

# m321 - praktische Arbeit (LB2)

## Lernziele

- Zusammenarbeit mit anderen Entwicklern im Rahmen der Entwicklung eines verteilten Systems üben
- Systemkomponente mit klar definierter Schnittstelle entwickeln
- Systemkomponente in verteiltem System integrieren
- Umsetzung von Hochverfügbarkeitsmassnahmen

## Rahmenbedingungen

- **Gruppenarbeit:** 4 Personen pro Gruppe (eine 3er Gruppe)
- **Aufwand / Umfang:** ca. 20 Lektionen pro Person (wird im Unterricht zur Verfügung gestellt und sollte für ein gutes Resultat ausreichend sein)
- **Laufzeitumgebung:** Stellen Sie sicher, dass alle Ihre Systemkomponenten einfach in Betrieb genommen werden können. Die Laufzeitumgebung können sie frei wählen: LernMAAS, AWS, lokale VMs, Docker.
- **Besprechung Arbeitszwischenstand:** Nachdem Sie mit der Umsetzung begonnen haben und bereits einen gewissen Arbeitsstand erreicht haben (aber noch nicht fertig sind) - ca. in der Hälfte der Zeit - sollten Sie Ihren aktuellen Arbeitsstand mit der Lehrperson besprechen. Gehen Sie dafür auf die Lehrperson zu und vereinbaren mit ihr den Zeitpunkt, wann die Besprechung stattfinden soll.
- **Verwendung bestehender Software:** Sie dürfen bereits bestehende Software verwenden. Dies kann helfen Zeit zu sparen und den Fokus auf die Vertiefung der Modulthemen zu legen. Sie müssen aber keine bestehende Software verwenden, sondern können auch alles von Grund auf entwickeln. Wenn Sie jedoch bereits bestehende Software verwenden, dann geben Sie diese möglichst früh im Projektverlauf der Lehrperson ab.
- **Verwendung von GenAI:** Sie dürfen GenAI frei verwenden (dokumentiert wo und wie), denken sie aber daran dass damit die Erwartungen an Qualität und Funktionsumfang steigen. Es wird erwartet dass sie verstehen wie die generierten Teile funktionieren und es auch erklären können.

Bezüglich Aufwand entscheidet jeder für sich persönlich ob noch ausserhalb des Unterrichts zusätzlich Zeit in die Projektarbeit investiert werden soll oder nicht. Bitte achten Sie aber bei Ihren Überlegungen auf Ihre Work/Life-Balance (nicht dass es zu viel wird).

# Grundidee

Ihre Gruppe wird gemeinsam ein verteiltes System entwickeln. Dabei ist die Idee, dass jeder Entwickler mindestens für eine Systemkomponente innerhalb des verteilten Systems die Verantwortung trägt. Bitte stellen Sie innerhalb der Gruppe sicher, dass die Aufwände fair verteilt sind. Eine Person darf auch mehr wie eine Systemkomponente entwickeln. Achten Sie aber dabei darauf, dass es für niemanden "zu viel" wird.

Welche Aufgabe das verteilte System erfüllen soll, dürfen Sie komplett frei entscheiden. Wichtig ist, dass das System aus mindestens gleich vielen Systemkomponenten besteht wie die Gruppe Mitglieder hat. Jede Person ist dann für eine Systemkomponente und deren Implementierung, Integration ins verteilte System und Dokumentation verantwortlich. Bitte teilen Sie die Verantwortlichkeiten bei der Abgabe der Planung der Lehrperson schriftlich mit.

Sollten Sie keine eigenen Ideen haben, was für ein verteiltes System Sie entwickeln möchten, finden Sie in folgender Liste einige Vorschläge, die Sie für sich adaptiert oder auch 1 zu 1 übernehmen dürfen:

- **Online-Shop:** Mögliche Aufteilung: Warenkorb / Kasse, Benutzerverwaltung, Produktverwaltung und Lagerverwaltung
- **Mini Dropbox:** Mögliche Aufteilung: Benutzerverwaltung, Storage Service und Statistik
- **Notenverwaltung:** Mögliche Aufteilung: Benutzerverwaltung, Klassenverwaltung, Notenerfassung und Statistik
- **Lagerverwaltung:** Mögliche Aufteilung: Benutzerverwaltung, Lagerverwaltung, Produktverwaltung und Statistik.
- **IoT-Lösung:** Beispielsweise für freie Parkplätze in der Stadt aufgeteilt in Benutzerverwaltung, Parkplatzverwaltung, Informationstafelverwaltung (der Tafeln die Anzeigen wo was frei ist) und Statistik.

Diese Liste ist nicht abschliessend und soll Ihnen lediglich als Gedankenanstoss für die Umsetzung Ihrer eigenen Ideen dienen. Die Verwendung von bestehenden Applikationen ist - wie oben erwähnt - grundsätzlich erlaubt. Wenn Sie aber eine bestehende Applikation verwenden wollen, dann ist es **zwingend erforderlich**, dass Sie mir bei der Abgabe der Planung auch den **Sourcecode der bestehenden Applikation mit abgeben**, den Sie in der Arbeit nutzen und erweitern werden.

## Vorgehen

Folgendes Vorgehen skizziert Ihnen den ungefähren Ablauf der Projektarbeit. Sie dürfen dieses Vorgehen auch für Ihre Zwecke anpassen. Der Ablauf soll Ihnen lediglich einen möglichen sinnvollen Weg bezüglich des Vorgehens skizzieren:

1. Entscheiden Sie sich innerhalb der Gruppe, welche Aufgaben Ihr verteiltes System erfüllen soll und notieren Sie sich die Anforderungen an das verteilte System und entscheiden Sie, wie Sie die Aufteilung machen würden. Überlegen Sie auch,

welche Massnahmen Sie für die Hochverfügbarkeit umsetzen und wie Sie da die Verantwortlichkeiten innerhalb des Teams verteilen werden.

2. Sprechen Sie mit der Lehrperson über Ihre Idee und die Planung des Aufbaus des verteilten Systems. **Die Lehrperson muss Ihre Planung freigeben.** Die Freigabe durch die Lehrperson soll sicherstellen, dass Ihre Planung realistisch ist und den Anforderungen im Auftrag gerecht wird.
3. Planen Sie gemeinsam die Schnittstellen - wie die einzelnen Systemkomponenten miteinander interagieren sollen und wie die Massnahmen für die Hochverfügbarkeit zusammenspielen sollen. **Tipp:** Nehmen Sie sich bei diesem Schritt genügend Zeit, um alles genau durchzudenken.
4. Testen Sie allfällige Konzepte, mit denen Sie die Hochverfügbarkeit umsetzen möchten, auf deren Tauglichkeit. Wenn Sie das bereits jetzt tun, weiss jeder bei der Entwicklung seiner Teile, auf was zu achten ist, dass die HA-Massnahmen dann auch so funktionieren wie angedacht.
5. Jeder setzt nun die Systemkomponenten um, für die er die Verantwortung trägt. Dabei sind Sie Ihr eigener Herr und dürfen **frei und unabhängig** von den anderen Gruppenmitgliedern entscheiden, **mit welchen Technologien** Sie Ihre Systemkomponente umsetzen möchten - wichtig ist lediglich, dass die vorher definierte Schnittstelle eingehalten wird. Sollten dennoch Anpassungen an der Schnittstelle nötig werden, müssen Sie sich mit allen Gruppenmitgliedern abstimmen, die Ihre Schnittstelle verwenden. **Tipp:** Dokumentieren Sie gewisse Dinge parallel zur Entwicklung. Das erleichtert die Fertigstellung der Dokumentation Ihrer Systemkomponente vor der finalen Abgabe.
6. Wenn alle Gruppenmitglieder Ihre eigene Systemkomponente fertig entwickelt haben, sollten Sie die Verbindung zu den anderen Komponenten im System und das Gesamtsystem testen.
7. Setzen Sie nun die zuvor angedachten Hochverfügbarkeitsmassnahmen um.
8. Allfällig entdeckte Probleme beim Testing sollten Sie nun noch beheben.
9. Finalisieren Sie die Dokumentation und geben Sie Code und Dokumentation der Lehrperson wie gefordert ab.

## Abgabe - erwartete Inhalte

Als Abgabe werden folgende Teile erwartet:

- **Dokumentation:** Die Dokumentation soll Ihr Vorgehen sowie Ihr verteiltes System (die einzelnen Teile und wie diese zusammenhängen) beschreiben. Folgende Inhalte sollten vorhanden sein:
  - Titelblatt
  - Inhaltsverzeichnis
  - Vorgehen (Projektplanung)
  - Anforderungen an das verteilte System, Features des verteilten Systems
  - Verteiltes System im Überblick (inkl. Zusammenhänge zwischen den einzelnen Systemkomponenten, Tipp: grafische Darstellung)
  - Dokumentation der einzelnen Systemkomponenten (jeweils durch die, für die Systemkomponente verantwortliche Person, zu erstellen)
  - Abschliessende Worte (Fazit, Ausblick oder ähnliches)

- gegebenenfalls Anhang (beispielsweise mit Quellenangaben, Testprotokollen vom Testing oder ähnliches)
- **Code:** als Link zum GIT-Repository oder als ZIP-Archiv in der Abgabe. Der Sourcecode der einzelnen Systemkomponenten dürfen in getrennten Repositories oder in einem gemeinsamen Repository abgegeben werden (was Ihnen lieber ist). Wenn Sie mit Branches im GIT-Repository arbeiten, lassen Sie die Lehrperson bitte wissen, welches der Branch ist, der den Finalen stand des Projekts enthält.

## Bewertungsraster

Sie erhalten gemäss untenstehender Tabelle für verschiedene Kriterien Punkte. Ihre erreichte Note berechnet sich anschliessend nach der Formel  $[\text{erreichte Punkte}] / [\text{maximal mögliche Punkte}] * 5 + 1$ . Bei einigen Kriterien erhalten alle Gruppenmitglieder dieselben Punkte - bei anderen Kriterien ist die Punktevergabe für jede Person individuell.

Note wird bei 5.5 gecapped, um die Maximalnote von 6 zu erreichen werden mindestens 3 Punkte bei den [optionalen Inhalten](#) erwartet. Sie können die optionalen Inhalte auch zur Kompensation des Hauptteils verwenden.

Was bei dem jeweiligen Kriterium gilt entnehmen Sie der Spalte **Art der Bewertung**. Je nach Situation kann es auch sinnvoll sein, bei einem "Gruppenkriterium" Einzelbewertungen zu vergeben oder bei "Einzelkriterien" eine Gruppenbewertung. Sollte die Art der Bewertung angepasst werden, wird die Lehrperson diesbezüglich mit Ihnen Rücksprache nehmen und erläutern, wieso die Anpassung nötig ist und gegebenenfalls auch Ihre Meinung dazu abholen.

In der Spalte **Punktevergabe** finden Sie Beispiele, was wie viele Punkte gibt.

Kriterium	Punktevergabe	Art der Bewertung
<b>Formale Vorgaben (max. 3 Punkte)</b>	<p>3 P.: Alle geforderten Inhalte vorhanden (Siehe <a href="#">Abgabe - erwartete Inhalte</a>)</p> <p>2 P.: Kleinigkeiten nicht ganz vollständig bzw. mussten nachgefordert werden</p> <p>1 P.: Wichtige Inhalte fehlen in der Abgabe bzw. wurden auf Rückfrage nicht nachgeliefert</p> <p>0 P.: Abgabe fehlt</p>	Gruppe
<b>Gestaltung Dokumentation (max. 2 Punkte)</b>	2 P.: Dokumentation ist einheitlich gestaltet	Gruppe

Kriterium	Punktevergabe	Art der Bewertung
	<p>1 P.: Dokumentation keine Einheit - eher viele "Einzelteile" die nicht aufeinander abgestimmt sind</p> <p>0 P.: Abgabe fehlt</p>	
<b>Inhalt Anforderungen / Features (max. 3 Punkte)</b>	<p>3 P.: Anforderungen &amp; Features des Systems in sehr guter Qualität beschrieben (funktionale wie auch nicht-funktionale Anforderungen liegen vor, eine Priorisierung der Anforderungen wurde vorgenommen - z. B. zwingende und optionale Anforderungen)</p> <p>2 P.: Anforderungen &amp; Features des Systems in guter Qualität beschrieben</p> <p>1 P.: Klare Mängel bei der Beschreibung von Anforderungen / Features</p> <p>0 P.: Definition / Beschreibung fehlt</p>	Gruppe
<b>Inhalt Überblick verteiltes System (max. 3 Punkte)</b>	<p>3 P.: In sehr guter Qualität (inkl. grafischer Darstellung der Zusammenhänge) vorhanden</p> <p>2 P.: In guter Qualität vorhanden (grafische Darstellung fehlt/unklar)</p> <p>1 P.: Qualität weist klare Mängel auf</p> <p>0 P.: Überblick fehlt</p>	Gruppe
<b>Beschreibung Systemkomponenten (max. 9 Punkte)</b>	<p>9 P.: Beschreibung klar und verständlich in sehr guter Qualität</p> <p>6 P.: Beschreibung in guter Qualität vorhanden</p> <p>3 P.: Beschreibung weist klare Mängel auf</p> <p>0 P.: Beschreibung fehlt</p>	Einzel
<b>Code Qualität (max. 3 Punkte)</b>	<p>3 P.: Code überall einheitlich strukturiert und wenn nötig kommentiert, unnötiger</p>	Einzel

Kriterium	Punktevergabe	Art der Bewertung
	<p>Code (auskommentierte Codeteile, nicht verwendetes) wurden bereinigt</p> <p>2 P.: Code meistens einheitlich strukturiert und wenn nötig kommentiert</p> <p>1 P.: Klare Mängel bei Code Qualität vorhanden</p> <p>0 P.: Abgabe fehlt</p>	
<b>Qualität der umgesetzten Systemkomponenten (max. 9 Punkte)</b>	<p>9 P.: Umsetzung vollumfänglich funktional, State of the Art bei Umsetzung eingehalten</p> <p>6 P.: Umsetzung grundsätzlich funktional</p> <p>3 P.: Klare Mängel bei der Umsetzung vorhanden</p> <p>0 P.: Abgabe fehlt</p> <p>Hinweis: In diesem Kriterium wird nur bewertet ob Ihre Systemkomponente alles richtig macht. Ob die anderen Systemkomponenten Ihre Systemkomponente richtig verwenden wird hier nicht beurteilt.</p>	Einzel
<b>Funktionalität / Umfang (max. 9 Punkte)</b>	<p>9 P.: Sehr komplexe und / oder umfangreiche Systemkomponente(n) umgesetzt, Umsetzung funktional</p> <p>6 P.: Umfang im erwarteten Rahmen, kleinere Mängel bei der Funktionalität</p> <p>3 P.: Umfang sehr gering, klare Mängel bei der Funktionalität</p> <p>0 P.: Abgabe fehlt</p>	Einzel
<b>HA (high availability) (max. 6 Punkte)</b>	<p>6 P.: Praktisch alle Teile des verteilten Systems sind hochverfügbar (mind. 2 Massnahmen sind funktional umgesetzt)</p>	Gruppe

Kriterium	Punktevergabe	Art der Bewertung
	<p>4 P.: Massnahmen für eine Hochverfügbarkeit wurden angedacht und umgesetzt, sind jedoch nur teilweise funktional</p> <p>2 P.: Massnahmen für Hochverfügbarkeit weisen klare Mängel auf, sind aber grundsätzlich angedacht / vorhanden</p> <p>0 P.: Es wurden keine Massnahmen für Hochverfügbarkeit umgesetzt</p>	
<b>Demonstration funktionierendes verteiltes System (max. 6 Punkte)</b>	<p>6 P.: Keine "Pannen" bei der Demonstration, Ablauf der Demonstration sinnvoll gewählt</p> <p>4 P.: Kleinere Pannen bei der Demonstration (die aber erklärt / korrigiert werden können)</p> <p>2 P.: Grössere Pannen, keine Erklärung wieso (od. Erklärung die nicht plausibel / korrekt ist)</p> <p>0 P.: Demonstration nicht gemacht bzw. ein Zeigen nicht möglich (weil das verteilte System nicht vorhanden ist)</p>	Gruppe

Maximal können 53 Punkte erreicht werden.

## Abgabe - optionale Inhalte

Erwartet wird, dass Sie das Testing Ihres Codes parallel zur Entwicklung Ihrer Systemkomponente tun. Wenn Sie das Testing sauber planen (Testcases definieren) und dokumentieren (Testprotokoll schreiben) und / oder automatisiertes Testing (beispielsweise Unit-Testing, GUI-Testing, Performance-Tests oder ähnliches) verwenden, um die Qualität Ihrer Systemkomponente etwas systematischer zu überprüfen, dann können Sie dadurch Extrapunkte sammeln. Diese Extrapunkte zählen nicht zu der maximal möglichen Punktezahl und erlauben es Ihnen ein gewisses "Polster" aufzubauen, falls Sie bei gewissen Punkten im regulären Bewertungsraster schlechter abschneiden sollten wie erhofft.

Kriterium	Punktevergabe	Art der Bewertung
<b>Testdefinition (max. 2 Punkte)</b>	2 P.: Beschreibung von Testcases und Testumgebung in guter bis sehr guter Qualität vorhanden  1 P.: Beschreibung von Testcases und Testumgebung vorhanden  0 P.: Keine Testcases bzw. Testumgebung definiert	Einzel
<b>Testprotokoll (max. 2 Punkte)</b>	2 P.: Testprotokoll in guter bis sehr guter Qualität vorhanden  1 P.: Testprotokoll vorhanden  0 P.: Keine Testprotokoll vorhanden	Einzel
<b>Automatisiertes Testing (max. 3 Punkte)</b>	3 P.: Automatisierte Tests in guter bis sehr guter Qualität mit hoher Testabdeckung vorhanden  2 P.: Automatisierte Tests vorhanden, Testabdeckung ok  1 P.: Automatisierte Tests vereinzelt vorhanden  0 P.: Keine automatisierten Tests vorhanden	Einzel