

**Philosophische** Fakultät III

**Sprach-, Literatur- und Kulturwissenschaften**

**Institut für Information und** Medien**, Sprache und Kultur (I:IMSK)  
Lehrstuhl für Medieninformatik**

Einführung in die Anwendungsprogrammierung

Modul: MEI-M03.3

Sommersemester 2017

URGaining

Jannik Bikowski, Thomas Fischer, Lisa Sanladerer, Titus Tissot

1768542, …, …, …

Medieninformatik / Informationswissenschaft, Medieninformatik / Informationswissenschaft, Medieninformatik / Kunstgeschichte, Medieninformatik / Geschichte,

5. Semester B.A., 5. Semester B.A., …, 4. Semester B.A.

Jannik Bikowski

84184 Ast

Tel.: 015780466999

Email: jannik.bikowski@stud.uni-regensburg.de

Abgegeben am 24.09.2017

Inhalt

1. Kurzbeschreibung 4
2. Technische Vorrausetzungen 5
3. Problemstellung und Lösungsvorschlag 6
4. Design & Implementierung 8
   1. User Interface 9
   2. Implementierung 9
5. Testing 11
6. Finaler Zustand und Ausblick 12
7. Projektmanagement 13

# Kurzbeschreibung

URGaining ermöglicht dem Nutzer ein übersichtliches und intuitives Trainings- und Workout-Management. Die leichte Bedienung und das vertraute Interface helfen fitnessbegeisterten

Nutzern, ihre Erfolge beim Training zu speichern und gibt mit entsprechenden Tabellen stets einen guten Überblick über absolvierte Workouts. Neue Trainingeinheiten mit zugehörigen Übungen sind schnell und einfach erstellt und werden für individuelle Statistiken auf einer lokalen Datenbank gespeichert. Mit der Facebook-Einbindung lassen sich Trainingserfolge mit Freunden teilen.

# Technische Vorrausetzungen

Nennen und beschreiben Sie in diesem Kapitel die technischen Voraussetzungen die für den Betrieb bzw. die Installation Ihrer App notwendig sind. Sie sollten in diesem Kapitel darlegen, welche externen Anwendungen (Server) für die Nutzung Ihrer Applikation nötig sind und wo bzw. wie diese heruntergeladen, eingerichtet und gestartet werden können. Mit diesen Informationen muss es möglich sein, Ihre Anwendung in vollem Umfang zu testen und etwaige Abhängigkeiten aufzulösen bzw. notwendige Infrastruktur-Anwendung in Betrieb zu nehmen. Zusätzlich erwähnen Sie hier externe Bibliotheken und Frameworks, die Sie für die Entwicklung Ihrer Anwendung genutzt haben. Geben Sie dabei auch an, für welche Funktionen oder Komponenten die jeweiligen Bibliotheken eingesetzt werden.

*URGaining* ist eine speziell für mobile Android-Geräte entwickelte Anwendung, die ohne eigene Server-Infrastruktur funktioniert. Für die Nutzung werden deshalb ausschließlich Android-Smartphones bzw. -Tablets empfohlen, die minimale API-Version ist 19 (4.4 KitKat).

Um auf die Share-/Teilen- bzw. Login-Funktion zugreifen zu können, ist sowohl ein Facebook-Account, als auch eine Internetverbindung notwendig.

# Problemstellung und Lösungsvorschlag

Oft sieht man im Fitnessstudio Trainingsenthusiasten mit Notizblock und Stift umherlaufen, damit sie ihre Fortschritte genauestens dokumentieren können. Doch da in der heutigen Zeit beinahe jeder ein eigenes Smartphone besitzt, bietet es sich an diese Fortschritte innerhalb einer App festzuhalten.

Mit URGaining hat der Nutzer während des Trainings stets eine Übersicht über seine Leistungen aus dem vergangenen Training und hat so einen Anreiz diese bei seinem aktuellen Workout zu übertreffen. Sollte ihm das Workout einmal zu schwer fallen kann er sich mit der Deload Funktion über einen vorher festgelegten Wert ein passendes Arbeitsgewicht für die einzelnen Übungen errechnen lassen. Dies erspart im Zeit, die er sonst in externen Apps aufwenden würde.

Außerdem bietet die App dem Nutzer eine Übersicht über seinen Gesamtfortschritt der einzelnen Übungen, gemessen an der errechneten Maximalkraft, in Form einer Grafik. Sollte der Nutzer eine befriedigende Steigerung erzielt haben, kann er diese Grafik direkt mit seinen Freunden auf Facebook teilen.

Desweiteren hat der Nutzer die Möglichkeit eine oder mehrere Übungen in die Berechnung seiner kombinierten Maximalkraft miteinzubeziehen. So haben z.B. Powerlifter auf der Startseite eine Übersicht über ihre Maximalkraft bestehend aus den Disziplinen „Kniebeugen“, „Bankdrücken“ und „Kreuzheben“, oder bei Gewichthebern „Reißen“ und „Stoßen“.

# Design & Implementierung

In diesem Kapitel sollen Sie die grundsätzlichen Konzepte der Designphase und Ihr Vorgehen während der Implementierung der Anwendung darlegen. Beschreiben Sie dazu die Architektur, die Sie sich für die Anwendung überlegt haben und dokumentieren Sie, wie Sie bezüglich der Implementierung der verschiedenen Funktionen vorgegangen sind. Dem Leser sollte deutlich werden, welche Komponenten und Module Sie für die Anwendung geplant haben und wie Sie bei deren Umsetzung vorgegangen sind. Beschreiben Sie dazu, aus welchen Activities bzw. Ansichten Ihre Anwendung besteht, welche Aufgabe diese einzelnen Komponenten haben und wie die Kommunikation bzw. Interaktion zwischen diesen Bestandteilen aussieht. Erwähnen Sie dabei auch mögliche – externe – Datenquellen und deren Anbindung. Beschreiben Sie die wichtigsten Klassen die Sie für die Implementierung der verschiedenen Funktionalitäten eingesetzt bzw. geschrieben haben und erwähnen Sie mögliche Probleme und Lösungsansätze. Beschreibend und begründen Sie kurz, in welchen Reihenfolge die verschiedenen Komponenten der Anwendung umgesetzt wurden.

Die Anwendung besteht aus drei Activities. Eine Übersichtsdarstellung (OverviewActivity) dient als Einstiegspunkt der App und stellt den nächsten Entsorgungstermin für die als Favorit markierte Straße dar. Durch einen Wisch über den Bildschirm wechselt der Nutzer zu eine weitere Activity (StreetListActivity), die die Suche nach verschiedenen Straßen erlaubt. Die angezeigt List enthält – alphabetisch sortiert – alle verfügbaren Straßen. Über ein einblendbares Suchinterface kann gezielt nach einem Straßennamen gesucht werden. Die angezeigt Liste wird dabei – automatisch – in Abhängigkeit der jeweiligen Nutzereingabe gefiltert. Durch den Klick auf einen Listeneintrag gelangt der Nutzer zu einer Detailansicht der ausgewählten Straße (StreetDetailActivity), die die Entsorgungstermine für die jeweilige Straße enthält. Diese Termine sind aufsteigend sortiert und in einer Liste angeordnet. Jedes Item der List enthält den entsprechenden Termin sowie die Art des Abfalls (Restmüll, Papier, ...). Über ein Icon der Actionbar kann die aktuell angezeigt Straße als Favorit gesetzt werden. [...]

Neben den sichtbaren Komponenten verfügt die Anwendung über eine Reihe von Modulen, die zur Beschaffung der nötigen Daten eingesetzt werden. Eine zentrale Kontroller-Einheit (DataController) kapselt dabei den Zugriff auf die Entsorgungsgtermine und sorgt für deren Aktualisierung. Andere Komponenten können über ein Listener-Pattern an diesen Kontroller angebunden werden. Der Kontroller baut die Verbindung mit den nötigen Internetseiten auf und verarbeitet die angeforderten Webseiten mit den Entsorgungsterminen. Die ge-*parsten* Inhalte werden in JAVA-Objekte überführt und über die Listener-Schnittstelle zur weiteren Verwendung innerhalb der Anwendung bereit gestellt. [...]

## User Interface

Beschreiben Sie an dieser – oder einer anderen sinnvollen – Stelle den Aufbau des User Interfaces. Aus welchen Komponenten (Views) bestehen die einzelnen Activities und wie kann der Benutzer mit Ihnen interagieren?

Abbildung 1: Startbildschirm der Anwendung

## Implementierung

Die wesentliche Aufgabe der Anwendung besteht in der Bereitstellung der Entsorgungsdaten. Initial wurde daher die Verarbeitung bzw. die Beschaffung dieser Daten getestet und implementiert.

Die nötigen Daten konnten auf der Website der Firma Meindl identifiziert werden. Im Wesentlichen finden sich die nötigen Angaben auf zwei Unterseiten des Internetauftritts. Eine Liste[[1]](#footnote-2) enthält alle Straßennamen, die von dem Entsorgungsbetrieb angefahren werden. Der HTML-Code dieser Seite beinhaltet für jede Straße einen Link zu einer Detailansicht[[2]](#footnote-3), auf der die eigentlichen Entsorgungsdaten aufgeführt werden. In einem ersten Schritt wurde die Übersichtseite (Straßenliste) bzw. der HTML-Code dieser Seite geladen. Dazu wurde die entsprechende URL über einen HTTP-Request (über den HTTP-Client des Apache-Pakets) aufgerufen. Der zurückgegebene HTML-Code der Website konnte anschließend durch die Verwendung der jsoup-Bibliothek weiter verwendet werden. Eine vorherige Analyse der HTML-Struktur zeigte, das die Liste der Straßennamen bzw. deren einzelne Bestandteile über verschiedene CSS-Klassen identifiziert werden können. Ein entsprechend implementierter Parser konnte über die einzelnen Straßen iterieren und die wesentlichen Bestandteile herausfiltern. Der Straßenname, die Postleitzahl sowie eine eindeutige ID – vergeben durch die Website – wurden in eine eigene JAVA-Repräsentation (Street-Klasse) überführt. Der Link zu der detaillierten Auflistung der Entsorgungstermine ist für alle Straßen gleich und unterscheidet sich nur durch die gespeicherte ID. Als Ergebnis dieses ersten Schrittes konnte eine Liste alle Regensburg Straßen - in Form von JAVA-Objekten – erstellt werden, für die Entsorgungsdaten vorliegen. [...]

In einem weiteren Schritt wurde die Detailansicht der jeweiligen Straßen verarbeitet. Über die bekannte URL – mit ergänzter, individueller Straßen-ID – wurde der HTML-Code bezogen und durch Einsatz der jsoup-Bibliothek verarbeitet. [...]

Nach erfolgreichem Test der Datenanbindung wurde die erarbeitet Funktionalität in wiederverwendbarem, integrierbarem Code umgesetzt. Zentrale Komponenten dazu ist der DataController, der den Aufruf und die Verarbeitung der verschiedenen Webseiten steuert und die erstellen Datensätze (Straßen und jeweilige Entsorgungsdaten) über ein Interface bereitstellt. Andere Komponenten können den Kontroller instanziieren; über den Konstruktor wird dabei eine Referenz auf einen OnDataChangedListener (Interface) übergeben. Über öffentliche Methoden des Kontrollers kann die Aktualisierung der Straßen-Liste (update()) sowie die Beschaffung von Entsorgungsterminen für eine bestimmte Straße (fetchGarbageDaysFor Street(Street street)) ausgelöst werden. Der Listener wird vom Kontroller benachrichtigt, wenn eine Aktualisierung der Datensätze erfolgreich war. Über die entsprechenden Methoden des Interfaces werden Listen an verfügbaren Straßen (Street) bzw. Entsorgungstermine (GarbageDay) übermittelt. [...]

Nach der Fertigstellung der Komponenten zur Datenbeschaffung wurde die Benutzer-Schnittstelle der Anwendung implementiert. Dazu wurden die geplanten und entworfenen Activities umgesetzt. [...] Für die Darstellung der Straßenliste innerhalb der StreetListActivity wurden ein Adapter sowie ein XML-Layout entworfen, der eine Liste von Street-Objekten mit einem ListView verknüpft und die relevanten Inhalte darstellt. [...]

# Testing

Beschreiben Sie die wesentlichen Probleme, die durch den Usability-Test aufgedeckt wurden. Erläutern Sie kurz, welche Maßnahmen ergriffen wurden, um diese Problem zu beheben und geben Sie an, im welchem Umfang die identifizierten Schwachstellen ausgebessert wurden. Beschreiben Sie, wie Sie Ihre Anwendung angepasst haben, um die identifizierten Problem zu lösen. Dabei sollten Sie erwähnen welche Komponenten oder Klassen Ihren App betroffen sind und wie sich die Änderungen auf Ihr ursprüngliches (Interaktions-)Konzept ausgewirkt haben.

# Finaler Zustand und Ausblick

Beschreiben Sie kurz den finalen Zustand Ihrer Anwendung. Erläutern Sie den implementierten Funktionsumfang und begründen Sie fehlende Features, die in den Anforderungen erwähnt wurden, in der finalen Version jedoch nicht verfügbar sind. Überlegen Sie, inwieweit Ihre Anwendung durch zusätzliche Funktionen sinnvoll erweitert werden könnte.

# Projektmanagement

Beschreiben Sie kurz, wie Sie bezüglich der gruppeninternen Abstimmung vorgegangen sind. Erwähnen Sie Tools und Kommunikationswege, die Sie für die Gruppenarbeit eingesetzt haben. Sie können hier auch einen kurzen Überblick über die Arbeitsaufteilung geben und explizit erwähnen, welche Bestandteile von welchen Teammitgliedern umgesetzt wurden.

1. http://www.entsorgungsdaten.de/kalender/ [↑](#footnote-ref-2)
2. Beispiel: http://www.entsorgungsdaten.de/module.php5?fid=15&gem=30&mod=entsorgungsdaten&ot=1047 [↑](#footnote-ref-3)