

# HotkeyApp

T. Dunker, S. Herrmann, T. Meier, A. Stahl, S. Voigt

# Allgemeines

Das Softwareprojekt HotkeyApp soll Nutzern helfen, Befehle von ihrem Smartphone für eine von Ihnen ausgewählte Applikation auszuführen. Die Softwarekomponenten wurden so aufgebaut, dass eine Beteiligung der Community vorausgesetzt wird, jedoch nicht zwingend erforderlich ist. Jedes Individuum der Community kann sogenannte HotkeyMaps selbst erstellen und anderen Mitgliedern zur Verfügung stellen, muss dies aber nicht. Es ist auch möglich private HotkeyMaps zu erstellen, welche zum Beispiel für Inhouse-Software verwendet werden, von denen die Community ohnehin keinen Nutzen hätte. Die Aufgaben der jeweiligen Komponenten werden in den nächsten Unterpunkten dieses Kapitels behandelt.

## Das Team und seine Aufgaben

### SpringBoot-Applikation

Bildet die Community-Komponente des Systems. Stellt ein REST-Service zur Verfügung. Eine ausführliche Auflistung der implementierten REST-Services kann unter dem Kapitel: [REST-Service](#) eingesehen werden.

- Alexander Stahl - Implementierung aller Klassen, Interfaces, Resources und User-Testtreiber
- Sebastian Voigt - Implementierung aller Domain-Klassen, Dokumentation und HotkeyMap-Testtreiber

### Android-Applikation

Stellt die Benutzerschnittstelle zur Ausführung von Hotkeys dar. Eine Ausführung der App ist ohne eine ausgewählte HotkeyMap in der Ausführungs-Applikation nicht möglich, da diese eine HotkeyMap an die App liefert. Das Android-Smartphone und der PC mit der Ausführungs-Applikation müssen sich im selben Netzwerk befinden.

- Timm Dunker
- Sebastian Herrmann

### Ausführungs-Applikation

Diese Komponente wird auf dem Computer des Benutzers ausgeführt. Sie nimmt die von der [Android-Applikation](#) gesendeten Befehle entgegen und führt diese aus. Sie ist auch für die Auslieferung der von Benutzer gewählten Map an die [Android-Applikation](#) verantwortlich. Eine bestehende Internetverbindung ist nur zum beziehen aller von der Community bereitgestellten Maps erforderlich.

- Alexander Stahl - Implementierung der Model-/Controllerklassen sowie Dokumentation der Klassen
- Thorsten Meier - Implementierung der Controller- und View-Klassen

## Sonstige Aufgaben

- Sebastian Voigt - Erstellung der Klassen- und Paketdiagramme
- Alexander Stahl - Anfertigen dieser Dokumentation

## Fremdbibliotheken

Zur Umsetzung des Projekts wurden verschiedene Fremdbibliotheken verwendet. Ihr jeweiliger Einsatzbereich wird im folgenden näher erläutert:

- [JFoenix](#) - Eine JavaFX-Bibliothek, die Googles MaterialDesign nachempfunden ist und dieses für JavaFX zur Verfügung stellt. Die Wahl der Bibliothek erfolgte aufgrund ihrer Einfachheit sowie dem schlichten und modernen Erscheinungsbild der Oberflächenkomponenten.
- [JSONSimple](#) - Generelle Java-Bibliothek zum parsen und erstellen von Dateien im JSON-Format. An den [REST-Service](#) gestellte Anfragen zum erhalten von HotkeyMap-Informationen/-Sammlungen werden in Form von JSON-Responses zurückgeliefert. Durch die langjährige Bewährung dieser Bibliothek und Ihrer Einfachheit machte diese zu einem geeigneten Kandidaten.
- [KryoNet](#) - Ist eine Java-Bibliothek für den einfachen Aufbau von TCP-/UDP-Verbindungen zwischen zwei Systemen. Diese wird genutzt um die Kommunikation zwischen der [Ausführungs-Applikation](#) und der [Android-Applikation](#) aufzubauen.

## Installationsanleitung / Kompilierung

Das System wurde unter dem Linux-Derivat [Ubuntu 16.04](#) entwickelt und getestet. Prinzipiell sollte das System unter folgenden Betriebssystemen ausführbar sein:

- Linux
- Windows
- MacOS

Die Betriebssysteme MacOS und Windows wurden jedoch nicht getestet, somit kann eine fehlerfreie Ausführung der Programme auf diesen Plattformen nicht gewährleistet werden.

## Installationsanleitung

Die Anwendungen werden als [JAR-Dateien](#) zur Verfügung gestellt. Eine Installation dieser ist nicht notwendig. Die grundlegende Voraussetzung zum ausführen der Anwendung auf einem PC (Ausführungs-Applikation) bzw. Server (SpringBoot-Applikation) erfordert das [JavaRuntimeEnviroment](#) mindestens in der Version 7. Die zum senden von Befehlen benötigte Android-Applikation wird ab der Android-Version 4 unterstützt. Das Ausführen der entsprechenden Komponente erfolgt durch einfaches Doppelklicken auf die JAR-Datei (SpringBoot-/Ausführungs-Applikation). Soll der Einsatz der REST-Komponente in einem Betriebssystem ohne grafische Benutzerschnittstelle erfolgen so kann dieses mit folgendem Befehl bewerkstelligt werden:

```
java -jar HotKeyApp-REST.jar
```

## Kompilierung von Quelldateien

Für die Kompilierung der einzelnen Komponenten werden folgende IDEs empfohlen:

- Springboot-Applikation: [SpringBoot](#)

### Build mit Maven:

Das komplette System als JAR kompilieren: `mvn package`.

Das System testen : `mvn test`

Dokumentation erstellen (JavaDoc) : `mvn site`

- Android-Applikation: [Android-Studio 3](#) mit installiertem Android SDK in der Version 26
- Ausführungs-Applikation: [IntelliJ](#)

# Benutzerdokumentation

Im folgenden finden Sie eine Übersicht der grafischen Benutzerschnittstellen und ihrer Funktionen. Da das gesamte System aus zwei voneinander unabhängigen Benutzerschnittstellen besteht (*Android-Client* und *PC-Server*), werden diese auch separat behandelt. Die dritte Komponente, der REST-Service, wird in diesem Kapitel nicht behandelt, da er keine grafische Oberfläche besitzt und Benutzer prinzipiell über die Oberfläche des PC mit diesem interagieren.

## PC-Benutzerschnittstelle

### Einen neuen Benutzer Anlegen

Einige Interaktionen mit dem REST-Service benötigen eine Benutzerkonto, beispielsweise wenn sie für eine Map [Voten](#) oder selbst eine Map für die Community bereitstellen möchten. Wenn Sie nur lokal arbeiten möchten, benötigen Sie nicht zwangsläufig ein Benutzerkonto. Wir empfehlen trotzdem die Erstellung eines selbigen. Wenn Sie ein Benutzerkonto erstellen möchten, starten sie das Programm und gehen Sie wie folgt vor:

- 1.) Klicken Sie auf die markierte Schaltfläche zum öffnen des Optionsmenüs.
- 2.) Klicken Sie auf die markierte Schaltfläche mit dem Benutzerlogo. Es öffnet sich nun das Benutzerkontomenü.
- 3.) Stellen Sie sicher, dass die Checkbox *Neues Konto erstellen* ausgewählt ist. Geben sie nun ihren gewünschten Benutzernamen und ihr Wunschpasswort ein.
- 4.) Klicken Sie auf die Schaltfläche *Konto erstellen* unten rechts. Das Programm gibt ihnen eine Rückmeldung ob der Vorgang erfolgreich war, eine Übersicht der Fehlermeldungen, finden Sie im Kapitel [\[Fehlermeldungen und mögliche Gründe\]](#)

**CAUTION** Bitte beachten Sie, das Ihr Passwort aus mindestens 8 Zeichen bestehen muss.

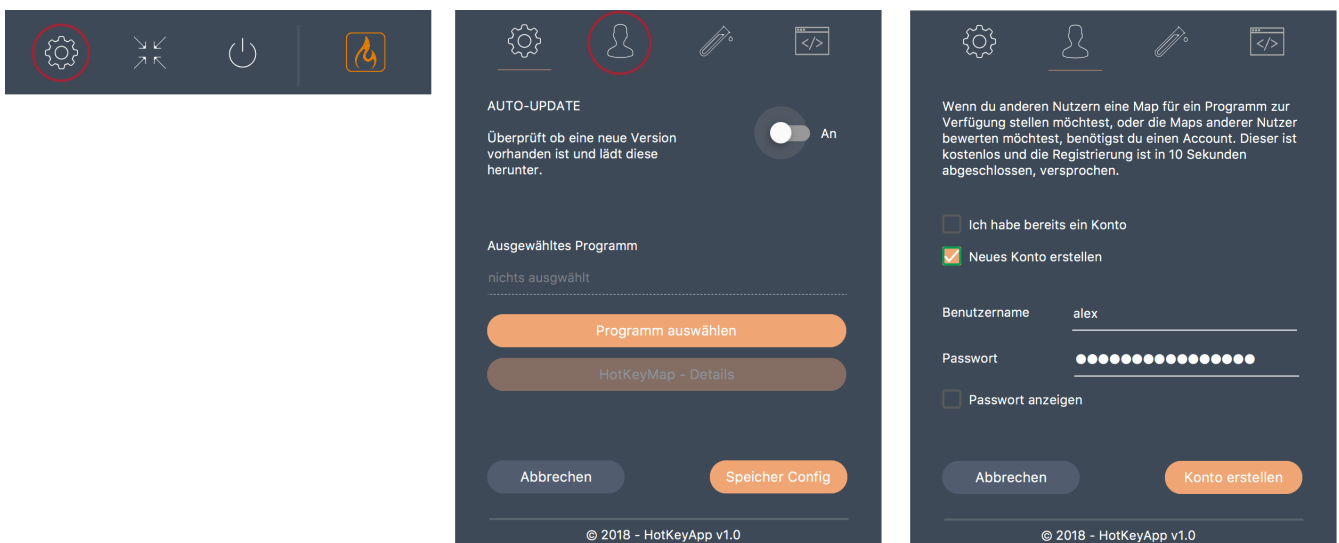


Abbildung 1: Schritte zum erstellen eines neuen Benutzerkontos.

## TIP

Wenn Sie ihre Benutzerdaten nicht nach jedem beenden des Programms erneut eingeben möchten, wechseln Sie nach erfolgreichem Abschluss des Vorgangs zurück ins Optionsmenü (Bild in der Mitte) und Klicken Sie auf *Speicher Config*. Ihre Benutzerdaten werden beim nächsten Start der Applikation automatisch geladen.

## Eine Map aus der Community öffnen

Die Community und wir erstellen HotkeyMaps für verschiedene Programme und stellen diese anderen Benutzern zur Verfügung. Die Applikation lädt beim starten, soweit Sie diese Option aktiviert haben, die neuste Version aller HotkeyMaps die auf dem REST-Service zur Verfügung stehen. Sie können dann eine dieser Maps auswählen.

- 1.) Klicken Sie auf die markierte Schaltfläche zum öffnen des Optionsmenüs.
- 2.) Klicken Sie auf die Schaltfläche *Programm auswählen*. Es öffnet sich nun das Auswahlfenster.
- 3.) Auf der linken Seite sehen Sie den Namen der Applikation für die eine HotkeyMap existiert. Auf der rechten Seiten bekommen Sie ein Feedback über den "Zustand" der Map(dieser Punkt wird näher im Kapitel [Voten](#) behandelt). Wählen Sie eine Map aus und drücken Sie den Button *Auswählen*.
- 4.) Um den Server mit der ausgewählten Map zu starten, klicken Sie auf das Flammenicon im Hauptmenü.

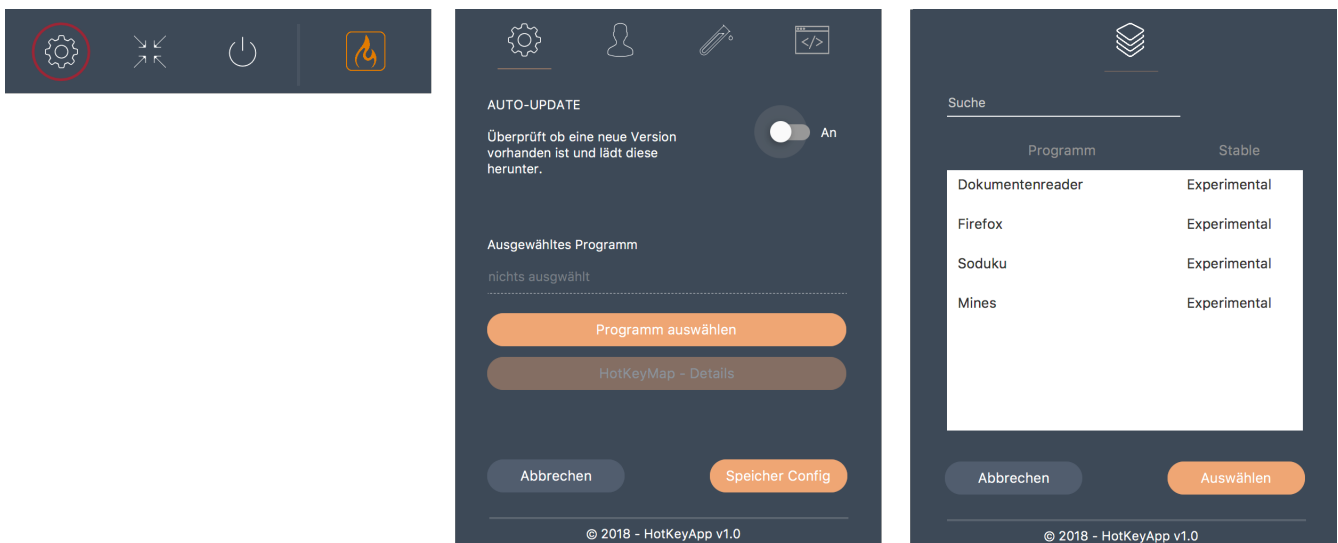


Abbildung 2: Eine von der Community bereitgestellte HotkeyMap auswählen.

## Voten

Haben Sie eine Map ausgiebig getestet, können Sie mit einem Klick auf die Schaltfläche *Vote Works* Ihre Stimme für die Funktionalität der Map abgeben, sodass andere Benutzer Maps besser einschätzen können. Wünschen Sie sich die Verbesserung oder Neuerstellung einer Map können sie dies durch einen Klick auf *Vote Wish* tun.

- 1.) Nachdem Sie den Schritt: [Eine Map aus der Community öffnen](#) abgeschlossen haben, wird die hier (in der Mitte) ausgegraute Schaltfläche *HotkeyMap Details* auswählbar. Drücken Sie sie.

- 2.) Nach dem Erscheinen des Detailfensters können Sie sich einen Überblick über die Eigenschaften der ausgewählten Map verschaffen. Sie können nun Ihre Stimme zu dieser Map abgeben durch das Klicken auf die Schaltfläche *Vote Works* oder *Vote Wish*.

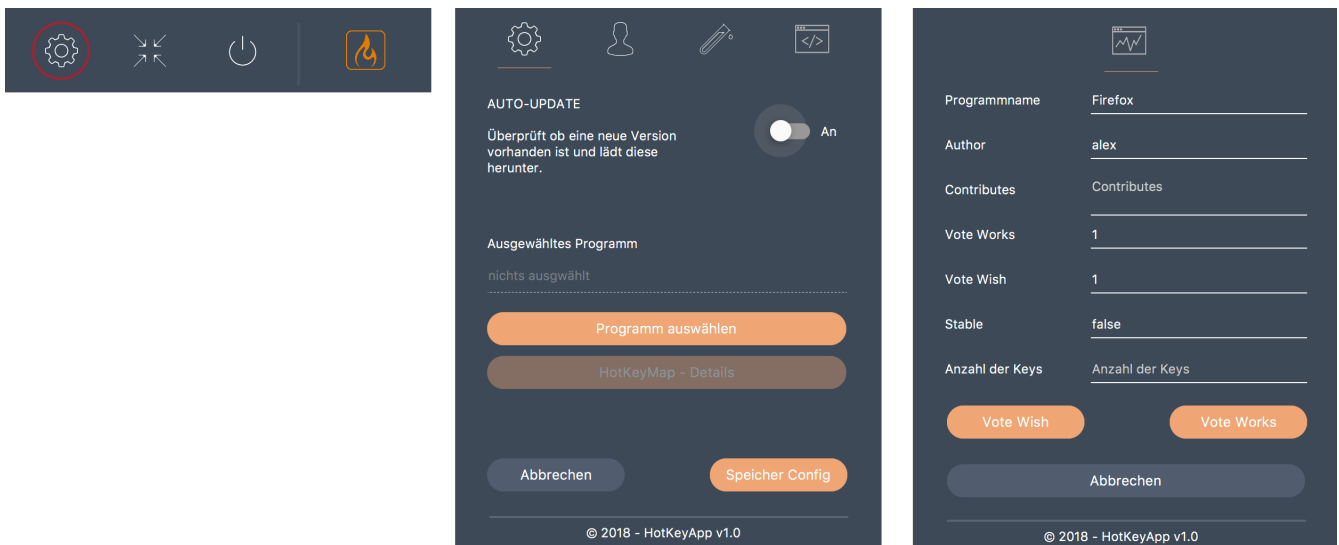


Abbildung 3: Eine von der Community bereitgestellte HotKeyMap bewerten.

## Eine Map erstellen / bearbeiten

Sie können eigene HotKeyMaps erstellen oder bestehende um weitere Optionen erweitern. Um eine eigene Map zu erstellen oder eine bestehende zu bearbeiten gehen Sie wie folgt vor.

- 1.) Klicken Sie auf die markierte Schaltfläche zum Öffnen des Optionsmenüs.
- 2.) Klicken Sie auf die markierte Schaltfläche mit dem Reagenzglas. Es öffnet sich nun das Map-Options-Menü.
- 3.) Klicken Sie auf die Schaltfläche 'Eigene Map erstellen'. Es öffnet sich das bearbeiten/erstellen Menü.

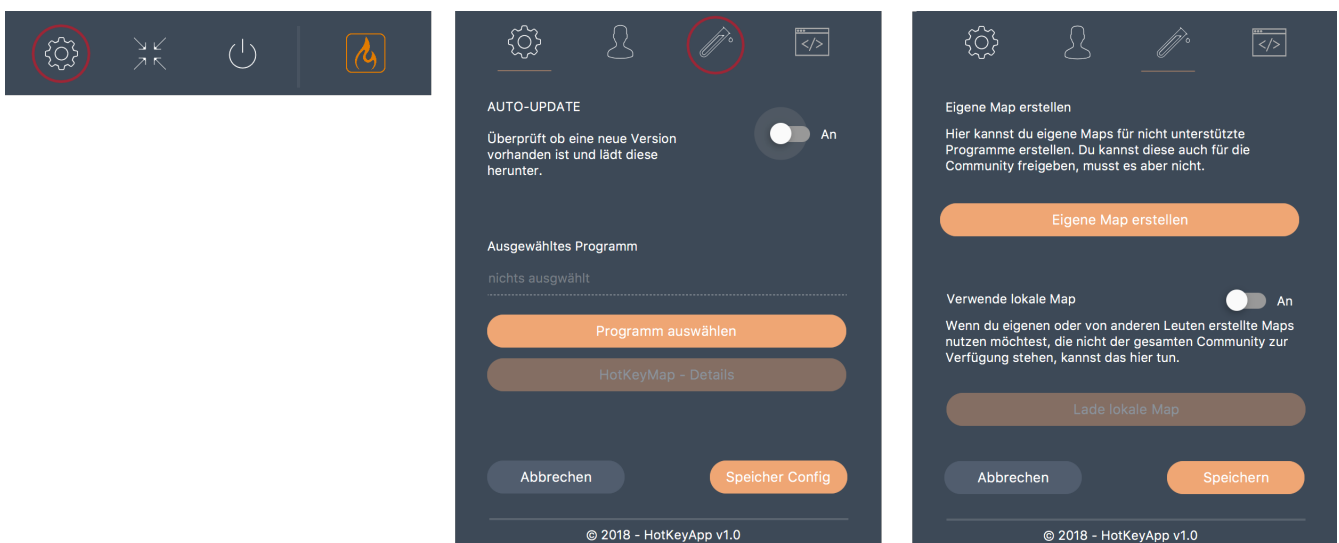


Abbildung 4: Eine eigene HotKeyMap erstellen / eine bestehende bearbeiten

Hier haben Sie mehrere Optionen die Ihnen zur Verfügung stehen. Beide Operationen unterscheiden sich nur minimal voneinander. Möchten Sie eine neue Map erstellen, stellen Sie sicher, dass Sie die Checkbox *Neue Map erstellen* ausgewählt haben.

- 1.) Vergeben Sie einen Namen für Ihre Map, idealerweise ist das die Applikation für die Sie diese Map erstellen z.B. Libre Office 5 oder so ähnlich.
- 2.) Geben Sie einen Aktionsname ein, beispielsweise *Kopieren*
- 3.) Geben Sie den entsprechenden Shortcut ein: *strg c* ← **Wichtig:** mehrere Tasten müssen durch ein Leerzeichen getrennt werden. Benötigen sie ein Leerzeichen (als Shortcut) tragen Sie bitte *space* ein.
- 4.) Wiederholen Sie Schritt 3 so oft Sie wollen. Es ist nicht möglich eine Aktionsnamen oder eine Kombination doppelt zu vergeben. Das Programm wird Ihnen in diesem Fall eine entsprechende Fehlermeldung liefern.
- 5.) **Wichtig:** Ihre Map ist bis jetzt nur temporär, möchten Sie sie zu einem späteren Zeitpunkt wiederverwenden oder erweitern vergessen Sie nicht sie mit einem Klick auf *Lokal Speichern* zu sichern.

Abbildung 5: Eine Map erweitern / erstellen / veröffentlichen.

## Eine lokale Map laden

Sie können eigene HotkeyMaps erstellen oder bestehende um weitere Optionen erweitern. Um eine eigene Map zu erstellen oder eine bestehende zu bearbeiten gehen Sie wie folgt vor.

- 1.) Klicken Sie auf die markierte Schaltfläche zum öffnen des Optionsmenüs.
- 2.) Klicken Sie auf die markierte Schaltfläche mit dem Reagenzglas. Es öffnet sich nun das Map-Options-Menü.
- 3.) Schalten Sie den Schieberegler *Verwende lokale Map* an.
- 4.) Sie können nun über gerade freigeschalteten Button *Lade lokale Map* eine auf Ihrem Computer gespeicherte Map laden. Diese wird dann automatisch als aktive Map gesetzt, sodass sie neue Kommandos über [Eine Map erstellen / bearbeiten](#) bearbeiten können. Die Statistiken für Ihre Map können Sie wie im Kapitel [Voten](#) einsehen. Bei lokalen Maps ist die Votingfunktion deaktiviert.



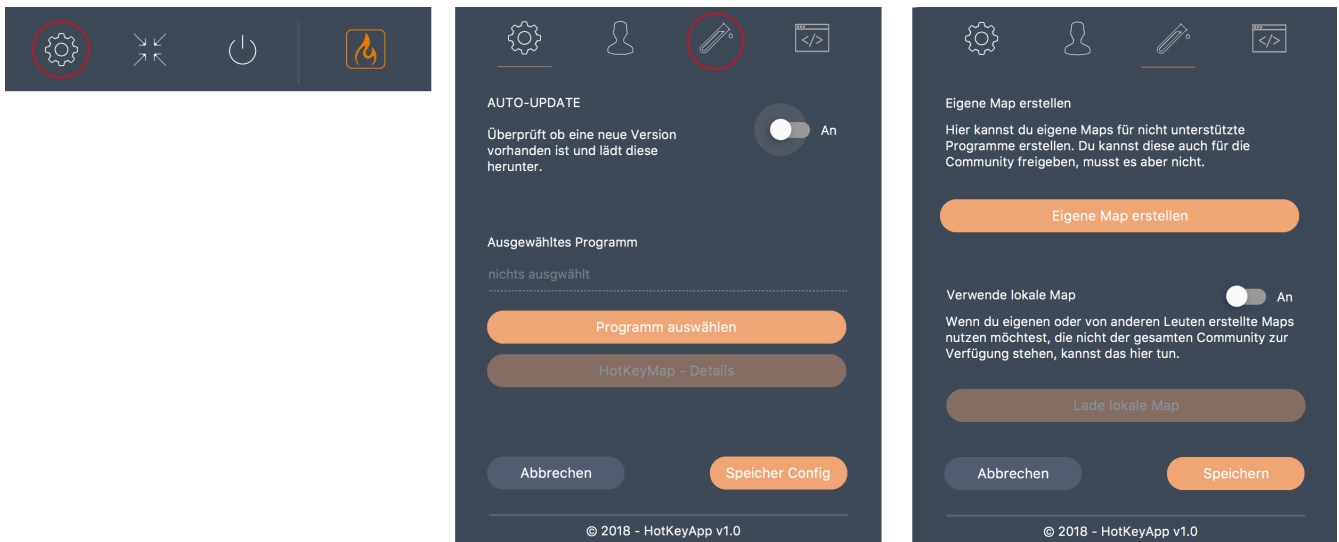


Abbildung 6: Eine lokal gespeicherte HotkeyMap laden

## Eine eigene Map veröffentlichen

Haben Sie eine HotkeyMap erstellt welche Sie mit der Community teilen möchten, haben Sie die Möglichkeit dies mit folgenden Schritten zu realisieren:

- 1.) Laden Sie eine lokal gespeicherte Map wie im Kapitel "[Eine lokale Map laden](#)"
- 2.) Klicken Sie nun auf die Schaltfläche *Eigene Map erstellen*
- 3.) Aktivieren Sie die Checkbox *Diese Map veröffentlichen*
- 4.) Mit dem drücken des Buttons *Veröffentlichen* wird Ihre Map an den REST-Service übermittelt.

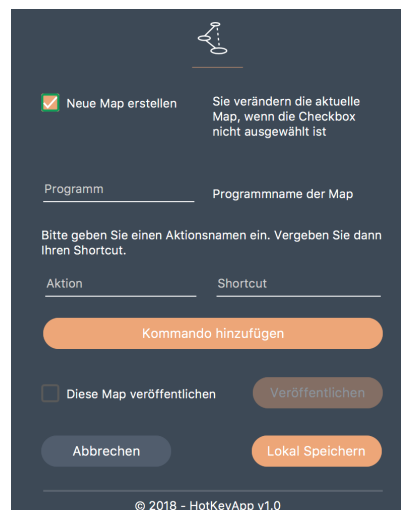


Abbildung 7: Eine eigene HotkeyMap veröffentlichen und der Community zur Verfügung stellen.

## Server starten um Befehle zu Empfangen

Haben Sie die Schritte "[Eine Map aus der Community öffnen](#)" oder "[Eine lokale Map laden](#)" erfolgreich abgeschlossen, können Sie den Server mit einem Klick auf das Flammenlogo starten. Ist der Server gestartet leuchtet das Flammenlogo orange, wie im Bild zu sehen.



Abbildung 8: Hauptmenü der HotkeyApp - Applikation

## Android-Client

Der Android-Client dient zum Auslösen der Kommandos. Ist der Server wie im Kapitel "[Server starten um Befehle zu Empfangen](#)" gestartet worden, können Sie nun die [Android-Applikation](#) auf Ihrem Smartphone starten. Die App bitte Sie eine Verbindung mit dem Server herzustellen (Abbil. links). Diese Operation lösen Sie durch Tippen des Buttons *CONNECT TO SERVER* aus. Die sucht nun in Ihrem Netzwerk nach einem gestarteten Server (Abbil. Mitte), ist dieser gefunden übergibt er der App die im Kapitel "[Eine Map aus der Community öffnen](#)" oder "[Eine lokale Map laden](#)" von Ihnen ausgewählte Map. Die App zeigt Ihnen nun Buttons mit den verfügbaren Befehlen. Zum Auslösen dieser drücken sie den entsprechenden Button.

**Wichtig:** Sie müssen das Programm in dem der Befehl ausgeführt werden soll, im Vordergrund haben.

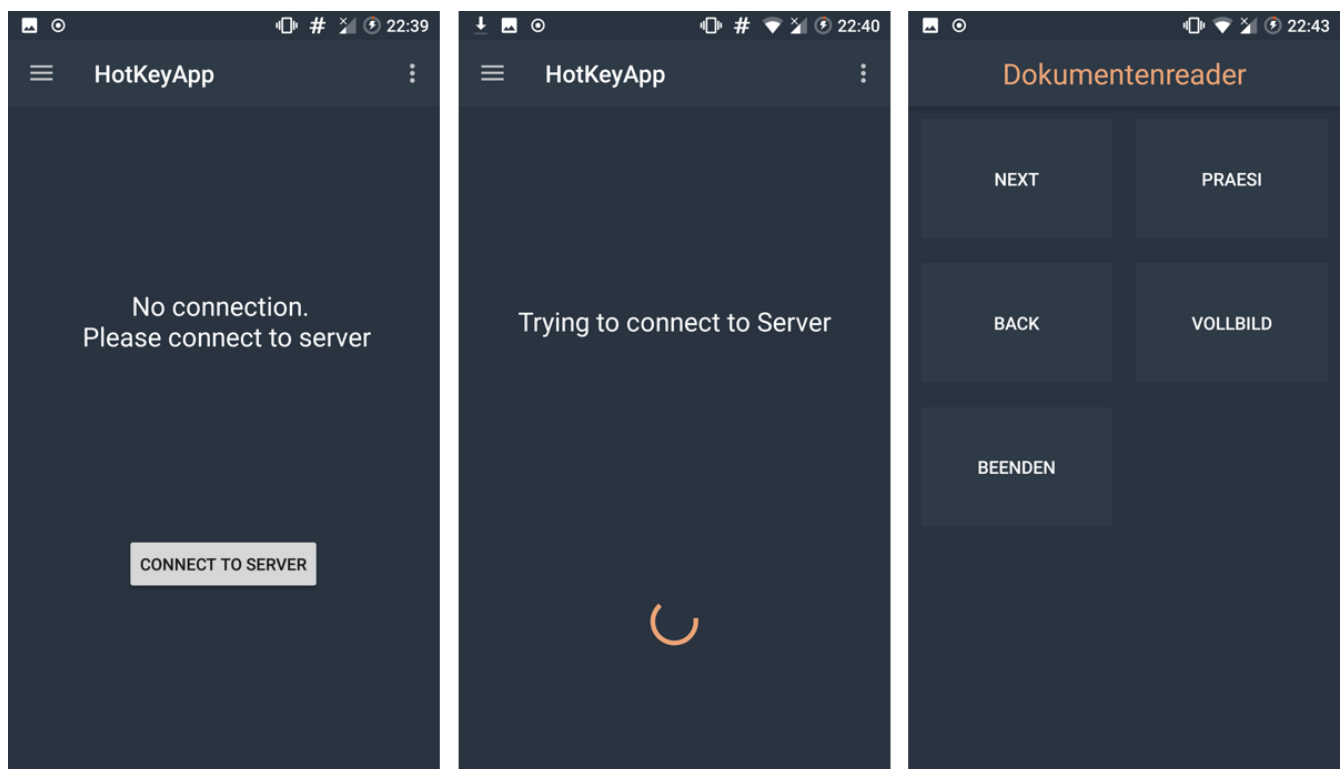


Abbildung 9: Oberfläche der Android-Client App.

# REST-Service

Der REST-Service bietet Möglichkeiten mit den Entitäten User und HotkeyMap des Systems zu interagieren. Die Einzelheiten der entsprechenden Entität und ihre erreichbaren REST-Adressen werden im folgenden behandelt. Der Server wartet nun auf Clients die sich zu ihm verbinden um Ihnen die von Ihnen ausgewählte Map zu übergeben und wartet auf Befehle die durch einen Clienten ausgelöst wurden.

## Paketdiagramm

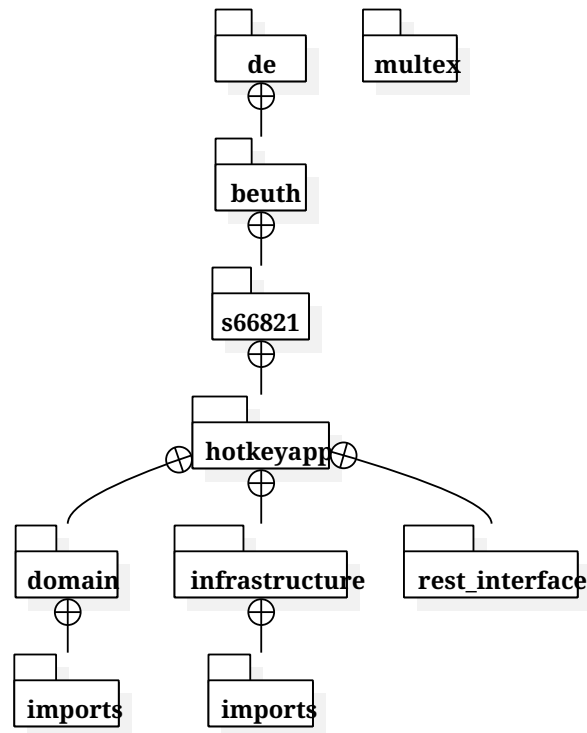


Abbildung 10: Aktuelle Paketdiagramm des REST-Services

# Fachklassendiagramm

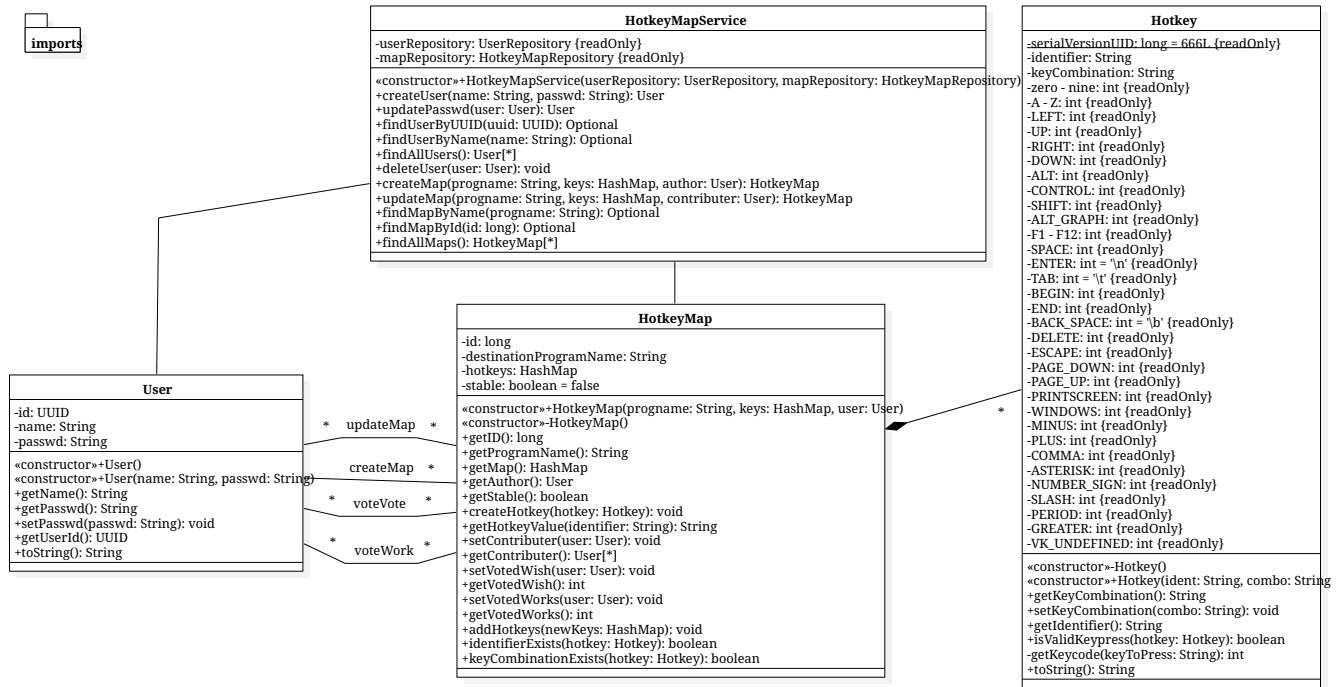


Abbildung 11: Aktuelles Fachklassendiagramm des REST-Services

## REST-Service Adressen

### Get-Methoden (Maps)

```

...
// Alle Maps
@GetMapping(path="/maps")
public ResponseEntity<HotkeyMapResource[]> findHotkeyMaps(@RequestParam(name="",
defaultValue="") final String progName){...}
// By ID
@GetMapping("/map/{id}")
public ResponseEntity<HotkeyMap> findHotkeyMap(@PathVariable final long id) {...}
// By Name
@GetMapping(path="/map/name/{progname}")
public ResponseEntity<HotkeyMap> findHotkeyMap(@PathVariable final String
progname){...}

```

## Post-Methoden (Map)

```
...
// Neue Map
@PostMapping("/map/new")
public ResponseEntity<HotkeyMapResource> createMap(@RequestBody final
HotkeyMapResource hotkeyMapResource){...}
// Vote Works
@PostMapping("/map/{id}/vote/works/")
public ResponseEntity<HotkeyMapResource> votedWorks(@PathVariable final long id,
@RequestBody final UserResource userResource){...}
// Vote Wish
@PostMapping("/map/{id}/vote/wish/")
public ResponseEntity<HotkeyMapResource> votedWish(@PathVariable final long id,
@RequestBody final UserResource userResource) {...}
@PostMapping("/map/addKeys")
public ResponseEntity<HotkeyMapResource> updateMap(@RequestBody final
HotkeyMapResource hotkeyMapResource) {...}
```

## Get-Methoden (User)

```
...
// Alle User
@GetMapping(path="/users")
public ResponseEntity<UserResource[]> findUsers( @RequestParam(name="",
defaultValue="") final String username) {...}
// By ID
@GetMapping("/user/id/{userId}")
public ResponseEntity<User> findUser(@PathVariable final UUID userId) {...}
// By Name
@GetMapping("/user/name/{username}")
public ResponseEntity<User> findUser(@PathVariable final String username) {...}
```

## Post-/Put-/Delete-Methoden (User)

```
...
// Neuer User
@PostMapping("/register")
public ResponseEntity<UserResource> createUser(@RequestBody final UserResource
userResource){...}
// Update password
@PutMapping("/user/update/{username}")
public ResponseEntity<User> updatePasswd(@RequestBody final UserResource
userResource){...}
// Delete User
@DeleteMapping("/user/delete/{username}")
public ResponseEntity<String> deleteUser(@PathVariable final String username){...}
```

## Schlussbewertung

Das von uns im ersten Teil der Veranstaltung (SoftwareEngineering I) geplante Projekt ließ sich erfolgreich umsetzen. Jedoch konnten wir auch feststellen, dass das Modell in seiner geplanten Ursprungsform nicht fehlerfrei war. Viele der im Vorfeld geplanten Assoziationen mussten im Laufe der Umsetzung angepasst werden. Es zeigt sich, dass iterative Entwicklungsprozesse, zumindestens aus unsere Sicht, für die Umsetzung von Software effektiver sind. Es ist schwierig ein Projekt von vornherein fehlerfrei zu planen. Es scheitert häufig an einer nicht fehlerfreien Planung des Projektes, beziehungsweise an Problemen die erst bei der Implementierung auftreten, denen im Vorfeld keine Beachtung geschenkt oder die schlicht und einfach nicht berücksichtigt wurden. Zusammengefasst ermöglichte uns das Projekt einen spannenden und vielversprechenden Einblick in neue Techniken, festigte Programmierkenntnisse und forderte die Einarbeitung in neue Komponenten und Praktiken.

## Angestrebte Verbesserungen

Im folgenden sollen einige der bei der Implementierung des Systems gekeimten Verbesserungen für die Zukunft konkretisiert werden.

- Kategorisierung nach Betriebssystem: Eine Applikation die für mehrere Betriebssysteme konzipiert wurde bspw. Adobe Photoshop), muss nicht zwangsläufig auf allen Betriebssystemen die gleichen Hotkeys verwenden.
- Kategorisierung nach Sprache: Die derzeitige Implementierung lässt das Mischen von Sprachen innerhalb einer HotkeyMap zu. Die hat zur Folge, dass innerhalb einer Map Identifier unterschiedlichen Sprachen angehören können. Ein Beispiel: User A erweitert eine Map M um den Key (strg x) mit dem Identifier: Ausschneiden. User B erweitert die Map M um den Key (strg v) Identifier: Insert usw. Man sieht an diesem Beispiel das Maps nicht an eine Sprache gebunden sind.

- Erleichtern der Hotkey Erstellung: Das einfache Abfangen der eingegebenen Tastenkombination, vergleichbar mit einem Keylogger, würde die Eingabe selbiger effizienter und weniger fehleranfällig gestalten, als eine manuelle Eingabe. Zudem wäre dies Sprachen unabhängig, da nur die Tastencodes geloggt werden und nicht Wörter in einer Sprache (bspw. Englisch) eingetippt werden müssen.

## Ausblick

Nach intensiver Beschäftigung und Implementierung des Projekts, sind weitere Verbesserungen und Ideen gekeimt. Eine weitere Ausarbeitung und Erweiterung des Systems ist in Planung. Eine Veröffentlichung unter einer freien Lizenz über Github oder einem alternativen Portal zur Veröffentlichung von OpenSource-Software ist nicht ausgeschlossen. Wir erhoffen uns dadurch Verbesserungen durch die Community der freien Softwarebewegung, durch die jeder profitiert, die Nutzer sowie die Entwickler des Systems. Wissen und Teilhabe sollte kein Privileg sein, sondern jedem zugänglich gemacht werden.

Es ist ein großer Vorteil im Leben, die Fehler, aus denen man lernen kann, möglichst früh zu begehen.

— Winston Churchill

Das Team HotkeyApp

T. Dunker, S. Herrmann, T. Meier, A. Stahl, S. Voigt