MVHS Zeiterfassung

Unterlagen zum Kurs
„Programme entwickeln für Android Smartphones“
an der Münchner Volkshochschule

Ersteller: Dipl.-Ing. (FH) Eugen Richter (WebDucer – IT & Internet Service)

Inhaltsverzeichnis

Geschichte von Android

Quelle: http://de.wikipedia.org/wiki/Android_(Betriebssystem)
Stand: 13.04.2011

Die Geschichte von Android beginnt im Jahr 2005 mit der Übernahme der Firma Android von Andy Rubin durch Google. Diese Firma war auf mobile Anwendungen mit Schwerpunkt auf standortbezogene Dienste. Im Jahr 2007 gründete Google mit weiteren 33 Partnern die Open Handset Alliance und kündigte die Entwicklung eines neuen mobilen Betriebssystems an. Im Oktober 2008 erschien das erste Android Mobiltelefon (HTC Dream / T-Mobile G-1).

Seit dem wurde das Android Betriebssystem weiter verbessert und neue Funktionen erweitert.

Versionsgeschichte

Version (Name)	Freigabedatum
1.1	10. Februar 2009
1.5 (Cupcake)	30. April 2009
1.6 (Donat)	15. September 2009
2.0 (Eclair)	26. Oktober 2009
2.1 (Eclair)	12. Januar 2010
2.2 (Froyo)	20. Mai 2010
2.3 (Gingerbread)	6. Dezember 2010
2.3.3 (Gingerbread)	23. Februar 2010
3.0 (Honeycomb)	23. Februar 2010 (nur Tablets)
3.1 (Ice Cream)	2011 (Zusammenführung des Codes von 2.3.3 und 3.0

Versionen

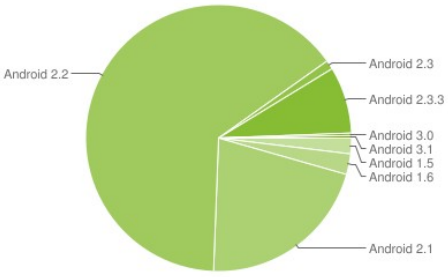
Versionsunterschiede (Features)

Version (Name)	Datum	Neue Features	Status
1.1	10.02.2009	• Speichern von MMS-Anhängen	unsupported
1.5 (Cupcake)	30.04.2009	• **Automatischer Wechsel zwischen Hoch- und Querformat** • **Bildschirm-Tastatur** • Aufnahme und Wiedergabe von Videos • Automatisches Verbinden und Stereo für Bluetooth • Weitere Sprachen neben Englisch und Deutsch	Supported
1.6 (Donut)	15.09.2009	• Virtual Private Networks konfigurierbar • **Differenzierte Energieverbrauchssteuerung** • Suchfunktion quellenübergreifend und selbstoptimierend • Text-to-Speech, Gestenerkennung und mehr als eine Bildschirmauflösung	Supported
2.0 (Eclair)	26.10.2009	• Digitalzoom und Unterstützung von Blitzlicht • **Unterstützung von Microsoft Exchange** • Bluetooth 2.1	Supported
2.1 (Eclair)	12.01.2010	• Animierte Hintergrundbilder • Informationen zur Signalstärke • **Erweiterungen von Webkit (HTML5-Unterstützungen, WebStorage, Geolocation, Video …)**	Supported
2.2 (Froyo)	20.05.2010	• Linux-Kernel 2.6.32, der weniger Arbeitsspeicher benötigt • Verwendung von Arbeitsspeicher, der größer als die bisher nutzbaren 256 MByte ist. • **JIT-Compiler, Erweiterungen für OpenGL ES 2.0, Unterstützung von Flash 10.1** • **Tethering** • **Speicherbarkeit von Apps auf der SD-Karte (App2SD)** • Android Cloud to Device Messaging Framework: Möglichkeit PUSH in die eigenen Anwendungen zu implementieren. • Bluetooth-Sprachwahl	Supported
2.3 (Gingerbread)	06.12.2010	• Linux-Kernel 2.6.35.7 • **Unterstützung von WebM** • **Unterstützung von HTML5 Audio** • Unterstützung von Google TV • **Unterstützung von Near Field Communication**	Supported

Version (Name)	Datum	Neue Features	Status
		• **Parallele Garbage Collection für ruckelfreie Animationen** • Verbesserte Integration von sozialen Netzwerken • **Unterstützung von Gyroskopen (nicht zu verwechseln mit Bewegungssensoren) und anderen Sensoren (u.a. Barometer, Schwerkraftsensor)** • **Integrierter SIP-Client für VoIP** • Integrierter Downloadmanager • **Unterstützung des Ext4-Dateisystems**	
2.3.3 (Gingerbread)	23.02.2011	• **Dual-Core-Unterstützung** • **Unterstützung von Dual-Core-Apps auf Single-Core-Geräten** • Verbesserte Unterstützung der NFC-Technik • Verbesserte Bluetooth-Unterstützung • Kleinere Verbesserungen	Current
3.0 (Honeaycomb)	23.02.2011	• **Benutzerfreundlichere Oberfläche** • **Verbesserte Unterstützung für Tablet-Computer** • Google Talk mit Videotelefonie • Neue Browser-Funktionen: Synchronisierung der Lesezeichen mit Google Chrome, Tabs, automatisches Ausfüllen von Formularen und Inkognito-Modus beim Surfen	Current
3.1 (Ice Cream)	2011	• Zusammenführung der Entwicklungslinien 2.x und 3.x und Google TV	Future

Versionsverbreitung

Quelle: http://developer.android.com/resources/dashboard/platform-versions.html
Stand: 1. Juni 2011

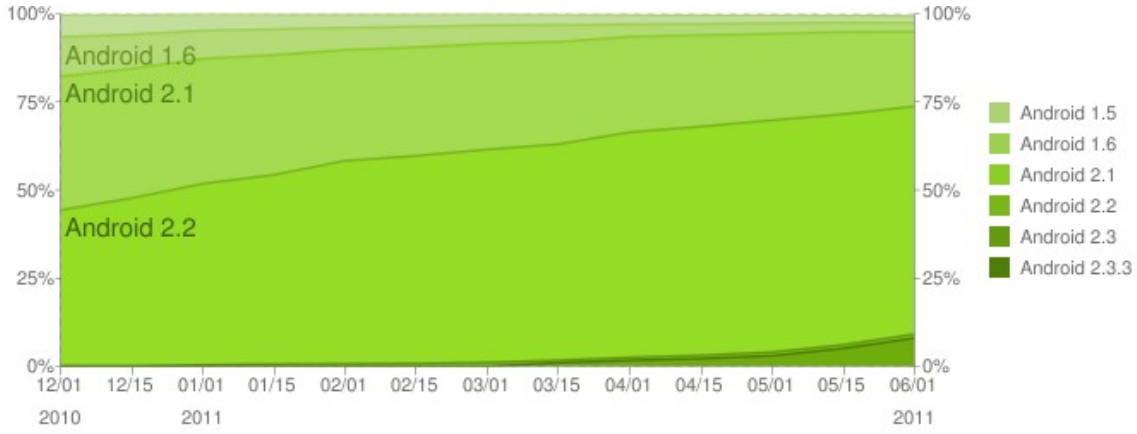


Unter http://developer.android.com/resources/dashboard/platform-versions.html liefert Google Statistiken zur Verbreitung der unterschiedlichen Versionen. Die Daten werden durch die Zugriffe auf Android Market zusammengetragen. Somit beinhaltet diese Statistik keine Geräte, die keinen Zugriff auf Android Market haben (z.B.: 1&1 Tablet, der Zugriff nur auf 1&1 AppStore hat und weitere günstige Geräte).

Version	API Level	Verbreitung
Android 1.5	3	1.90%
Android 1.6	4	2,50%
Android 2.1	7	21,20%
Android 2.2	8	64,60%

Version	API Level	Verbreitung
Android 2.3	9	1,10%
Android 2.3.3	10	8,10%
Android 3.0	11	0,30%
Android 3.1	12	0,30%

Verlauf der Verbreitung



Hardware

Sensoren

- GPS
- Kompass
- Gyroskop
- Beschleunigungssensor
- Lagesensor
- NFC
- Annäherungssensor
- Kamera
- Mikrofone

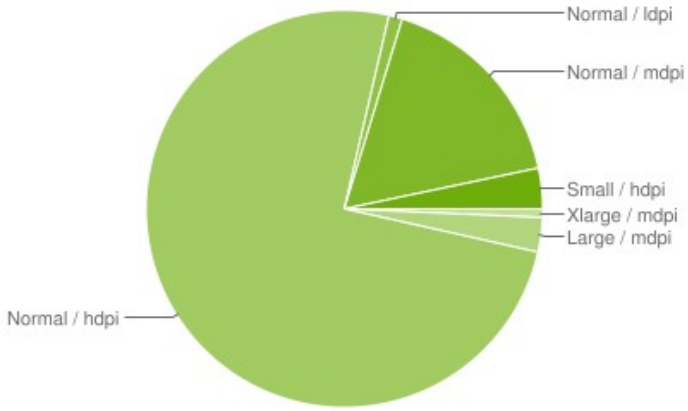
Auflösungen

- QVGA (320 x 240) [Low Desitiy]
- HVGA (320 x 480) [Midle Density]
- WVGA (800 x 480) [Hight Density]
- WXGA (1024 x 600 / 1280 x 800) [High Density] Tablets

Auflösungsverbreitung

Quelle: http://developer.android.com/resources/dashboard/screens.html
Stand: 01.Juni 2011

	ldpi	mdpi	hdpi	xhdpi
Small			3,30%	
Normal	1,10%	17,00%	75,20%	
Large		2,80%		
xLarge		0,70%		



Prozessoren

- ARM Prozessoren mit 400 – 1GHz (Single Core)
- ARM Prozessoren mit 600 – 1,6 GHz (Dual Core)

Verbreitungswege

Die Android Apps können im Gegensatz zu Apple Apps von jeder Quelle bezogen werden. Es gibt keine Pflicht das Google Market zu benutzen. Aus diesem Grund etablieren sich langsam auch alternative Markets, die unter anderem andere Vergütungsmodelle für Programmierer und andere Bezahlmethoden anbieten.

Weiterhin bieten einige Smartphone Hersteller (entweder nur diesen oder parallel zu Google Market) herstellerspezifische Markets (z.B.: für Samsung Galaxy Tab).

Folgende Bezugsquellen für Apps sind möglich (kein Anspruch auf Vollständigkeit):

- Google Market
- Amazon AppStore (nur USA momentan)
- Androidpit
- Entwickler-Homepage
- Eigener PC/Mac

Bei allen Quellen, außer natürlich Google Market, muss auf dem Android Gerät der Bezug von Apps aus unbekannten Quellen aktiviert sein (Einstellungen → Anwendungen).

Bei den kostenpflichtigen Apps bekommt der Entwickler (bei Google und Amazon) 70% der Einnahmen und 30% der Betreiber des Markets.

Weitere Verdienstmöglichkeiten ergeben sich durch die Werbeeinblendungen in den Apps (z.B. durch AddMobile – gehört mittlerweile auch zu Google).

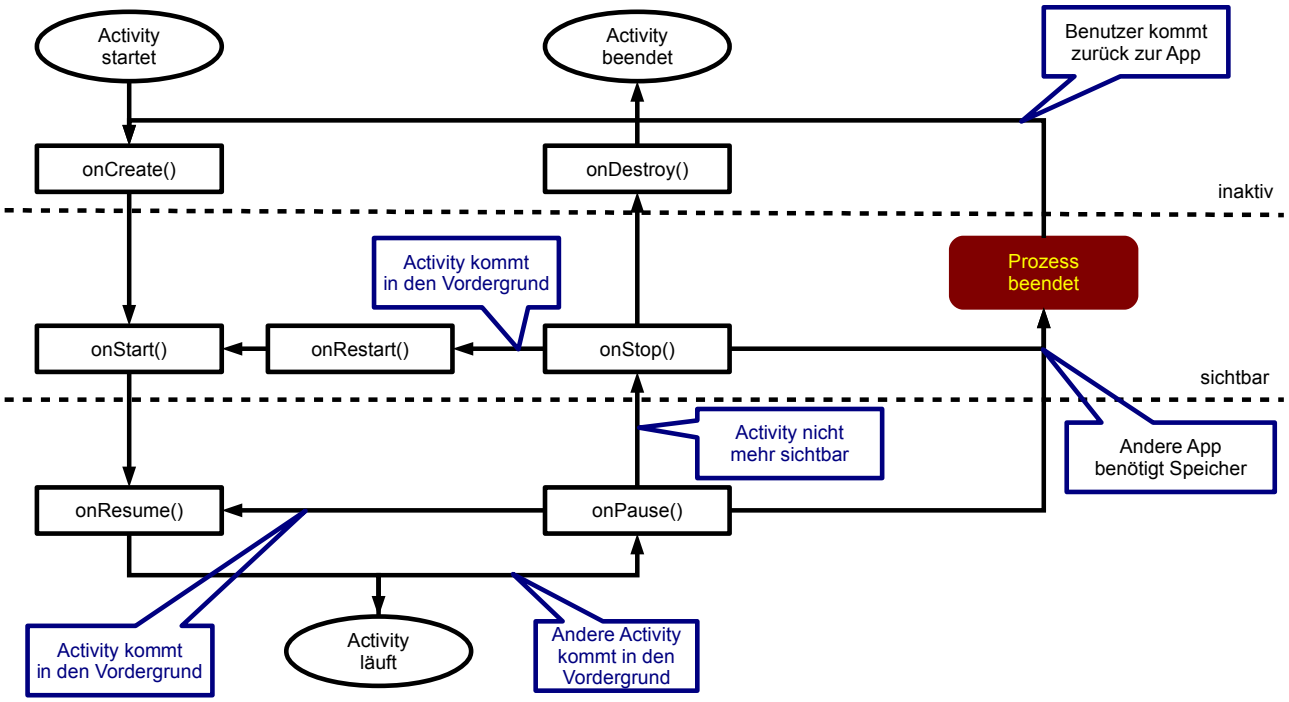
Programmierung

Die Entwicklung von Android Apps erfolgt grundsätzlich in Java. Android unterstützt dabei Java 1.5 komplett und das Android SDK liefert zusätzlich noch spezielle Bibliotheken, die an das mobile System angepasst wurden (Bluetooth, JSON, XML.Sax, JUnit, W3C.DOM). Die Apps laufen auf dem Gerät (und im Emulator) in einer von anderen Apps abgeschotteten Sandbox. Die eingesetzte JVM wurde von Google neu entwickelt und an die Architektur eines mobilen Systems mit wenig Ressourcen angepasst. Die JVM heißt unter Android Darvik VM diese kann keinen nativen Java-Code ausführen, genauso wenig wie die Oracle JVM den Darvik Byte-Code nicht ausführen kann. Beim Kompilieren eines Android Projektes wird aus dem Java-Byte-Code (Kellerautomat) ein Darvik-Byte-Code (Registerautomat) erstellt.

Nur Performance lastige Anteile wurden in Android in C/C++ erstellt (Treiber, Mediencodecs, Webbrowser Engine, SQLite und OpenGL). Diese Möglichkeit ist auch für die Entwicklung eigener Apps gegeben, und sollte nur dann verwendet werden, wenn es Performance-Engpässe gibt. Da für C/C++ Code kein Garbage Collector vorhanden ist, muss der Programmierer sich selbst um die Freigabe der Ressourcen kümmern.

Neben den eigentlichen Apps sind auch die sogenannten Widgets möglich. Diese können einen schnellen Zugriff auf die Daten Ihrer App gewähren, ohne dass der Benutzer die komplette App starten muss. Eine gut durchdachte Erweiterung einer App um ein oder mehrere Widgets kann den Ausschlag zur Konkurrenz geben.

Lebenszyklus der Activity



MVHS Zeiterfassungs-App

Aufgabenstellung

Es soll eine App für die Arbeitszeiterfassung erstellt werden, die auf den Android-Geräten ab Version 1.6 lauffähig ist. Die App soll in mehreren Stufen durch das Hinzufügen von neuen Features aufgebaut werde.

Tag 1

Vorgesehene Features / Schritte

- Anlegen eines neuen Android-Projektes
- Erstellung des Layouts für die Zeiterfassung
- Erstellung der Logik für die Zeiterfassung
- Planung der Datenbank (mithilfe der Firefox-Erweiterung SQLite Manager)

Anlegen eines Android-Projektes

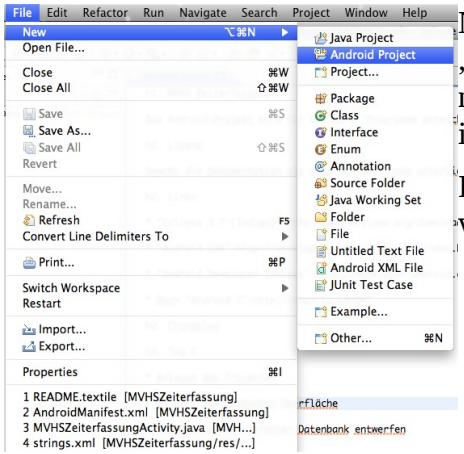


Abb. 1: Neues Android Projekt

Neues Android-Projekt wird über das Menü „File“ → „New“ → „Android Project“ angelegt (Abb. 1). Sollte dieser Menüpunkt nicht sichtbar sein, gehen Sie auf „Others“ und wählen Sie dann im folgenden Dialog „Android Project“.

Im folgenden Assistenten müssen folgende Angaben gemacht werden (Abb. 3 und 4):

- Project name (Projektname ohne Sonderzeichen)
- Build Target (Compiler-Version, mit der die App kompiliert wird)
- Application Name (Bezeichnung der App, so wie diese im Menü des Smartphones erscheinen soll)

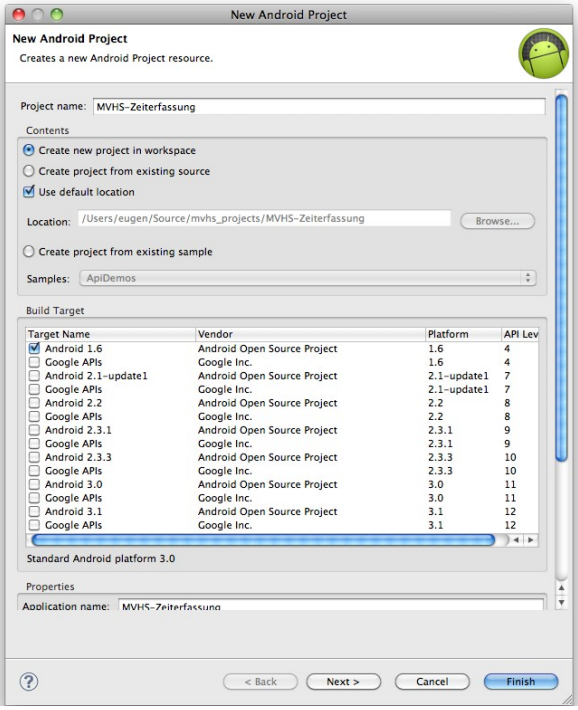


Abb. 2: Angaben für neues Projekt (1)

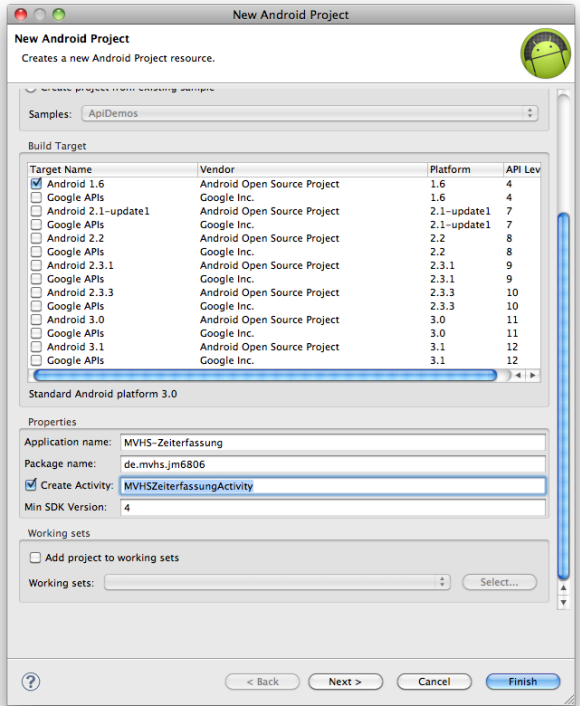


Abb. 3: Angaben für neues Projekt (2)

- Package name (Hierarchische Bezeichnung des Java-Packages. Of als umgekehrte URL angegeben)

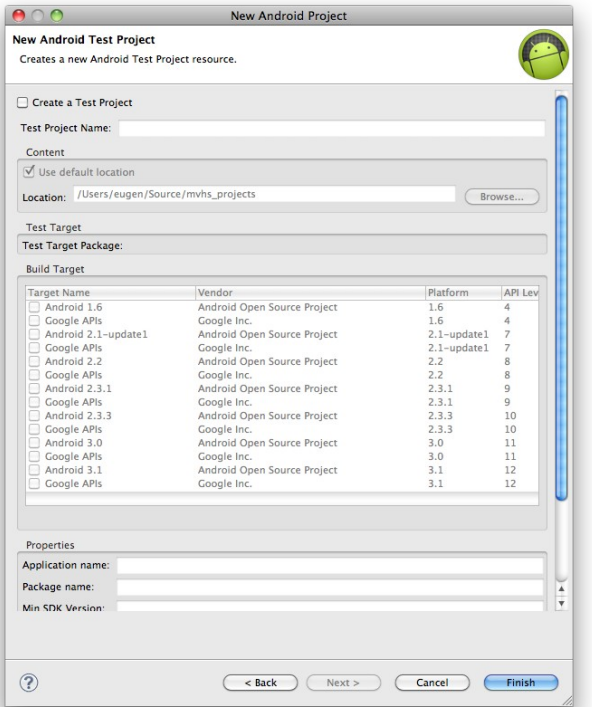


Abb. 4: JUnit Test-Project

- Min SDK version (minimale Anforderungen an das Gerät)

Für komplexere Apps ist es empfehlenswert zur App auch TestUnits zu schreiben, um bei neueren Versionen die Funktionalität automatisiert testen zu können. Das kann im nachfolgenden Dialog mit erledigt werden (Abb. 4).

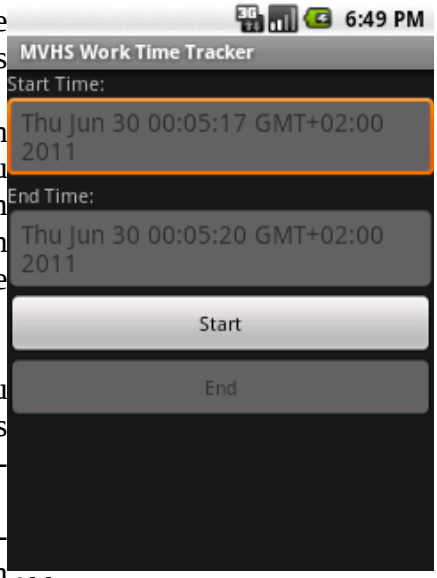


Abb. 5: Layout

Layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
<TextView
    android:layout_width="fill_parent" android:layout_height="wrap_content"
    android:text="@string/start_time"/>
<EditText
      android:id="@+id/txtStartTime" android:layout_width="fill_parent"
      android:layout_height="wrap_content">
</EditText>
<TextView
      android:text="@string/end_time" android:layout_width="wrap_content"
      android:layout_height="wrap_content">
</TextView>
<EditText
      android:id="@+id/txtEndTime" android:layout_width="fill_parent"
      android:layout_height="wrap_content">
</EditText>
<Button
      android:id="@+id/btnStart" android:layout_width="fill_parent"
      android:layout_height="wrap_content" android:text="@string/button_start"
      android:onClick="onButtonClick">
</Button>
<Button
      android:id="@+id/btnEnd" android:layout_width="fill_parent"
      android:layout_height="wrap_content" android:text="@string/button_end"
      android:onClick="onButtonClick">
</Button>
</LinearLayout>
```

Logik

In der ersten Entwicklungsphase soll beim Betätigen des Knopfes „Start“ die aktuelle Zeit in das erste Textfeld ausgegeben werden und der „End“-Knopf aktiviert werden. Beim Betätigen des Knopfes „End“ soll die aktuelle Zeit in das zweite Textfeld ausgegeben werden und der „Start“-Knopf wieder aktiviert werden.

Datenbank

Feldbezeichnung	Typ	Beschreibung
_id	Integer	Primärschlüssel des Datensatzes
start_time	Text	Startdatum des Datensatzes in Form „YYYY-MM-DD HH:mm:ss“ (z.B.: 2011-06-29 19:03:53)
end_time	Text	Enddatum des Datensatzes in Form „YYYY-MM-DD HH:mm:ss“ (z.B.: 2011-06-29 19:23:12)

Tag 2

Vorgesehene Features / Schritte

- Einführung in SQLite
- Implementierung der DB-Schnittstelle
- Persistente Zeiterfassung (Speichern der Daten in der DB)
- Zugriff auf die Daten auf den Emulator / Smartphone über DDMS-Perspektive

SQLite unter Android

Für den Zugriff auf die SQLite-Datenbank steht unter Android die abstrakte Klasse `SQLiteOpenHelper` zur Verfügung. Diese Klasse regelt mit den Methoden `onCreate` und `onUpdate` die Verwaltungsmethoden die Erzeugung und Aktualisierung der Datenbank.

Die SQL-Anweisungen können mit dem DB-Parameter aus dem Helper (`SQLiteOpenHelper.getReadableDatabase()` oder `SQLiteOpenHelper.getWritableDatabase()`) auf vier Arten durchgeführt werden.

1. `db.rawQuery`: Als Parameter wir der auszuführende SQL-String übergeben.
2. `db.query`: Als Parameter werden dabei die Bestandteile der Abfrage übergeben.
3. Prepared Statements: Vorkompilierte SQL Anweisungen (sehr performant im Vergleich zu den beiden anderen Methoden).
4. Spezielle Änderungsfunktionen (update, delete, insert).

Abfrageoperationen

SQLite rawQuery	SQLite query

`SELECT`
`  _id,`
`  start_time,`
`  end_time`
`FROM`
`  time_tracking`
	`db.query(`
	`  false, // distinct?`
	`  ”time_tracking” // From Tabelle`
	`  new String[] { // Select Spalten`
	`  ”_id”,`
	`  ”start_time”,`

```
WHERE
      _id < 100
ORDER BY
      start_time DESC
```

```
      "end_time"},
      "_id <", // Where-Bedingung
      new String[] { // Parameter für
Where
      "100"},
      null, // Group By Anweisung
      null, // Having Anweisung
      "start_time DESC", // Order By
      null); // Limit
```

Änderungsoperationen

Operation	SQLite rowQuery	Spezialfunktion
Insert	`INSERT INTO time_tracking (start_time, end_time) VALUES ('2011-07-06 08:00:00', '2011-07-06 17:00:00')`	`Content Values werte = new ContentValues(); werte.put( "start_time", "2011-07-07 07:00:00"); werte.put( "end_time", "2011-07-06 17:00:00"); db.insert( "time_tracking", null, werte);`
Update	`UPDATE time_trackin SET start_time = '2011-07-06 07:15:00', end_time = '2011-07-07 17:45:00' WHERE _id = 100`	`ContentValues werte = new ContentValues(); werte.put( "start_time", "2011-07-06 07:15:00"); werte.put( "end_time", "2011-07-06 17:45:00"); db.update( "time_tracking", werte, "_id=?", new String[]{"100"});`
Delete	`DELETE FROM time_tracking WHERE _id = 100`	`db.delete( "time_tracking", "_id=?", new String[]{"100"});`

Prepared Statements

Die Statements liefern als Ergebnis immer 1 x 1 Zelle. Somit sind diese nicht für „normale“ Abfragen geeignet, aber für spezielle Aufgaben, die nur einer einzigen Wert (z.B. „SELECT count(*) ...“) zurückliefern oder gar kein Ergebnis notwendig ist (Update, Delete, Insert). Diese Statements werden beim Kompilieren des Projektes für SQLite kompiliert und erreichen dadurch sehr hohe Performance (ca. 4 x schnelle als rowQuery Anweisungen).

Mögliche Statements sind:

- `execute()`: nicht für `SELECT`, `UPDATE`, `INSERT`, `DELETE`, aber für `CREATE`, `DROP` (Tabelle, View, Index) geeignet.
- `executeInsert()`: für `INSERT` Anweisungen geeignet.
- `executeUpdateDelete()`: Erst ab Android 2.2 verfügbar und für `UPDATE` und `DELETE` Anweisungen geeignet.
- simpleQueryForLong(): liefert einen Integer in einer 1 x 1 Tabelle als Ergebnis. Gut für „`SELECT count(*) FROM tabel`“ geeignet.

- `simpleQueryForString()`: Arbeitet genauso wie simpleQueryForLong, liefert aber statt Integer einen Strings-Wert zurück.

Beispiel

```
SQLiteStatement stmInsert = db.compileStatement(
      ”INSERT INTO time_tracking ” +
      ”(_id, start_time, end_time) ” +
      ”VALUES (?,?,?)”);
stmInsert.bindLong(1001);
stmInsert.bindString(”2011-07-06 07:00:00”);
stmInsert.bindString(”2011-07-06 17:00:00”);
long id = stmInsert.executeInsert();
```

Tag 3

Vorgesehene Features / Schritte

- Bereinigung bei "Start"-Drücken
- Lokales Datumsformat bei der Anzeige
- Listenansicht der gespeicherten Daten
- Menü
- Kontextmenü
  - Edit
  - Delete
  - New

Anhang

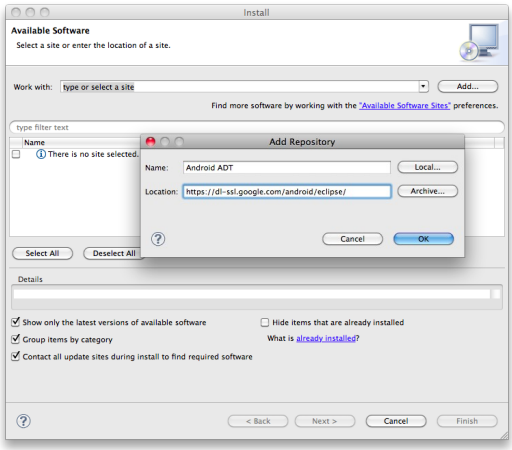
Installation von Eclipse IDE und ADT

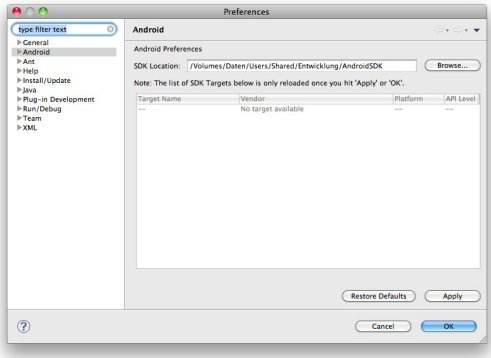
Installation von Eclipse

- Eclipse Classic ab Version 3.5 herunterladen (http://www.eclipse.org/downloads/).
- Die heruntergeladene Datei in einen Ordner entpacken (z.B.: `C:\Entwicklung\Eclipse` unter Windows, oder `/Users/Shared/Entwicklung/Eclipse` unter Mac OS X).
- Eclipse starten und das Standard-Workplace (z.B.: `C:\Entwicklung\Android` unter Windows oder `/Users/Shared/Entwicklung/Android` unter Mac OS X) vergeben.

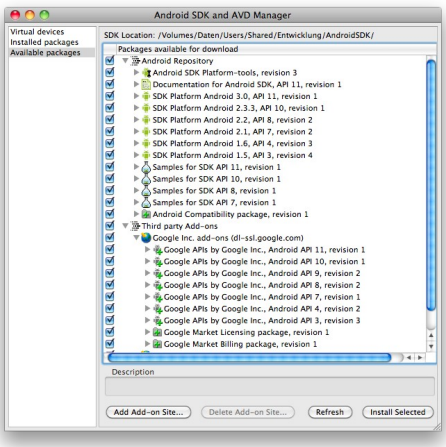
Installation von ADT

5. Android SDK für das gewünschte Betriebssystem herunterladen (http://developer.android.com/sdk/index.html).
6. Die heruntergeladene Datei in einen Ordner entpacken (z.B.: `C:\Entwicklung\AndroidSDK` unter Windows oder `/Users/Shared/Entwicklung/AndroidSDK` unter Mac OS X).
7. Eclipse starten und zu **Help → Install New Software** gehen.

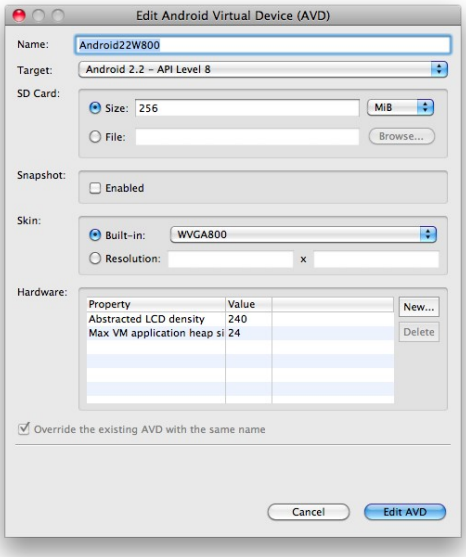




8. Bei „**Work with:**“ auf „**Add...**“ klicken, um neues Software-Repository hinzuzufügen (siehe Bild).
9. Namen frei vergeben und als „Location“ https://dl-ssl.google.com/android/eclipse/ eingeben https://dl-ssl.google.com/android/eclipse/ eingeben.
10. **OK** klicken.
11. Developer Tools als zu installierende Komponenten auswählen (es dauert einige Zeit, bis die Repository eingelesen wird).
12. Die Installation mit „**Next**“ → „**Next**“ → „**Accept**“ und „**Finish**“ abschließen.
13. Eclipse neu starten.
14. Gehen Sie zu Preferences (unter Winsdows über **Window → Preferences**, unter Mac OS X über **Eclipse → Prefenrences** zu erreichen.
15. Gehen Sie zum Punkt „Android“ und tragen Sie da



den Speicherort von Android-SDK ein (nach dem Vorschlag wäre es unter Windows der Ordner `C:\Entwicklung\AndroidSDK` und unter Mac OS X `/Users/Shared/Entwicklung/AndroidSDK`).



16. Nun gehen Sie zu „**Window**“ → „Android-**SDK and AVD Manager“ → „Available packages“ und installieren Sie die benötigten (z.**B.: Android 1.6 und Android 2.2) oder alle SDK-Versionen.
17. Nach dem Neustart von Eclipse gehen Sie wieder zu „Android-**SDK and AVD Manager“.**
18. Legen Sie unter „**Virtual Devices**“ die virtuellen Android-Geräte, für die Sie entwickeln möchten (z.B.: ein Android 2.2 Gerät mit WXGA Auflösung [800x480] und 2GB SD-Karte). Sie können auch mehrere virtuelle Geräte anlegen und beim Start eines Android-Projektes dann eins davon auswählen.
19. Damit sind alle Vorbereitungen für die Android-Programmierung getroffen.

Git – Versionsverwaltung unter Eclipse

Quelle: http://de.wikipedia.org/wiki/Git

Git (dt. [ɡɪt]) ist eine Freie Software zur verteilten Versionsverwaltung von Dateien, die ursprünglich für die Quellcode-Verwaltung des Linux Betriebssystemkerns entwickelt wurde.

Die Entwicklung von Git wurde im April 2005 von Linus Torvalds begonnen, um das bis dahin verwendete Versionskontrollsystem BitKeeper zu ersetzen, das durch eine Lizenzänderung vielen Entwicklern den Zugang verwehrte. Die erste Version erschien bereits wenige Tage nach der Ankündigung.

Git läuft auf fast allen modernen UNIX-artigen Systemen, wie Linux, Solaris, Mac OS X, FreeBSD, DragonFly BSD, NetBSD, OpenBSD, AIX und IRIX. Auf Microsoft Windows läuft es entweder mit Hilfe der Cygwin-Umgebung oder mit Msysgit, bzw. der seit einiger Zeit verfügbaren TortoiseGit-Shell-Erweiterung (ähnlich TortoiseSVN).

Die derzeit aktuelle Version wird produktiv für die Entwicklung des Linux-Betriebssystemkerns und im Rahmen vieler weiterer Projekte eingesetzt, darunter Git selbst, Samba, X.Org, Qt, GNOME, One Laptop per Child, Ruby on Rails, VLC, Wine, LilyPond, TaskJuggler, DragonFly BSD, Android, BusyBox, Amarok, Perl 5, x264 und phpBB Derzeitiger Maintainer von Git ist Junio Hamano.

Installation des Plug-ins

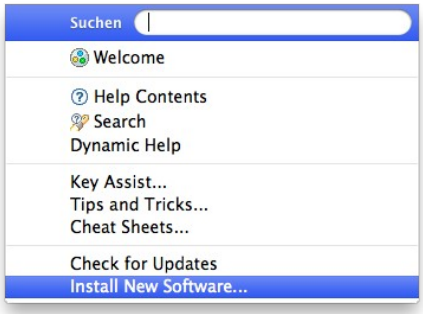


Abb. 6: Install New Software

Klicken Sie unter „Help“ auf den Punkt „Install New Software“ um zum Installationsdialog für Erweiterungen von Eclipse zu gelangen (Abb. 1).

Unter „Work with“ (Abb. 2) wählen Sie „--All Available Sites--“ um alle verfügbaren Quellen für Eclipse einzubeziehen. Die Aktualisierung der Liste kann abhängig von Rechenleistung und Internetverbindung mehrere Minuten dauern. Geben Sie in der Suchmaske „git“ ein. In der Ergebnisliste markieren Sie die Plug-ins JGit (Java-Implementierung des Git-Clients) und EGit (Team-Plug-in für Eclipse). Klicken Sie auf „Next“.

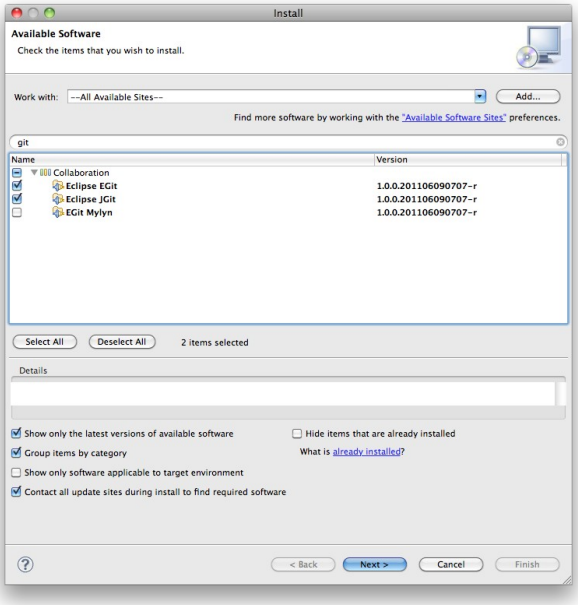


Abb. 7: Auswahl der Plug-ins

Nach der Überprüfung der Abhängigkeiten erscheint ein Bestätigungsdialog mit der Auflistung aller zu installierenden Komponenten (Abb. 3). Klicken Sie hier wieder auf „Next“.

Im nächsten Dialog (Abb. 4) bestätigen Sie die Lizenzvereinbarung mit „I accept the terms of the license agreement“ und dem Klick auf „Finish“.

Nach der Installation (Abb. 5) muss Eclipse neu gestartet werden (Abb. 6).

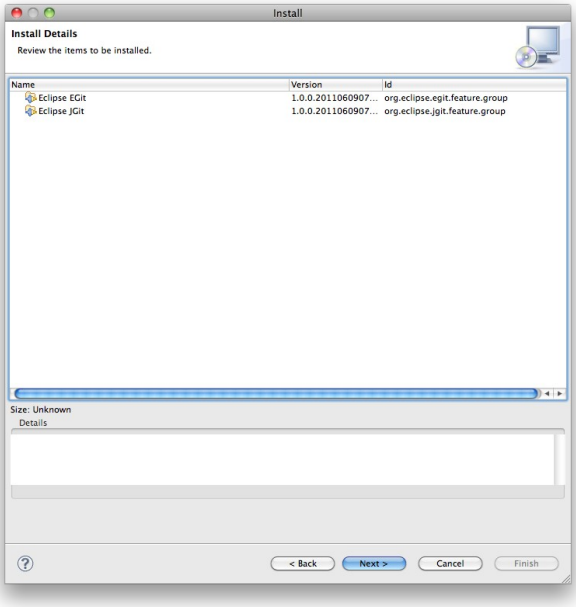


Abb. 8: Zu installierende Komponenten

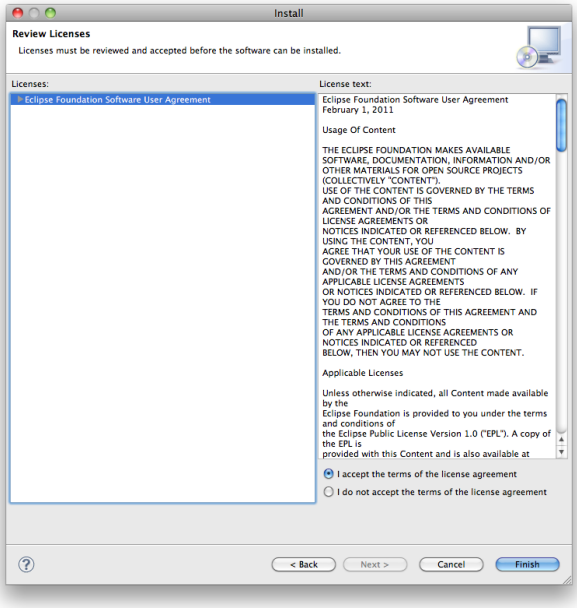


Abb. 9: Lizenzbestätigung

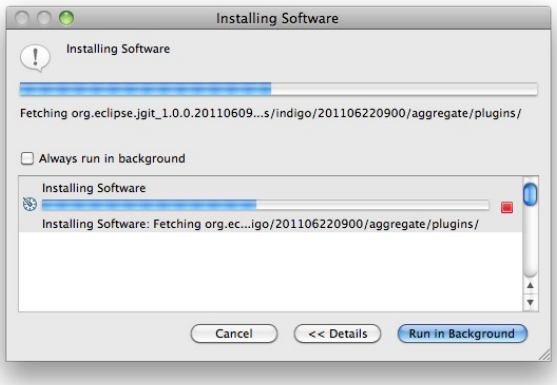


Abb. 10: Installation

Abb. 11: Neustart der Entwicklungsumgebung

Import eines Projektes

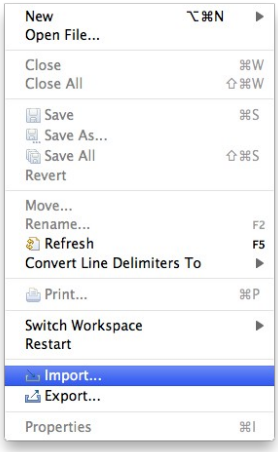


Abb. 12: Import

Zum Import eines Git-Projektes (lokal oder hier von GitHub) klicken Sie zuerst unter „File“ auf den Menüpunkt „Import“ (Abb. 7).

Als Quelle wählen Sie in der Liste unter „Git“ den Punkt „Projects from Git“ und klicken Sie auf „Next“ (Abb. 8).

Im nächsten Dialog (Abb. 9) klicken Sie auf „Clone“ um die Adresse des Git-Repositories anzugeben (Abb. 10). Die notwendige Adresse bei Github unter der Projektadresse (z.B. wie in Abb. 11 dargestellt).

Klicken Sie auf „Next“.

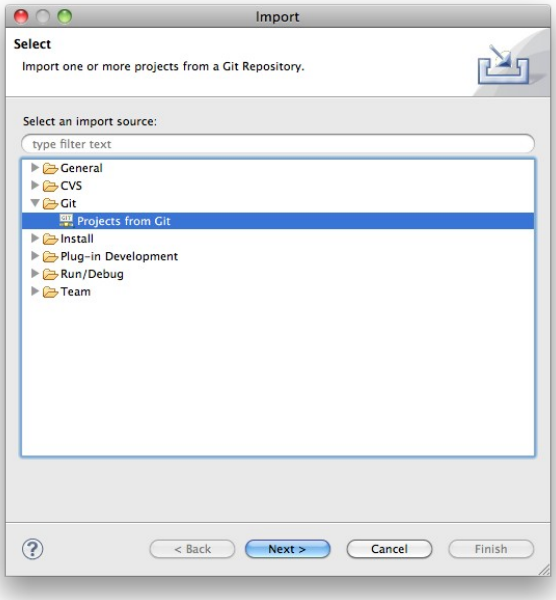


Abb. 13: Auswahl der Quelle

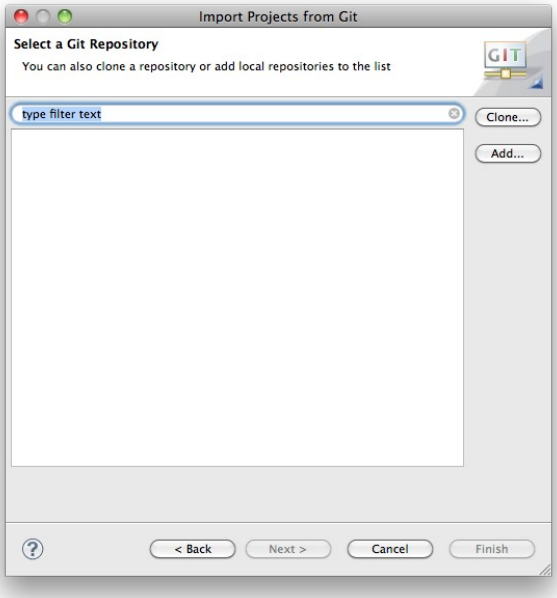


Abb. 14: Clone-Adresse auswählen

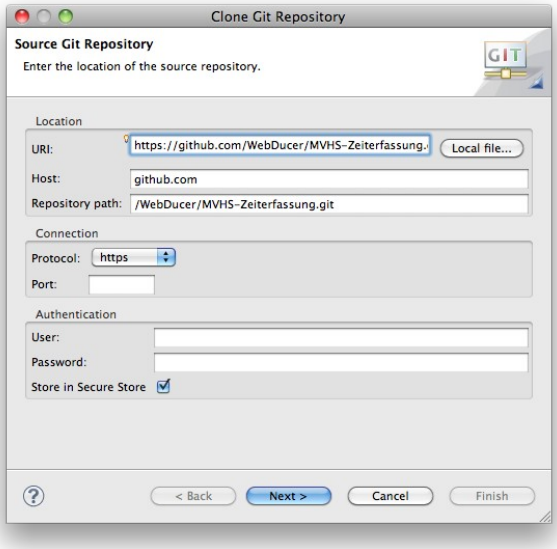


Abb. 15: Adresse der Git-Repository

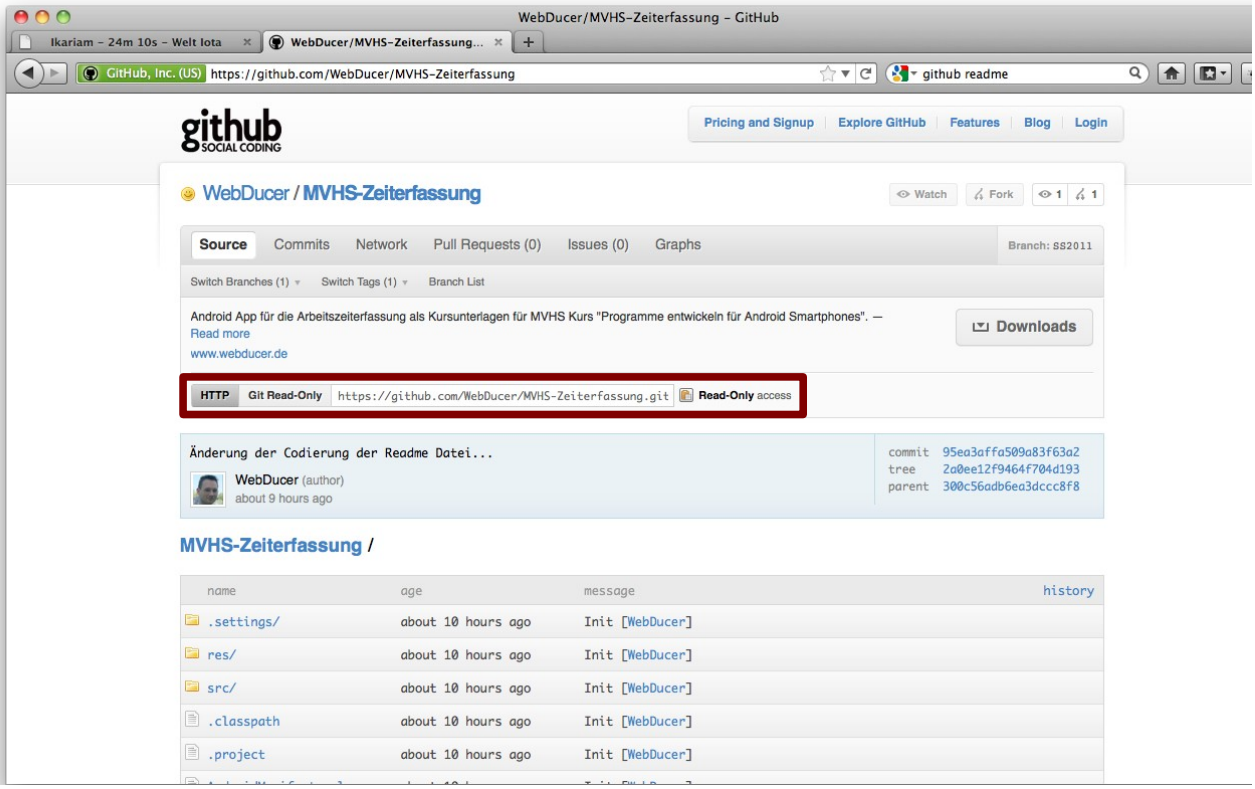


Abb. 16: GitHub - Repository-Adresse

Im nächsten Dialogfenster (Abb. 12) können Sie aus den verfügbaren Branches (Abzweigungen des Quellcodes) da zu importierende auswählen. Klicken Sie danach wieder auf „Next“.

Das folgende Dialog (Abb. 13) bietet die Möglichkeit den Speicherort des Projektes auszuwählen. Weiterhin ist hier auch möglich einzustellen, welcher Branch nach dem Import als das Aktuelle markiert wird (wenn mehrere Branches importiert werden) und wie das Quell-Repository bezeichnet werden soll. Klicken Sie, nach dem Sie alle Angaben Ihre Wünschen angepasst haben, auf „Next“.

Im nächsten Dialog haben Sie die Möglichkeit, falls erwünscht, eine Verknüpfung mit dem Code-Review Service „Gerrit“ herzustellen. Klicken Sie hier anschließend auf „Finish“.

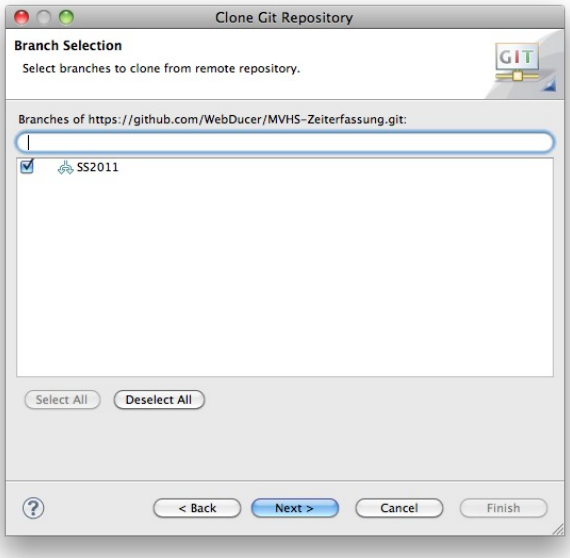


Abb. 17: Auswahl der Abzweigungen

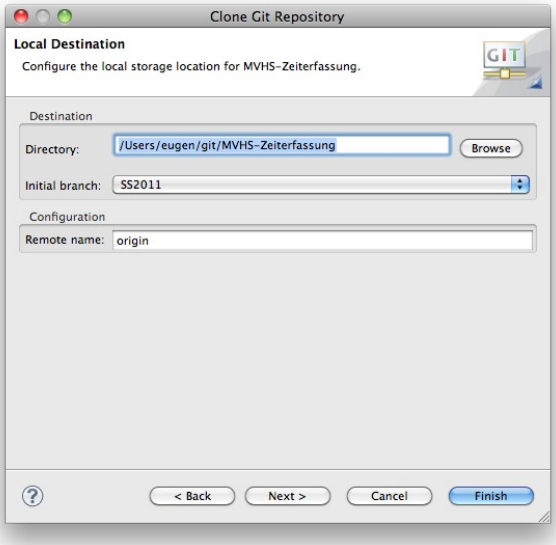


Abb. 18: Speicherort des Projektes

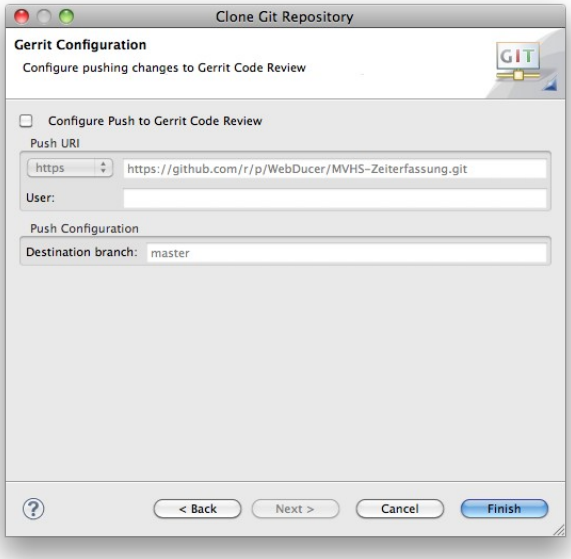


Abb. 20: Gerrit Codereview

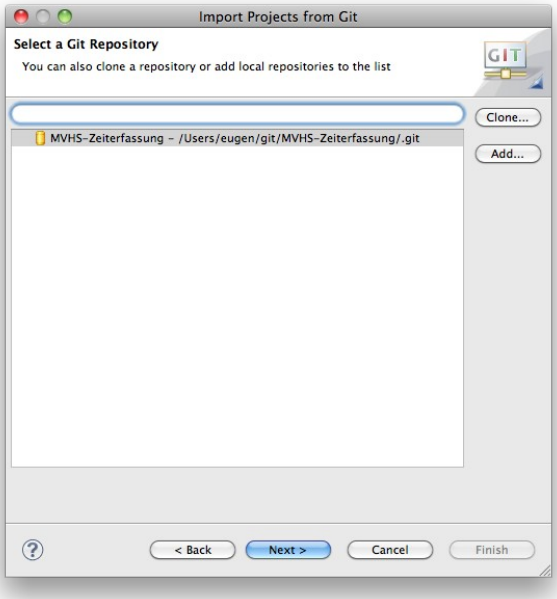


Abb. 19: Repostory auswählen

In der Liste erscheint nur das gerade angelegte Git-Repository (Abb. 15). Wählen Sie dieses aus und klicken Sie auf „Next“.

Im nächsten Dialog (Abb. 16) sehen Sie das Arbeitsverzeichnis, das aus dem Git-Repository importiert wird. Klicken Sie hier wieder auf „Next“.

Im letzten Schritt können Sie auswählen, welche Projekte aus dem Arbeitsverzeichnis importiert werden sollten (Abb. 17). Klicken Sie auf „Finish“.

Das Projekt steht nun in Worspace zur Verfügung.

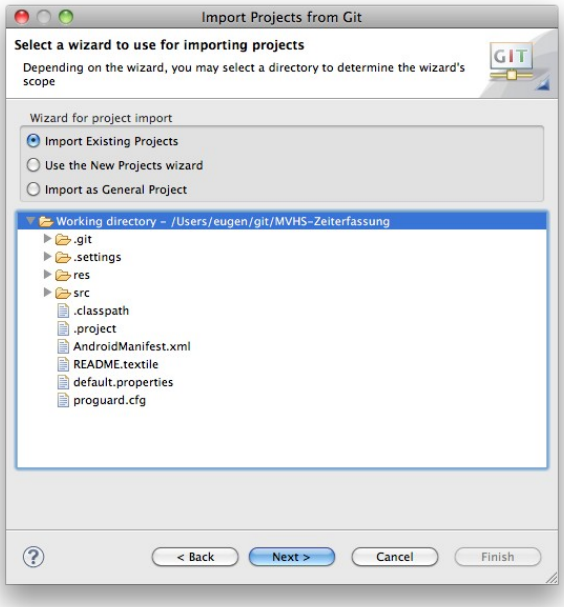


Abb. 21: Importverhalten

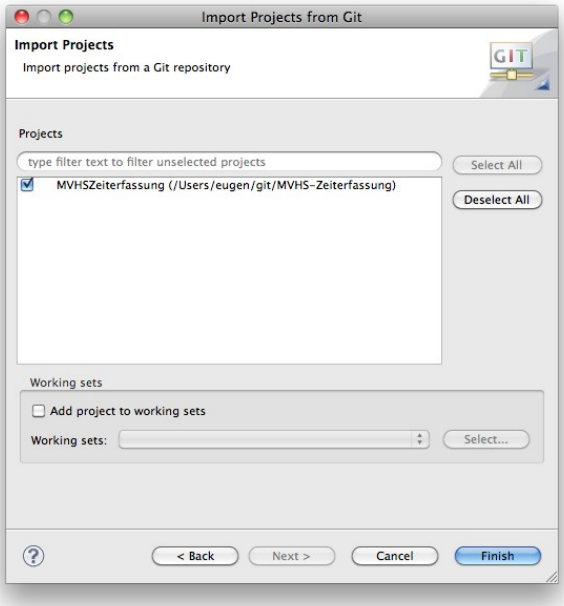


Abb. 22: Projekte auswählen