# Implementation and Deployment: Tripedia

Yulin Zhang
*7th Group*
*Software Engineering*
Montreal, Canada
silveralex2023820@gmail.com

Yuhang Chen
*7th Group*
*Software Engineering*
Montreal, Canada
yuhang.chen@mail.concordia.ca

Jiaxi Yang
*7th Group*
*Software Engineering*
Montreal, Canada
yjxyang2@outlook.com

Boyang Wang
*7th Group*
*Software Engineering*
Montreal, Canada
wangboyang0626@outlook.com

## I. Architectural Design Process

**Question: Identify and articulate what are your architectural designs and the associated software engineering process.**

Architectural designs serve as the blueprints for our software systems, shaping their form and defining the relationships among components. Our design choice comes with its own set of considerations and implications. In chapter, we will dissect the architectural design in Tripedia system.

### A. *Architectural Design*

### B. *Software Engineering Process*

## II. Architecture Design Consideration

**Question: Indicate if any revision to your architectural design is necessary. If the answer is Yes, please explain what revision, and the reason.**

In this chapter, we will explore the senarios about our system. And we will give reasons about our descision about revision.

### A. *Revision Consideration*

### B. *Reasons*

## III. The Adoption of MVC and Layered Architecture

**Question: Further adopt MVC and Layered Architecture Pattern to your ICDE-App. Describe your design decisions, and discuss the pros and cons of the design.**

In this chapter, we discuss the adoption of MVC and Layered Architecture Patterns into Tripedia system. By embracing these architectural patterns, we aim to achieve a balance between flexibility, scalability, and maintainability. Finally, we explain the pros and cons of our system architecture.

### A. *The Adoption of MVC Architecture Pattern*

### B. *The Adoption of Layered Architecture Pattern*

### C. *The Pros and Cons*

## IV. Software Metrics and Granularity of Components

**Question: Please make a statistical count of software metrics from all your tasks implemented so far and form a table given the template below.**

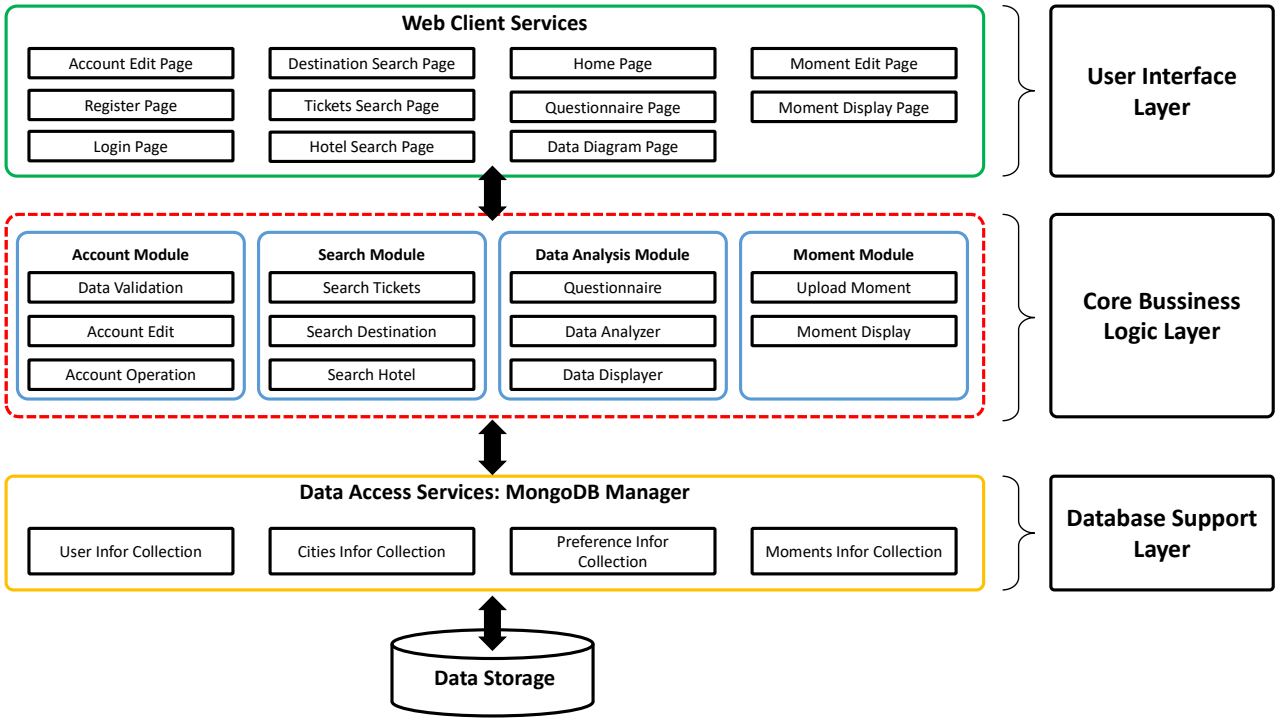


Fig. 1. The Adoption of Layered Architecture Pattern in Tripedia.

TABLE I
THE SOFTWARE METRICS AND GRANULARITY OF COMPONENTS.

| **Task Name** | ***LOC*** | ***Component Granularity Level*** | ***Numbers of Units*** |
|---|---|---|---|
| Account Login and Password Validation | | Module | 1 |
| Data Displayer | | Python Class | 1 |
| Data Analyzer | | Python Class | 1 |
| Questionnaire | | Module | 1 |