

Dokumentation Firmenprojekt im SRZ: Intectainment - communication is key



- **Schüler:** Bruno Hoffmann, Karl Jahn, Töm Nitsche & Jakob Paridon
- **Kurs:** [Informatik III/IV 4 SRZ \(https://tu-dresden.de/ing/informatik/srz\)](https://tu-dresden.de/ing/informatik/srz)
- **Firma:** [intecsoft group \(https://www.intecsoft.de/\)](https://www.intecsoft.de/)
- **Firmenbetreuer:** Claus Jungmann, Eugen Gorning
- **AG-Leiter:** Tassilo Tanneberger

Ziel des Projektes

Intectainment ist ein Infotainment-System, das, ähnlich zu anderen sozialen Netzwerken, Kommunikation und das Veröffentlichen von Inhalten ermöglicht.

Dazu kann jeder Nutzer einen Kanal (ggf. mit Beschreibung) erstellen, auf dem er Informationen herausgibt. Diese Kanäle können von anderen Usern abonniert werden, welche sich die entsprechenden Zusammenhänge auf ihren Startseiten anzeigen lassen können.

Jede Startseite besteht aus einzelnen Kacheln, welche Posts abonniert Kanäle anzeigen; somit ist auch jede Startseite individuell und für den entsprechenden Nutzer angepasst.

Alle Posts eines jeden Kanals sind ähnlich strukturiert: mithilfe der [Markdown-Syntax \(https://www.markdownguide.org/basic-syntax\)](https://www.markdownguide.org/basic-syntax) lassen sich einfache und übersichtliche Texte schreiben, in denen sich auch Formatierungen, Links, Bilder usw. einbinden lassen.

Weitere Funktionen umfassen das Merken und spätere Abrufen von Posts sowie das Redigieren (insb. durch Löschen) von Posts durch eine Redaktion.

Intectainment soll den Austausch zwischen Mitarbeitern in einer Firma unterstützen und verbessern.

Deployment

Anforderungen

Intectainment benötigt eine SQL-Engine (z. B. sqlite3 oder MySQL) sowie eine Python Runtime (getestet auf Python 3.10).

Installation der benötigten Python Frameworks und Bibliotheken:

```
pip install -r requirements.txt
```

siehe auch Umsetzung - Libraries und Frameworks

Erste Ausführung/Initialisierung:

```
python run.py --init
```

Dadurch wird eine Konfigurationsdatei `Intectainment/config.conf` erstellt.

In dieser sollte wenigstens der Database-URI Parameter angepasst werden.

Wurde die Config angepasst, kann durch das Pressen einer beliebigen Taste die Initialisierung fortgesetzt werden.

```
[Server]
server = localhost
port = 3000
secretKey = [OWN KEY]

[Database]
URI = [YOUR DATABASEURI]
```

Standard-Nutzer

Beim Initialisieren wird ein Nutzer Admin mit dem Passwort `intectainment` hinzugefügt, welcher Administrationsberechtigungen hat. Es wird empfohlen das Kennwort und ggf. auch den Nutzernamen später zu ändern.

Database-URI-Schemata:

- MySQL: `mysql+pymysql://[user]:[password]@[host]/[database]`
- sqlite3: `sqlite:///path/[file].db`

Je nach verwendeter Datenbank-Engine können für die Installation zusätzliche Module nötig sein.

Einfügung eines Testdatensatzes

Im Administrationsinterface, welches sich über die Home-Leiste erreichen lässt, gibt es unter dem Reiter "Starthilfe" die Möglichkeit, einen Standard-Datensatz einzufügen, welcher einige Kanäle, Kategorien, Posts und Nutzer beinhaltet.

Diese Option bietet sich an, wenn Sie Ihr lokales System auf Funktionstüchtigkeit überprüfen wollen.

Starten des Servers

```
python run.py
```

Umsetzung

Libraries und Frameworks

Die Backend-Verwaltung wird mithilfe des Frameworks [Flask](https://flask.palletsprojects.com/en/2.0.x/)

(<https://flask.palletsprojects.com/en/2.0.x/>) und dessen Erweiterung [Flask-SQLAlchemy](https://flask-sqlalchemy.palletsprojects.com/en/2.x/)

(<https://flask-sqlalchemy.palletsprojects.com/en/2.x/>) gestaltet.

Flask ermöglicht das dynamische Erstellen von Websites und das Abrufen/Eintragen von

Informationen aus bzw. in einer Datenbank mit Nutzung von Python-Code.

Wir haben uns für die Nutzung des Frameworks entschieden, da es das Anzeigen von gleichförmigen, aber unterschiedlichen Daten (also z. B. Posts, Kanäle und Dashboards, welche alle der gleichen Struktur folgen, jedoch trotzdem verschiedene Eigenschaften haben) deutlich vereinfacht.

Zum Verschlüsseln von Passwörtern wurde die [bcrypt-Library \(https://pypi.org/project/bcrypt/\)](https://pypi.org/project/bcrypt/) genutzt.

Das Frontend basiert hauptsächlich auf dem Framework [Bootstrap 5 \(https://getbootstrap.com/docs/5.0/getting-started/introduction/\)](https://getbootstrap.com/docs/5.0/getting-started/introduction/).

Die Übersetzung von Markdown in HTML wurde mit dem Converter [Showdown \(http://showdownjs.com/\)](http://showdownjs.com/) umgesetzt.

Wir haben uns für einen in Javascript geschriebenen Converter entschieden, da der Server somit nur für das Speichern und Abrufen von Posts zuständig ist und die Markdown-zu-HTML-Konvertierung auf den Client ausgelagert wird. Dieser Aspekt ist besonders wichtig, da das Ergebnis der Konvertierung bereits beim Schreiben von Posts angezeigt werden soll.

Die Nutzung von Showdown soll nicht nur dabei helfen, den Server zu entlasten, sondern auch dabei, die Latenz zu verringern.

Datenbankarchitektur

Die wichtigsten Daten von Intectainment, welche die Grundlage für das System bilden, sind unten dargestellt.

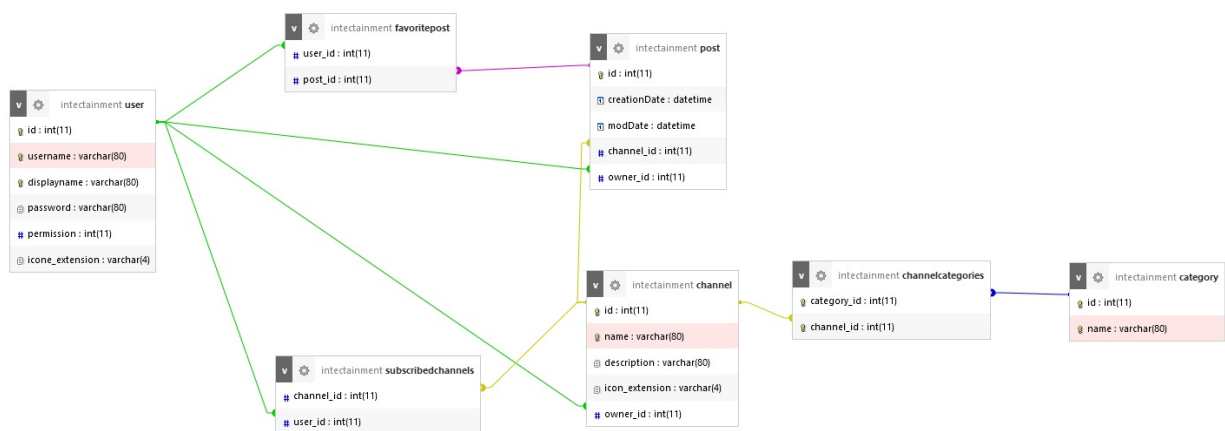


Abb. 1: Vereinfachtes Datenbankmodell von Intectainment

Alle essentiellen Funktionen - Nutzerprofile, Kanäle und Posts sowie deren Erweiterungen (namentlich Abonnements, Kategorien und das Markieren von Veröffentlichungen) - werden so wie hier dargestellt realisiert.

Die Darstellung umfasst jedoch nur die Grundlagen des Backend-Systems und besitzt somit keinen Anspruch auf Vollständigkeit.

Wichtige Abläufe

Erstellung von Nutzern

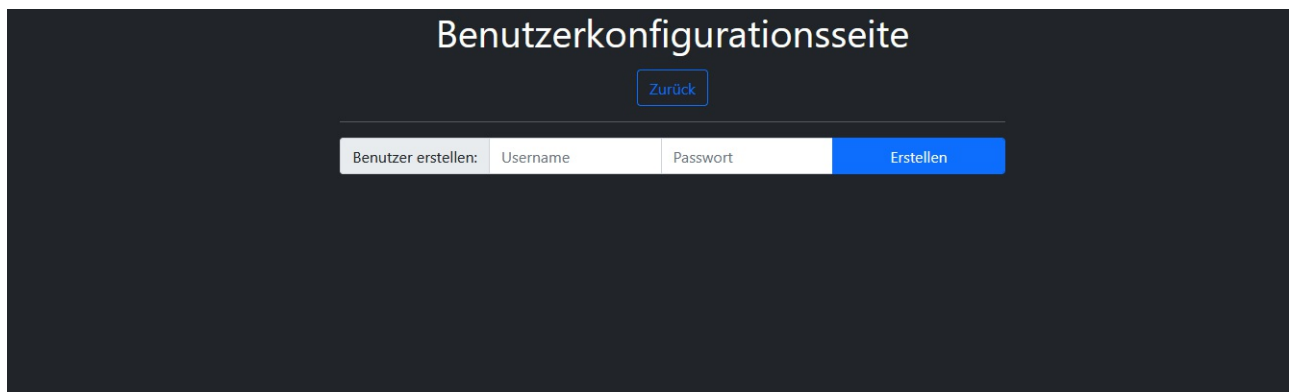


Abb. 2: Erstellen von Nutzern mithilfe des Admin-Interfaces

Die verschiedenen User, die die lokale Instanz von Intectainment nutzen, müssen von dem Administrator des Netzwerk in einem eigenen Interface erstellt werden.

Es gibt keine Möglichkeit, nicht vom Administrator genehmigte Accounts zu erstellen. Auch wenn durch diese Regelung die erste Einrichtung des Systems etwas anspruchsvoller macht, kann so verhindert werden, dass Unbefugte Zugriff auf das Netzwerk erlangen.

Dem Nutzer ist ein Berechtigungslevel in Form einer Zahl zugeordnet. 0 - Guest, 10 - User, 100 - Moderator, 255 - Admin. Dieser Parameter entscheidet über die Zugriffsrechte des Nutzers und kann in der Nutzereinstellung verändert werden.

Erstellen von Kanälen

Jeder Nutzer kann Kanäle erstellen, die gemeinschaftlich von allen Mitarbeitern des Unternehmens genutzt und betrieben werden. Den Kanälen kann bei ihrer Erstellung eine Kategorie zugewiesen werden (z. B. C++, IoT, Web-Development usw.), die die Suche nach und das Entdecken von Kanälen erleichtert.

Wie bereits erwähnt, werden alle Kanäle nicht von einzelnen Usern, sondern von der gesamten Mitarbeiterschaft betrieben. Das bedeutet, dass jeder angemeldete Nutzer auf einem beliebigen Kanal Posts verfassen und veröffentlichen kann. Dies soll dabei helfen, dass jeder Nutzer Texte zu für ihn interessanten Themen verfassen kann und keine Konkurrenz zwischen verschiedenen Kanälen zum gleichen Thema entsteht.

Kanalansicht und -navigation

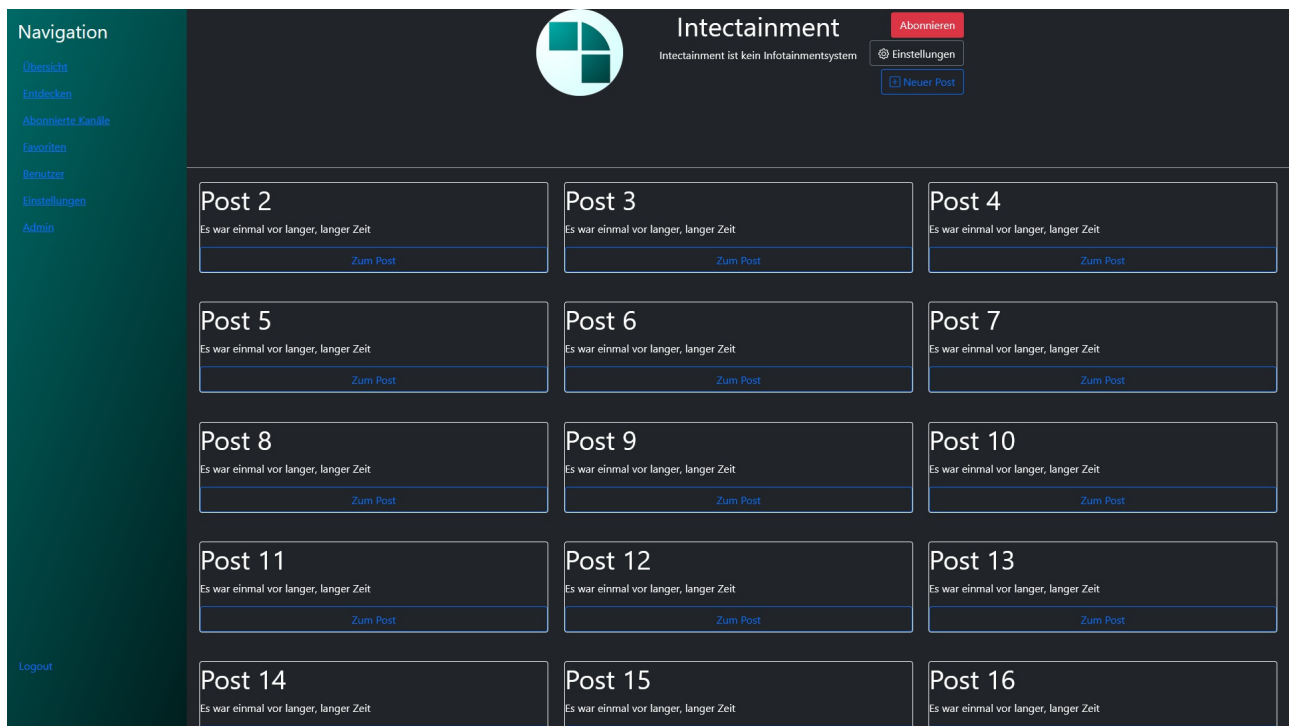


Abb. 3: Beispielsicht eines Kanals inklusive der neuesten Posts

Wählt man einen Kanal aus, wird eine Vorschau der aktuellsten Posts angezeigt, welche die ersten Zeilen des Textes umfasst. Über die Schaltfläche "Zum Post" lässt sich der Text in voller Länge aufrufen. Dort kann man diesen auch als Favoriten markieren und ggf. bearbeiten.

Als angemeldeter Nutzer kann man Kanäle abonnieren bzw. deabonnieren, aber auch Posts erstellen und auf dem entsprechenden Kanal veröffentlichen. Ist man der Ersteller des Kanals, kann man auch die Einstellungen (z. B. Beschreibung, Kategorie und Kanalbild) anpassen.

Erstellen und Bearbeiten von Posts

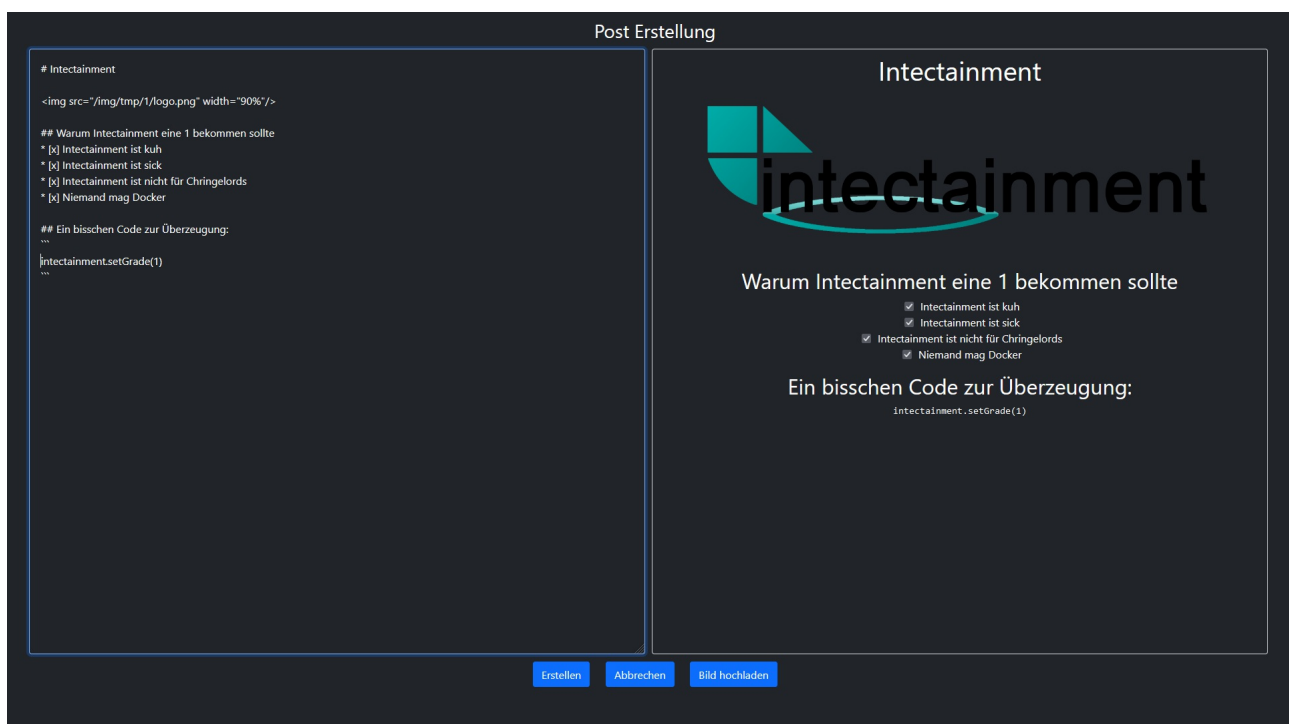


Abb. 4: Beispielsicht eines Posts während der Erstellung

Nach der Auswahl des Kanals, auf welchem der entsprechende Post erscheinen soll, kann der Nutzer mithilfe der Markdown-Syntax einen Text verfassen, welcher in Echtzeit zu HTML konvertiert und neben dem Eingabefeld angezeigt wird.

Für das Hochladen von Bildern gibt es eine eigene Schaltfläche; mit dieser kann man Bilder auf dem genutzten Server hochladen und über einen eigens dafür generierten Link im Text einbinden.

Alternativ ist auch das Einbinden von Bildern aus externen Quellen möglich.

Bei der Veröffentlichung wird der Post als Markdown-Datei auf dem Server gespeichert. In der Datenbank wird zudem eine Verlinkung zu der Datei eingetragen, welche zusätzliche Informationen wie das Datum der Erstellung, den Kanal und das Datum der letzten Bearbeitung speichert.

Auch nach der Veröffentlichung kann der User, welcher den Post erstellt hat, diesen in beliebigem Maße bearbeiten bzw. löschen.

Redaktion und Moderation

Um die Verwaltung des Systems kümmert sich mit den Administratoren auch eine Redaktion. Diese ist die einzige Instanz, welche - neben dem entsprechenden Nutzer - Posts und Kanäle bearbeiten kann.

Probleme

Das Projekt weist keine schwerwiegenden Probleme auf. Auch wenn die grundlegende Funktionalität gewährleistet ist, konnten nicht alle im Pflichtenheft festgeschriebenen Features umgesetzt werden.

Insbesondere die Erstellung individueller Dashboards mit zugeordneten Kacheln wurde nicht realisiert; da diese Funktion für Nutzer einen hohen Zeit- und Arbeitsaufwand bedeutet hätte, haben wir uns dafür entschieden, statt der Inhalte festgelegter Kanäle die jeweils neuesten Post aller abonnierten Kanäle anzuzeigen.

Auch Nutzerprofile bzw. Kanalbeschreibungen konnten nicht umgesetzt werden.

Erweiterungsmöglichkeiten

Intertainment stellt als soziale Plattform ein Projekt mit vielen verschiedenen

Erweiterungsmöglichkeiten dar, weswegen im folgenden Abschnitt nur einige, besonders bedeutsame beleuchtet werden können.

Nutzerprofile

Nutzerprofile enthalten momentan nur die essentielle Information des Namen des entsprechenden Users. Eine mögliche Erweiterung wäre die Personalisierung von Profilen durch Beschreibungen, Verlinkungen zu anderen Profilen und ähnlichem.

Nutzung auf Displays

Als Sammelpunkt für Informationen über und Geschehnisse im Unternehmen wäre auch die Nutzung von Intertainment auf Bildschirmen (z. B. im Eingangsbereich oder auf den Gängen der jeweiligen Firma) ohne Nutzerinput denkbar. Ein eigens für diese Aufgabe eingerichtetes Dashboard würde die aktuell wichtigsten Informationen bereitstellen.

Diese Bildschirme könnten für verschiedene Nutzergruppen angepasst werden: während neue Kunden sich einen Überblick über das Unternehmen verschaffen wollen, sind für Entwickler

Updates von Projekten oder Änderungen im Zeitplan eher von Bedeutung.
Die Positionierung der Displays im Gebäude wäre dann von der entsprechenden Zielgruppe abhängig.

Nutzung externer Ressourcen

Neben den von Usern geschriebenen Posts wäre auch die Nutzung externer Quellen möglich: statt für jedes Update einen Post zu verfassen, könnten Kanäle auch die Möglichkeit haben, von z. B. GitHub bereitgestellte Informationen (im Bezug auf die Aktivität, die letzten Updates, die mitarbeitenden Personen usw.) zu nutzen.

Die nutzbaren Ressourcen hängen hierbei vom jeweiligen Unternehmen ab: wenn vorhanden, ist die Nutzung von Social-Media-Kanälen möglich; auch eine Anbindung an die offizielle Website der Firma wäre denkbar.