

Klein, aber fein: Ad-hoc Lösungen zeigen im Flipped Classroom

Christian Kohls¹ und Dennis Dubbert¹

Abstract: Dieser Beitrag beschreibt die Herausforderungen beim Vergleichen verschiedener studentischer Beiträge während eines Flipped Classrooms. Als Lösungsansatz wird ein Werkzeug zum ad-hoc Hochladen von Bildern vorgestellt. Mit dem Werkzeug können Studierende ihre Lösungen und Beiträge senden, ohne sich einzuloggen oder eine App zu installieren.

Keywords: Flipped Classroom, Kooperation

1 Kontext: Flipped Classroom und Live-Lösungen

Der Begriff „Flipped Classroom“ meint meist vor allem die Umkehrung der Lernaktivitäten während der Präsenz- und Selbstlernphasen. Zentrales Ziel des Flipped Classrooms ist eine sinnvollere Nutzung der wertvollen Präsenzzeit und eine Aktivierung der Studierenden zur intensiven Teilnahme. Dabei erarbeiten die Studierenden häufig alleine oder in kleinen Gruppen verschiedene Lösungsansätze. Ein zentrales Problem stellt die einfache und schnelle Präsentation von Lösungen dar. Bislang können aus organisatorischen Gründen nur einzelne Lösungsvorschläge präsentiert werden. Aus der Evaluation unserer Lehrveranstaltung im Bereich „Algorithmen und Programmierung“ geht hervor, dass sich Studierende explizit wünschen, dass alternative Lösungen anderer Studierender besprochen werden. Voraussetzung hierfür ist jedoch, dass diese schnell und unkompliziert geteilt werden können. Es gibt bereits verschiedene digitale Systeme, mit denen sich Antworten und Lösungen sammeln lassen (z. B. Audience-Response-Systeme wie PINGO, File-Sharing-Dienste wie DropBox oder ownCloud, virtuelle Pinnwände wie Patlet oder Linoit, kollaborative Werkzeuge wie Baiboard oder Popplet). Allerdings ergeben sich in der Praxis oft Probleme, die den reibungslosen Ablauf der Veranstaltung erheblich beeinträchtigen:

- Proprietäre Lösungen erfordern ein Setup und Login. Dies führt zu Zeitverlust und Stress während der Präsenzveranstaltung.
- Studierende werden ausgeschlossen, wenn sie nicht auf der Plattform sind oder kein passendes Gerät haben.
- Oft fehlen adäquate Moderationsmöglichkeiten (kein gezieltes Vergleichen von Lösungen oder Hervorheben von Aspekten).

¹ TH Köln, Institut für Informatik, Campus Gummersbach, Steinmüllerallee 1, 51643 Gummersbach,
{vorname.nachname}@th-koeln.de

- Audience-Resonse-Systeme (z. B. PINGO) eignen sich gut für einfache Antwortformen (Multiple Choice, Begriffe sammeln), aber nicht für komplexere Lösungen, wie sie in der Informatik üblich sind, z. B. Strukturdiagramme, Codeabschnitte, Skizzen.

2 Werkzeug-Vorstellung

Um diese Probleme zu umgehen haben wir ein webbasiertes Werkzeug entwickelt, mit dem per QR-Code eine Verbindung zu einer Session aufgebaut wird. Falls kein QR-Code-Reader verfügbar ist, kann auch eine kurze URL eingegeben werden. Ein Login ist nicht erforderlich. Die Zuordnung geschieht über eine eindeutige Session-Id, die in der URL bzw. dem QR-Code enthalten ist. Bilder können sofort hochgeladen werden. Sie erscheinen als kleine Vorschaubilder. Als Dozierender kann man nun einzelne Bilder auswählen und für alle präsentieren. Zudem können zwei Bilder gleichzeitig ausgewählt und miteinander verglichen werden.

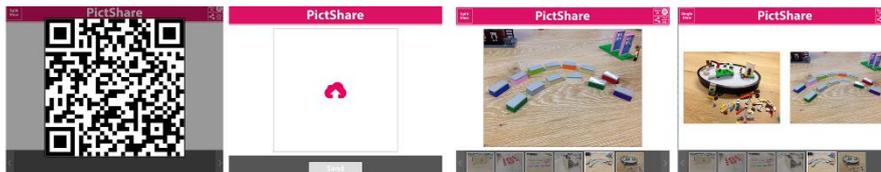


Abb. 1: Bilder ad-hoc hochladen und Ergebnisse vergleichen

Neben dem Einsatz in Vorlesung und Seminaren wird das Werkzeug bereits in Designräumen eingesetzt, um den Entwicklungsprozess auf einem interaktiven Whiteboard zu dokumentieren. Die Bilder werden vom Smartphone ans interaktive Whiteboard gesendet, ohne dass eine App dafür installiert werden muss. Das Werkzeug ist für viele Fachdisziplinen geeignet, z. B. zur Präsentation von Pseudocode, Versuchsaufbauten, Rechercheergebnissen, Architekturen oder Design-Skizzen.

3 Technische Lösung

Serverseitig wird ein Node.js Server eingesetzt. Bilder werden während der Session in einer MongoDB Datenbank gespeichert, damit weitere Clients diese anzeigen können, z. B. auf weiteren Displays. Sobald eine Session beendet wird, werden alle Bilder gelöscht. Eine Session ist dann beendet, wenn kein Client mehr mit dem Server verbunden ist. Clientseitig werden HTML5 und JavaScript eingesetzt. Die Kommunikation geschieht über WebSockets. Die Architektur folgt dem Publish-Subscribe-Muster, d. h. alle angemeldeten Clients werden über Änderungen sofort informiert. Zukünftige Versionen werden eine clientseitige Verkleinerung des Bildes vor dem Senden an den Server durchführen, um die erforderliche Bandbreite zu reduzieren.