
# OnLaunch

Endbericht | Call 17 | Projekt ID 6448

# Inhalt

# 1 Einleitung

Wir von kula haben die Mission Mobile App Entwicklung zugänglich und leistbar zu machen. Dafür bauen wir Tools wie OnLaunch. OnLaunch ist eine In-App-Message-Display-Plattform, die darauf abzielt, das Nutzererlebnis bei mobilen Apps zu verbessern, indem sie zielgerichtete Nachrichten (Messages) beim Start der App ermöglicht. Sie befähigt mobile App-Entwickler:innen und nicht-technische Personen, benutzerdefinierte Messages für ihre Nutzer:innen zu erstellen und zu verwalten, um eine optimale Nutzerbindung und -zufriedenheit zu gewährleisten.

# 2 Projektbeschreibung

Wir haben folgende Eckpunkte, die OnLaunch ausmachen, identifiziert:

- **Unterstützung verschiedener Message-Typen:** OnLaunch unterstützt eine Vielzahl von Message-Typen, die für unterschiedliche Anwendungsfälle konfiguriert werden können. Dies beinhaltet blockierende Messages, die es Entwickler:innen erlauben, Messages Nutzer:innen anzuzeigen, die die Funktionalität der mobilen App bei Bedarf einschränken können. Beispielsweise ist dies bei einem Wartungsmodus gewünscht, um Nutzer:innen ein gutes Nutzererlebnis zu gewährleisten.
- **Admin API:** Eine zentrale Komponente von OnLaunch, die es Entwickler:innen ermöglicht, OnLaunch in eigene Applikationen zu integrieren und somit Prozesse, wie beispielsweise das Erstellen und Bearbeiten von Messages, automatisch ausführen zu können. Die Admin API erlaubt eine einfache und sichere Konfiguration. Diese API ist speziell darauf ausgelegt, den Integrationsaufwand für Entwickler:innen zu minimieren und gleichzeitig eine hohe Leistung und Zuverlässigkeit zu gewährleisten.
- **Erweiterbarkeit und Flexibilität:** Die Plattform ist so gestaltet, dass sie leicht an die sich ändernden Bedürfnisse und Anforderungen der Developer-Community angepasst werden kann. Dies ermöglicht eine kontinuierliche Erweiterung der Funktionalitäten und die Integration neuer Features, die zur Effizienzsteigerung und Verbesserung des Nutzererlebnisses beitragen.

- **Community-Driven Development:** Als Open-Source-Projekt legt OnLaunch großen Wert auf die aktive Beteiligung und den Beitrag der Entwicklergemeinschaft. Durch die Zusammenarbeit mit Entwicklern aus aller Welt fördert OnLaunch Innovation und Vielfalt in der Entwicklung neuer Lösungen.

# 3 Verlauf der Arbeitspakete

## 3.1 Arbeitspaket 1 - <Planung und Projektstart>

Dieses Arbeitspaket markiert den Grundstein unserer Förderungsperiode. Die Tätigkeiten umfassen unter anderem das unterzeichnen von Verträgen, Aufstellen eines Detailprojektplans und dessen Überlieferung und Abnahme. Außerdem gehört zu diesem Meilenstein das Ausarbeiten einer detaillierten Liste der Projektergebnisse, sowie das Erstellen eines ersten Blogposts und das Beantragen der ersten Förderrate.

Bereits hier hatten wir gemerkt, dass es von der ursprünglichen Planung beim Einreichen bzw. Bewerbung für die Förderung Abweichungen entstanden sind. Das hat beispielsweise die Ursache, dass sich bei uns innerhalb des Unternehmens zu Veränderungen gekommen ist. Geändert haben sich so die Verantwortungen einzelner Teammitglieder, das Ausmaß der Anstellungen von Mitarbeiter:innen, sowie das Erhalten eines Investments. Auch hatten wir neue Informationen und Erkenntnisse über unsere Zielgruppe und deren Anforderungen, die unsere Planung beeinflusst haben.

In unserer Blog-Post-Serie auf der netidee-Seite haben wir verschiedene Einträge verfasst. Unter anderem beschreiben wir was Onlaunch ist und welche unterschiedlichen Anwendungsfälle unterstützt werden sollen.

## 3.2 Arbeitspaket 2 - <vorläufiges Server-Softwareskelett>

Hierbei handelt es sich um die Entwicklung einer vorläufigen Server-Software, die einen Prototypen von OnLaunch darstellt. Auf simple Weise ermöglicht dieser Prototyp es, Messages abfragen zu können. Die Fertigstellung dieses Pakets ermöglichte uns die Arbeitspakete 3, 4 & 5 anzufangen.

Es gab keine großen Überraschungen oder Abweichungen bei diesem Paket. Es war auch größtenteils Florians Aufgabe, war gut geplant und konnte zur völligen Zufriedenheit umgesetzt werden.

### 3.3 Arbeitspaket 3 - *<iOS Modul>*

Die Entwicklung dieses Pakets gestaltete sich sehr unkompliziert. Es wurde ein iOS Modul aufgesetzt, welches die Messages vom Server abruft, entsprechend verarbeitet und dem/der App-NutzerIn anzeigt.

Recht früh zeichnete sich ab, dass wir wesentlich mehr Zeit eingeplant hatten für dieses Paket, als wir tatsächlich benötigten. Der Grund dafür war unsere äußerst effiziente Arbeitsweise. Vor allem von Philip, der die erforderliche Arbeit in Kürze und wesentlich schneller als geplant fertig gestellt hat.

### 3.4 Arbeitspaket 4 - *<Android Modul>*

Ebenso wie bei dem iOS-Modul – ein Android Modul welche Messages vom Server abfragt und entsprechend verarbeitet. Es ist hier möglich, dass Entwickler:innen das Modul in ihre App implementieren können und damit Messages vom OnLaunch Server abfragen können.

Auch hier waren wir vor dem ursprünglich geplanten Arbeitsaufwand früher fertig. Dies ist auf Damian zurückzuführen, der gar nur wenige Stunden für die Implementierung benötigt hatte – wesentlich weniger als ursprünglich geplant war.

### 3.5 Arbeitspaket 5 - <Server mit GUI und Login>

Bei diesem Arbeitspaket handelte es sich mit Abstand um das Größte und Aufwändigste. Der Grund dafür sind die komplexe Anforderung und die umfangreichen Funktionalitäten, die dieser Server zu erfüllen hat.

Zwei der wichtigsten Funktionen sind das Erstellen und das Verwalten der Messages. Je mehr wir uns mit potenziellen Nutzer:innen austauschten, desto mehr mögliche Funktionen und Detailanforderungen wurden uns klar. Diese nahmen wir unter die Lupe und beurteilen sie nach Nutzen und Aufwand. Dadurch veränderte sich laufend, aber iterativ der Umfang unserer ursprünglich geplanten Features und Funktionen. Zum Beispiel haben wir einen Fehler bei der Wahl eines UI-Frameworks gehabt, bei dem wir uns anfangs für Material-UI entschieden hatten. Später bemerkten wir, dass es gewünschter ist, ein Open-Source Framework wie Chakra-UI in

Kombination mit Tailwind CSS zu verwenden. Dadurch kann die UI detaillierter und nach den Wünschen und Ansprüchen der spezifischen Anforderungen angepasst werden.

Auch haben wir hier mehr Anfangsarbeit für UI/UX eingerechnet als nötig war, da wir uns entschieden haben das Projekt eher iterativ zu entwickeln, uns auf die Funktionalität zu fokussieren, und das UI/UX am Schluss zu machen, damit wir schneller Probleme und Anforderungen von Entwickler:innen validieren konnten. Zudem konnten wir OnLaunch somit früher in unsere eigene App kula integrieren und es an echten Produkten testen. Dafür adaptierten wir unsere Planung und teilten Stunden auf andere Mitarbeiter:innen auf. Mehr zur UI-Umstellung kann in unserem netidee-Blog gelesen werden.

Durch den Austausch mit potentiellen Kund:innen/Entwickler:innen passten wir bis zum Abschluss des Pakets den Server bzw. dessen Funktionalität und weitere technische Details hin und wieder neu an. Dank Flo passierten diese Adaptionen und Anpassungen in kurzer Zeit.

### 3.6 Arbeitspaket 6 - <Dokumentation und formales Projektende>

Mit der Abarbeitung und Erfüllung der Vorgaben nähern wir uns dem formalen Ende der Förderung. Dieses Arbeitspaket, welches als letztes begonnen wurde, umfasst auch dieses Dokument als einen Teil der abschließenden Arbeiten.

# 4 Umsetzung Förderauflagen

Es sind keine speziellen Förderauflagen vereinbart.

# 5 Liste Projektendergebnisse

| | | | |
|---|---|---|---|
| *1* | *Projektzwischenbericht* | *CC BY-SA 4.0* | *https://www.netidee.at/onlaunch* |
| *2* | *Projektendbericht* | *CC BY-SA 4.0* | *https://www.netidee.at/onlaunch* |
| *3* | *Entwickler_innen-DOKUMENTATION des* | *Apache 2.0 bzw. MIT* | *https://www.netidee.at/onlaunch* |

| | | | |
|---|---|---|---|
| | *Projektergebnisses für andere Entwickler_innen ("Dritte"), die das Projektergebnis nach Projektende nutzen/weiterentwickeln wollen*<br><br>*Für Entwickler_innen (Systemkonzept, ggf. Grobspezifikationen):*<br><br>*a. WAS IST ES*<br><br>*b. FÜR WEN IST ES /WEM HILFT ES WODURCH*<br><br>*c. WIE FUNKTIONIERT ES (für Entwickler_innen: Übersicht und detailliertes Systemkonzept, SW-Struktur)* | | *https://github.com/kula-app/OnLaunch/blob/main/docs/index.md* |
| *4* | *Anwender_innen-DOKUMENTATION des Projektergebnisses für Anwender_innen, die das Projektergebnis nach Projektende nutzen wollen*<br><br>*Für Anwender_innen ("Bedienungsanleitung") :*<br><br>*a. WAS IST ES*<br><br>*b. FÜR WEN IST ES /WEM HILFT ES WODURCH*<br><br>*c. WIE FUNKTIONIERT ES* | *CC BY-SA 4.0* | *https://www.netidee.at/onlaunch* |
| *5* | *Veröffentlichungsfähiger Einseiter / Zusammenfassung*<br><br>** Kurzfassung WAS \| FÜR WEN \| WIE*<br><br>** Liste Projektergebnisse - also diese Liste, ggf. kompromiert*<br><br>** mit Angabe Open Source Lizenz/Webadresse* | *CC BY-SA 4.0* | *https://www.netidee.at/onlaunch* |

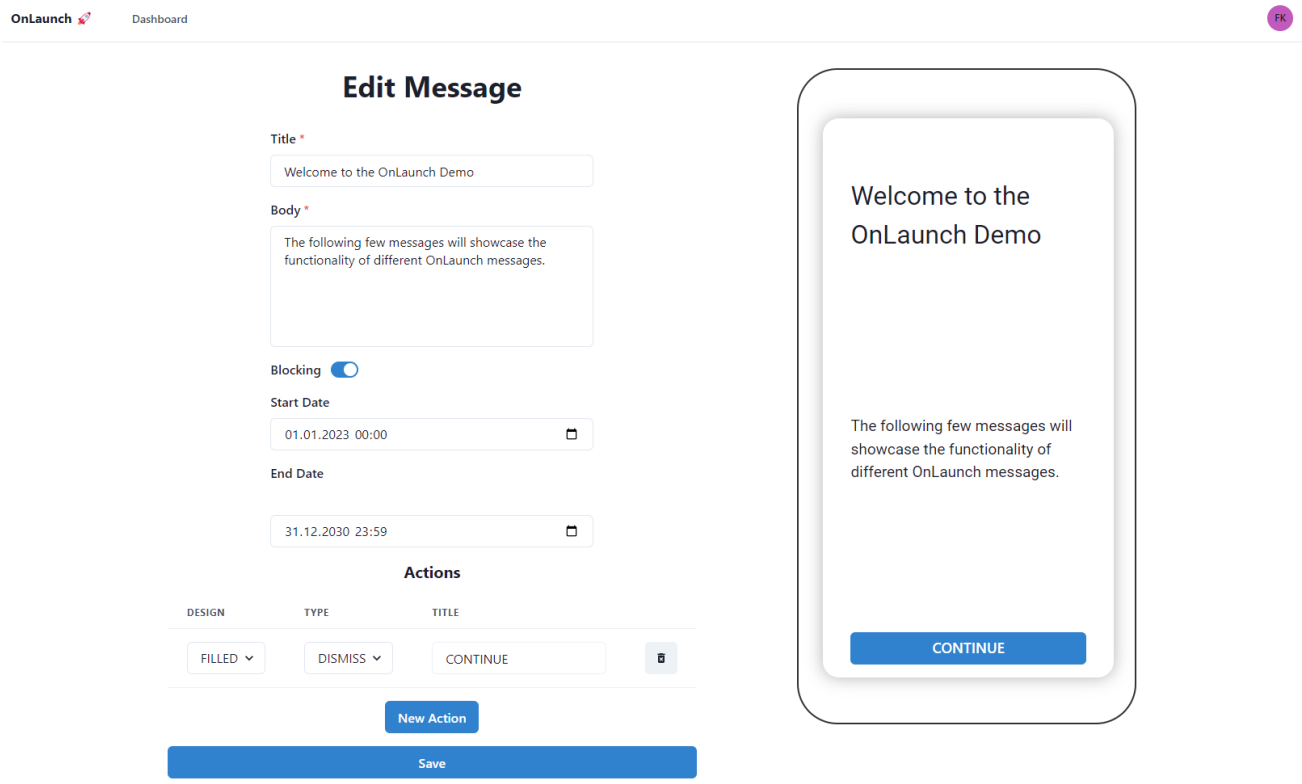
| | | | |
|---|---|---|---|
| | ** wo finden Dritte die Projektergebnisse (inkl. Dokumentation Anwender_innen bzw. Entwickler_innen)*<br>** mögliche Weiterentwicklungen/ weitere Einsatz-/ Nutzungsmöglichkeiten* | | |
| *6* | *Dokumentation Externkommunikation zur Erreichung Sichtbarkeit /Nachhaltigkeit (**als Teil des Endberichtes**)*<br>** Welche Maßnahmen wurden in welchem Umfang gesetzt*<br>** Jeweils Bewertung Aufwand / Nutzen*<br>** Lessons Learned / Empfehlungen für andere Projekte* | *CC BY-SA 4.0* | *https://www.netidee.at/onlaunch* |
| *7* | *Self-Hosted Server-/Cloud-Software* | Apache 2.0 | *https://github.com/kula-app/OnLaunch*<br>*https://www.netidee.at/onlaunch* |
| *8* | *iOS Software Modul* | *MIT* | *https://github.com/kula-app/OnLaunch-iOS-Client*<br>*https://www.netidee.at/onlaunch* |
| *9* | *Android Software Modul* | *MIT* | *https://github.com/kula-app/OnLaunch-Android-Client*<br>*https://www.netidee.at/onlaunch* |
| *10* | *Dokumentation* | *Apache 2.0 bzw. MIT* | *http://www.kula.app/onlaunch*<br>*https://www.netidee.at/onlaunch*<br>*https://github.com/kula-app/OnLaunch/blob/main/docs/index.md* |

# 6 Verwertung der Projektergebnisse in der Praxis

OnLaunch haben wir bereits in unser laufendes System bei kula eingebunden. Damit haben wir uns selbst bewiesen, wie ausgereift und stabil die Software an sich ist und wie hoch der Nutzen des Projekts für uns ist und wie viel Potential es für andere hat. Jede App, die mittels kula erstellt wird, bekommt OnLaunch direkt integriert.

Durch eines unserer neuesten Features, nämlich die Admin API, wird eine Server-zu-Server-Kommunikation automatisch und simpel ermöglicht. Diese verwenden wir bei kula, um auf OnLaunch-Daten zuzugreifen und zu verwalten.

Durch die graphische Oberfläche können auch nicht-technische Personen Messages erstellen und verwalten und erhalten zudem eine Vorschau dieser Message:




# 7 Öffentlichkeitsarbeit/ Vernetzung

- Wir planen einen Product Hunt Launch sobald das Projekt vollständig von der Open Source Community einsetzbar ist.
- Nach Bedarf führen wir weiter vereinzelt Kundeninterviews durch, um unser Produkt den etwaigen Wünschen und Anforderungen potenzieller Kund:innen/Entwickler:innen anzupassen.
- Wir suchen nach weiteren Contributors, als auch Tester:innen und Pilot-Kund:innen. Über Lukas, unseren ersten Contributor, haben wir in einem Blog-Post Genaueres beschrieben.
- Bei dem Demo Day des ESI-Programms des Innovation Incubation Centers der TU Wien durften wir im Juni OnLaunch pitchen, was uns einiges an Aufmerksamkeit und neuen Kontakten gesichert hatte.

# 8 Eigene Projektwebsite

| | |
|---|---|
| OnLaunch Server Repository: | https://github.com/kula-app/OnLaunch |
| OnLaunch Android Client Repository: | https://github.com/kula-app/OnLaunch-Android-Client |
| OnLaunch iOS Client Repository: | https://github.com/kula-app/OnLaunch-iOS-Client |
| kula / OnLaunch Webseite *: | https://www.kula.app/onlaunch |

* auf der aktuellen kula-Webseite werden wir einen Unterbereich für OnLaunch erstellen

### Aus der Community:

| | |
|---|---|
| OnLaunch Flutter Client Repository: | https://github.com/kula-app/OnLaunch-Flutter-Client |

# 9 Geplante Aktivitäten nach netidee-Projektende

Neben den bereits genannten Tätigkeiten in Kapitel 7 und vielen weiteren haben wir diese Aktivitäten geplant:

- **UI-Overhaul:** Zwar sind wir durch die Verwendung von Chakra UI und Tailwind CSS bereits stark aufgestellt und der Code selbst dadurch leicht anpassbar. Jedoch wollen wir das UI-Design verbessern und überholen, um unseren Nutzer:innen die bestmöglichste Erfahrung zu bieten.
- **Community-Growth:** Wir durften bereits einige Leute in unserer Community willkommen heißen, machen hier aber keinen Halt. Hier suchen wir vor allem nach Tester:innen, Contributer:innen und Pilot-Kund:innen.
- **Community-Feedback:** Von unserer aktuellen Community haben wir bereits einige Feedback-Punkte erhalten, welche OnLaunch noch besser geeignet machen für unsere Nutzer:innen. Deswegen wollen wir auf das Feedback eingehen und es umsetzen.

- **Weitere Arbeitspakete:** Da wir selbst Nutzer sind von OnLaunch, haben wir noch viele Ideen und Arbeitspakete, die wir bei OnLaunch umsetzen wollen. Bereits konkrete Ideen können laufend in unserem GitHub-Hauptrepositorium aufgefunden werden. An diesen wird ebenfalls laufend gearbeitet.

# 10 Anregungen für Weiterentwicklungen durch Dritte

OnLaunch erleichtert für jede:n Nutzer:in das Erstellen und Verwalten von Messages, welche innerhalb mobiler Apps angezeigt werden, soll. Dafür ist nur anfangs ein Programmieraufwand von Nöten. Durch die intuitive GUI können auch Menschen ohne Software-Hintergrund einfach damit umgehen, neue Messages erstellen und bearbeiten.

Wer nicht selbst den OnLaunch Server hosten möchte, kann sich bei uns gerne über unsere SaaS-Plattform einmieten. Anschließend muss man sich nur um das Integrieren in die mobilen Clients kümmern.

Von uns geplant war ursprünglich nur das Unterstützen von mobilen Clients für Android und iOS. Mittlerweile gibt es von der Community auch einen Flutter-Client. Wir freuen uns, wenn es in Zukunft noch mehr Clients gibt, um weitere Plattformen unterstützen zu können.

Durch den Austausch mit unserer aktuellen Community zeichnete es sich ab, dass es immer etwas geben wird, das man erweitern oder anpassen kann, um den größtmöglichen Nutzen für Entwickler:innen zu bieten. Unser GitHub-Haupt-Repository bietet einen Einblick, woran wir gerade arbeiten, und ebenfalls kann es genutzt werden, um mit uns in Kontakt zu treten, neue Issues (Arbeitspakete) vorzuschlagen und im besten Fall uns bei der Umsetzung zu helfen.