**Zusammenfassung**

**Arbeitsberichte**

## Inhaltsverzeichnis

[1 Kalenderwoche 19 0](#_Toc202903711)

[2 Kalenderwoche 20 0](#_Toc202903712)

[3 Kalenderwoche 21 1](#_Toc202903713)

[4 Kalenderwoche 22 1](#_Toc202903714)

[5 Kalenderwoche 23 1](#_Toc202903715)

[6 Kalenderwoche 24 2](#_Toc202903716)

[7 Kalenderwoche 25 2](#_Toc202903717)

[8 Kalenderwoche 26 3](#_Toc202903718)

[9 Kalenderwoche 27 3](#_Toc202903719)

# Kalenderwoche 19

|  |  |  |
| --- | --- | --- |
|  | **Arbeitsbericht KW 19** |  |
| **KW** | **Arbeitsbericht** | Σ |
|  | **Yixiang Li** |  |
| 19 | * Teilnahme an Gruppendiskussionen zur Entwurfsplanung der Softwarearchitektur * Teilnahme an Gruppendiskussionen zur Aufteilung der Backend-Aufgaben; Lu und ich waren für die Entwicklung von APIs und Datenbankkomponenten zuständig * Teilnahme an Gruppendiskussionen zur Evaluierung des geplanten Tech-Stacks * Erste Abstimmung mit dem Frontend-Team zur Klärung der API-Anforderungen * Einarbeitung in FastAPI, Entwurf des Backend-Strukturdiagramms für die aktuelle Aufgabe * Entwicklung eines Demo-Programms mit FastAPI * Verifizierung der Docker-basierten Bereitstellung von MongoDB | 8h |

# Kalenderwoche 20

|  |  |  |
| --- | --- | --- |
|  | **Arbeitsbericht KW 20** |  |
| **KW** | **Arbeitsbericht** | Σ |
|  | **Yixiang Li** |  |
| 20 | * Teilnahme an der Gruppendiskussion zur Festlegung der Aufgaben für die kommende Woche * Test der JSON-Datenformate vom Frontend; aufgrund ausstehender Anpassungen auf der Frontend-Seite wird diese Aufgabe auf nächste Woche verschoben * Unterstützung bei der Implementierung der Datenbank und der Integration von RabbitMQ * Vertiefte Auseinandersetzung mit FastAPI und einfache Simulation der HTTP-Methoden GET, PUT und POST | 8h |

# Kalenderwoche 21

|  |  |  |
| --- | --- | --- |
|  | **Arbeitsbericht KW 21** |  |
| **KW** | **Arbeitsbericht** | Σ |
|  | **Yixiang Li** |  |
| 21 | * Zuständig für die Kommunikation und Koordination mit dem API-Team, den anderen Backend-Teams sowie dem Frontend-Team. (3h) * Verantwortlich für die Validierung des vom Frontend übermittelten JSON-Dateiformats durch die Backend-API. (4h) * Überträgt das erfolgreich validierte JSON-Datenformat in die Datenbank. (2h) * CURD-Endpunkte von Encoding implementiert. (2h) | 11h |

# Kalenderwoche 22

|  |  |  |
| --- | --- | --- |
|  | **Arbeitsbericht KW 22** |  |
| **KW** | **Arbeitsbericht** | Σ |
|  | **Yixiang Li** |  |
| 22 | * Zuständig für die Kommunikation und Koordination mit dem API-Team, den anderen Backend-Teams sowie dem Frontend-Team. (3h) * Die Typen der Schnittstellen untersucht, die als Nächstes für encoding und result implementiert werden sollen.(3h) * Für die bereits implementierte PUT-Schnittstelle habe ich die Validierung der geänderten Inhalte realisiert.(2h) | 8h |

# Kalenderwoche 23

|  |  |  |
| --- | --- | --- |
|  | **Arbeitsbericht KW 23** |  |
| **KW** | **Arbeitsbericht** | Σ |
|  | **Yixiang Li** |  |
| 23 | * Die Validierungslogik für die Kodierung wurde neu geschrieben (2h) * Route modularisiert, um sie später für die anderen drei APIs vorzubereiten (1h) * Dockerization of fastapi service (4h) * It is now possible to use the commands docker-compose up and docker-compose up --build in the root directory to simultaneously start the three services in Docker: FastAPI, MongoDB, and RabbitMQ. (2h) * Enable RabbitMQ connection with port config (1h) | 10h |

# Kalenderwoche 24

|  |  |  |
| --- | --- | --- |
|  | **Arbeitsbericht KW 24** |  |
| **KW** | **Arbeitsbericht** | Σ |
|  | **Yixiang Li** |  |
| 24 | * Die CRUD-APIs für Datasets, Runs und Results sind bereits implementiert.(4h) * Darüber hinaus wurden für jede der drei Routenarten jeweils drei gültige und ein ungültiges Beispiel hinzugefügt.(1h) * Außerdem wurde eine Testfunktion für alle vier APIs (Encoding, Datasets, Runs und Results) implementiert.(1h) * Und für alle vier API-Routedateien wurden erklärende Kommentare zur besseren Verständlichkeit hinzugefügt.(1h) * Die README.md wurde aktualisiert und enthält nun Anweisungen zum Starten des Backend-Dienstes mit Docker.(1h) | 8h |

# Kalenderwoche 25

|  |  |  |
| --- | --- | --- |
|  | **Arbeitsbericht KW 25** |  |
| **KW** | **Arbeitsbericht** | Σ |
|  | **Yixiang Li** |  |
| 25 | * Batch-Abruf-Endpunkt für /api/resources hinzufügen (Post) (3h) * Beispiele für ungültige und gültige Anfragen hinzufügen(2h) * Dokumentation für den Batch-Abruf-Endpunkt hinzufügen (2h) * Kommunikation mit Frontend und Teamleitern(1h) | 8h |

# Kalenderwoche 26

|  |  |  |
| --- | --- | --- |
|  | **Arbeitsbericht KW 26** |  |
| **KW** | **Arbeitsbericht** | Σ |
|  | **Yixiang Li** |  |
| 26 | * Das Feld „measure\_index“ (run.py) wurde auf „0“ (Typ: „int“) initialisiert und wird später vom Worker aktualisiert. (1h) * Das Feld „qubits“ (run.py) wurde in den „Task-Daten“ hinzugefügt, um die Anzahl der vom Worker verwendeten Qubits anzugeben.(1h) * Das Format „encoding“ wurde erweitert, um das Feld „qubits“ zu unterstützen(2h) * Direkte Eingabe von „qubits“ vom Frontend aus möglich; (1h) * Name der Kodierung geändert (Typ -> Gate, Ziel -> Drähte).(0.5h) * Falls nicht angegeben, wird sie im Backend automatisch mithilfe von „Drähten + 1“ abgeleitet; (0.5h) * Validierungslogik implementiert, um sicherzustellen, dass die Anzahl der „qubits“ mit der Anzahl der „Eingaben“ übereinstimmt;(1h) * Alle chinesischen Kommentare im Frontend-Code ins Englische übersetzt.(1h) | 8h |

# Kalenderwoche 27

|  |  |  |
| --- | --- | --- |
|  | **Arbeitsbericht KW 27** |  |
| **KW** | **Arbeitsbericht** | Σ |
|  | **Yixiang Li** |  |
| 27 | * parameters und inputs entfernt (1h) * In params sind nun int, float oder "input\_n" erlaubt (2h) * qubits in qubit\_count umbenannt (3h) * Logik für inputs = qubit\_count angepasst (1h) * Entspricht nun der Logik des Validators sowie dem aktualisierten encoding-Schema und den Beispielen (1h) | 8h |