

Projektbeschreibung ourglass

Projektmitglieder

| Name | Matrikelnummer |
|------------------|----------------|
| Timo Czotscher | 2915944 |
| Andreas Stautner | 3149980 |



Funktion der App

ourglass ist eine App zur Erfassung von Arbeitszeiten. Die Zeiten lassen sich Projekten zuordnen und nach Tätigkeiten unterteilen. Sie bietet eine Übersicht über die gearbeiteten Stunden und die Tätigkeiten des Tages. Zusätzlich lässt sich zu jedem Projekt die Arbeitszeit der Mitarbeiter, die am Projekt beteiligt sind, anzeigen.

Projektstruktur und verwendete Androidkomponenten

Die vier Grundfunktionen der App sind durch einen zugehörigen Eintrag im Navigation Drawer erreichbar. Die Grundkomponente für jeden Unterpunkt ist eine Activity, welche die Funktionalität der Toolbar und des Navigation Drawers von einer Basis-Activity erbt. Die Stundenübersicht, Zeitkorrektur und Projektübersicht nutzen jeweils eine RecyclerView zum Darstellen der Einträge. Diese werden aus einer Firebase Echtzeitdatenbank mithilfe eines Firebase Adapters geladen.

Die Zeitkorrektur verwendet zudem einen Date Picker Dialog zum Auswählen des zu korrigierenden Tages. Die Elemente der Zeitkorrektur nutzen zum Setzen der Start- und Endzeit des jeweiligen Eintrags einen Time Picker Dialog. Die geänderten Zeiten werden sofort mit der Datenbank synchronisiert.

Die Projektübersicht implementiert einen Spinner, welcher seine Einträge aus der Datenbank bezieht, zum Auswählen des Projekts. Entsprechend des ausgewählten

Projekts zeigt die Recycler View alle mitwirkenden Mitarbeiter und ihre für das Projekt aufgewandten Arbeitsstunden an. Durch Tippen auf einen Eintrag erscheint ein Dialog, der die Tätigkeiten des Mitarbeiters für das Projekt mittels eines Kuchendiagramms der externen Bibliothek „hellocharts“ darstellt.

In der Stundenübersicht werden die Arbeitszeiten in einem Zeitraum angezeigt, welcher mittels zweier Date Picker eingegrenzt werden kann. Durch Tippen auf einen Eintrag wird eine neue Activity gestartet, die ein Fragment mit einem Kuchendiagramm zur Darstellung der Tätigkeiten des Tages enthält. Durch Auswählen einer Tätigkeit aus dem Kuchendiagramm erscheint ein Dialog, in welchem die zugehörigen Einträge mit Projekt, Notiz und Zeit einsehbar sind und bearbeitet werden können. Bei Klick auf einen Floating Action Button lassen sich in einem weiteren Fragment neue Einträge zum Kuchendiagramm hinzufügen. Dort lassen sich mittels Spinner die für jeden User in der Datenbank gespeicherten Tätigkeiten auswählen und neue Tätigkeiten hinzufügen.

In der Startseite werden die Ist- und Sollstunden des Monats sowie die Arbeitszeit des aktuellen Tages angezeigt. Für das visuelle Feedback wird beim Starten der Zeit durch Tippen auf das zentral eingebundeneourglass-Symbol eine Animation der Sanduhr gestartet.

Der Login wurde mithilfe von „FirebaseUI“ realisiert. Um zusätzlich zum Namen die wöchentliche Arbeitszeit neuer Nutzer abzufragen, werden diese zu einer weiteren Activity geleitet.

Per NFC-Tag lässt sich die Zeiterfassung, auch wenn die App nicht geöffnet ist, jederzeit bequem starten und stoppen.

Externe Abhängigkeiten

Für den Login der einzelnen User wird der „Firebase Authentication Service“ genutzt. Hierfür haben wir uns entschieden, da es eine einfache und sichere Möglichkeit ist, User zu verwalten. Für uns besteht ein weiterer Vorteil darin, dass der komplizierte Umgang mit sicherheitskritischen Daten (Hashen und Salten der Passwörter) ausgelagert wird. Zudem halten wir uns damit die Möglichkeit offen, später weitere AnmelDEMöglichkeiten ohne großen Aufwand hinzuzufügen.

Die Datenhaltung wird komplett mit der „Firebase Realtime Database“ realisiert. Diese übernimmt die Synchronisation der Nutzerdaten, sodass die Benutzung der App auf mehreren Geräten mit demselben Konto problemlos möglich ist. Ein großer Vorteil ist auch die gute Bibliothek zur Benutzung der „Firebase Realtime Database“, welche beispielsweise Firebase Adapter für die RecyclerViews und Multipath Updates umfasst. Durch die Nutzung eines externen Services werden sämtliche Probleme der Datenhaltung wie entsprechende Infrastruktur ausgelagert, sodass die Entwicklung der App im Fokus stehen kann.

Für eine anschauliche Darstellung wurde die externe Bibliothek „HelloCharts for Android“ verwendet. Diese stellt eine einfache Möglichkeit dar, das geplante Kuchendiagramm umzusetzen und bietet bereits Funktionen, um auf das Auswählen von Einträgen zu reagieren.

Soll-Ist-Vergleich

| Version | Feature | Realisiert | Kommentar |
|---------|------------------------|------------|---|
| 0.1 | Grundfunktion der App | ✓ | |
| | Stempelbutton | ✓ | |
| | Zeitanzeige | ✓ | |
| 0.2 | Visuelle Statusanzeige | ✓ | Animation des Logos während die Zeit läuft |
| | Stundensaldo | ✓ | |
| | Stundenübersicht | ✓ | |
| 0.2.1 | Push-Benachrichtigung | X | Nicht nötig, da die App beim Stempeln mit NFC automatisch geöffnet wird |
| | Haptisches Feedback | ✓ | |
| 1.0 | Manuelles Nachtragen | ✓ | |
| | Fehltage | ✓ | |
| 2.0 | Aktivitäten | ✓ | |
| 2.1 | NFC-Stempeln | ✓ | |
| 3.0 | Projektübersicht | ✓ | |

Nicht mehr geplant, lediglich Ausblick:

| | | |
|-----|------------------|---|
| 3.1 | Teamübersicht | X |
| 4.0 | Stundenhinweis | X |
| | Dauerhafte | X |
| | Benachrichtigung | |
| 4.1 | Widget | X |

Aufwand

Aufgrund unserer fehlenden Erfahrung waren unsere Aufwandsschätzungen der Features weit entfernt von der Realität. Das Projekt hat ca. die zweieinhalb- bis dreifache Zeit der ursprünglichen Schätzung beansprucht. Besonders die „Grundfunktion der App“ hätte in mehrere kleinere Unterpunkte unterteilt werden sollen. Auch die Möglichkeit, die Arbeitszeit in verschiedene Aktivitäten zu unterteilen, hat im Vergleich zu unserer Schätzung besonders lange gedauert.

Fazit

Wir hatten großen Spaß mit dem Entwickeln der App. Da die App-Entwicklung völlig neu für uns war und durch das Nutzen verschiedener Dienste wie Firebase konnten wir viel Erfahrung sammeln. Besonders hilfreich waren die Einblicke in die Versionsverwaltung, die Arbeit im Team mit Git hat sich als sehr praktisch erwiesen. Da wir uns bei jedem Feature erst einlesen mussten, war das Projekt allerdings sehr zeitintensiv. Rückblickend hätten wir die Features in kleinere Pakete unterteilen sollen, um flexibler bei der Entwicklung zu sein.