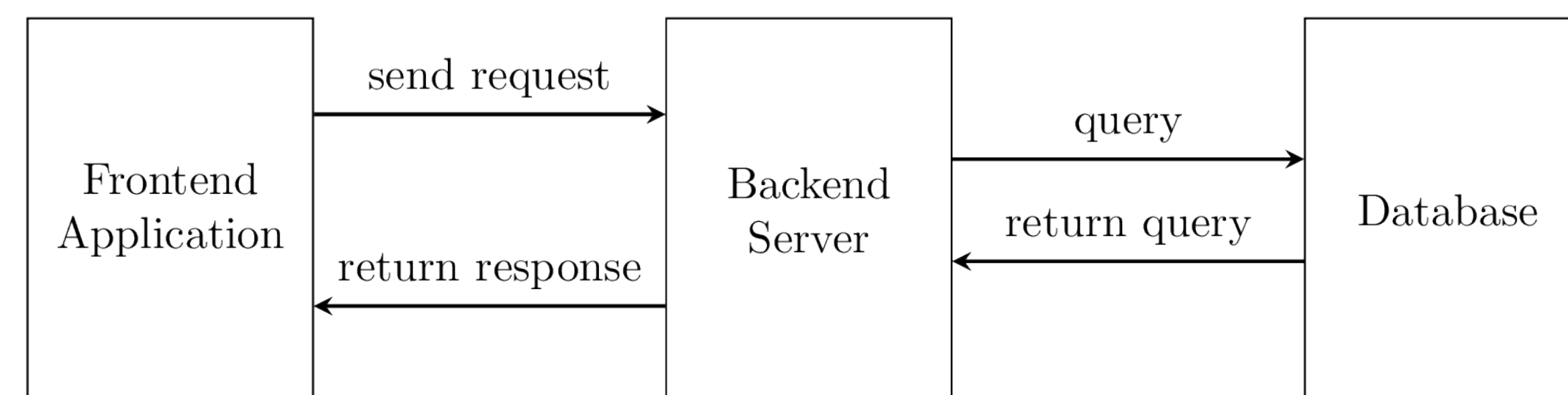


Figure: Application Architecture

Frontend Application. The frontend handles user interaction and information display. It focuses on:

- **User-friendly interface:** night mode, responsive screen etc.
- **User actions:** uploading books, leaving comments etc.
- **Third-party API implementations:** ChatGPT, New York Times API

***This part will be presented in the live demo.*

Backend Server. Backend server connects frontend to database and handles user data. The API calls are categorized as:

- **User:** manage user information, such as changing account password.
- **Thread:** manage discussion threads, such as fetching content from all the threads, creating threads, giving likes or dislikes.
- **Comment:** manage comments, such as fetching information of all the comments in a thread, creating comments, editing comments.
- **Book:** manage the book uploading action, such as adding book picture and retrieving book information.

Database. The non-relational database can store different data types in the documents. A query doesn't have to view several tables to find the desired data, which leads to a better performance compared to traditional database.

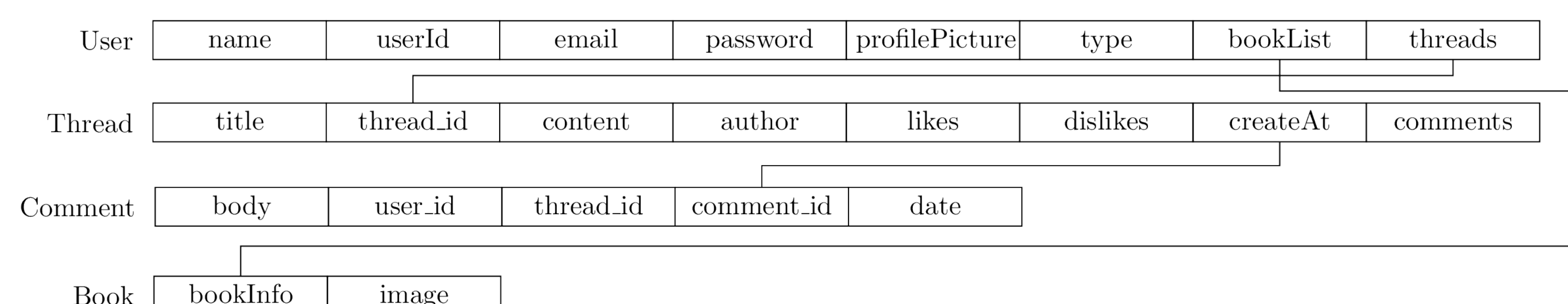


Figure: Relations between Documents in MongoDB

Documents in MongoDB are linked with their object ID. This allows developers to modify the database dynamically.

Testing

User Acceptance Tests:

The purpose of the test is to check whether the UI widgets are running properly before putting to production environment.

	User Interface related	User Actions	Third-Party APIs
Passed	7	24	4
Failed	1	7	2
Total	8	31	6

Table: User Acceptance Tests Evaluation

The majority of the failed tests were attributed to UI components incorrectly displaying information, with some of them showing incorrect updates.

→ *Improvements: automation testing, invite more testers*

Functional Tests:

The functional test for the backend server's API is conducted using Postman and involves running test script to verify the accuracy of the response structure, such as response status code, response time, and the presence of various response keys.

→ *Improvements: conduct testing with corner cases, such as including special characters*

Future Work

- **Real-Time Messaging System:** Users can message each other and meet new friends who share similar interests.
- **Recursive Reply:** A recursive reply allows users to respond to other users' comments. Users can reply to the target comment and have a more consistent conversation between user.
- **Social Login:** User can link their social media account and invite their friends to use the app.

Conclusion

This project aims to provide a cross-platform application that can offer a sustainable alternative for textbook disposal. The application allows user to perform basic actions, such as managing their accounts, start a discussion thread etc. Various testing are conducted to verify the functionalities. The scope of this project can be further extended to bring a better user experience.

References

- [1] M. D. Hopp and M. Händel. The structure of social networks and its link to higher education students' socio-emotional loneliness during covid-19. *Frontiers in Psychology*, vol 12, 2022.
- [2] H. Parlak Sert and H. Baškale. Students' increased time spent on social media, and their level of coronavirus anxiety during the pandemic predict increased social media addiction. *Health Information and Libraries Journal*, 2022.
- [3] Flutter architectural overview, updated 2023.
- [4] The modern application stack – part 2: Using mongodb with node.js | mongodb blog, updated 2023.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof. Zhang for his support and guidance.