

**Dokumentation
EMI-App**

SuperTag

Mihail Fomin

Dominic Sehring

Tom Graßmann

Jan Zimmermann

Albert Koch

Inhaltsverzeichnis

| | |
|--|----------|
| TEAMAUFTEILUNG | 3 |
| PFLICHTENHEFT | 3 |
| EXTRA FEATURES | 3 |
| | |
| SUPERTAG | 3 |
| ERSTE MOCKUPS | 4 |
| | |
| PROTOKOLL | 5 |
| 11.11.2016 | 5 |
| 18.11.2016 | 5 |
| 09.12.2016 | 5 |
| 19.12.2016 | 5 |
| 13.01.2017 | 5 |
| FINALE MOCKUPS | 5 |
| | |
| ZEITPLANUNG | 6 |
| SOLL-ZEITPLAN | 6 |
| IST-ZEITPLAN | 6 |
| | |
| APP-FUNKTIONALITÄT | 7 |
| MAIN ACTIVITY: | 7 |
| HINZUFÜGEN/BEARBEITEN ACTIVITY: | 7 |
| SHARE ACTIVITY: | 7 |
| | |
| KLASSENENTWÜRFE | 8 |

Teamaufteilung

- Projektleitung: Tom
- Mockup: Mihail, Jan, Albert
- Protokollierung: Dominic
- Koordination: Tom
- Dokumentation: Jan/Albert

Pflichtenheft

- Die App erlaubt das Erstellen, Anzeigen, Bearbeiten und Löschen von Tagebucheinträgen in Textform.
- Die App erlaubt das Erstellen, Abspielen und Löschen von Sprach- und Fotoformaten.
- Die angelegten Einträge sollen persistent abgelegt werden, in Form von XML-, Audio und Foto Dateien in Form einer Datenbank.
- Möglichkeit einzelne Inhalte zu teilen, durch implementieren einer Share Funktion
- Die App soll unter Android 4 und aufwärts kompatibel sein.
- App soll intuitiv nutzbar sein.

Extra Features

- Zeitstempel der letzten Bearbeitung

SuperTag

The only Tagebuch app you'll need

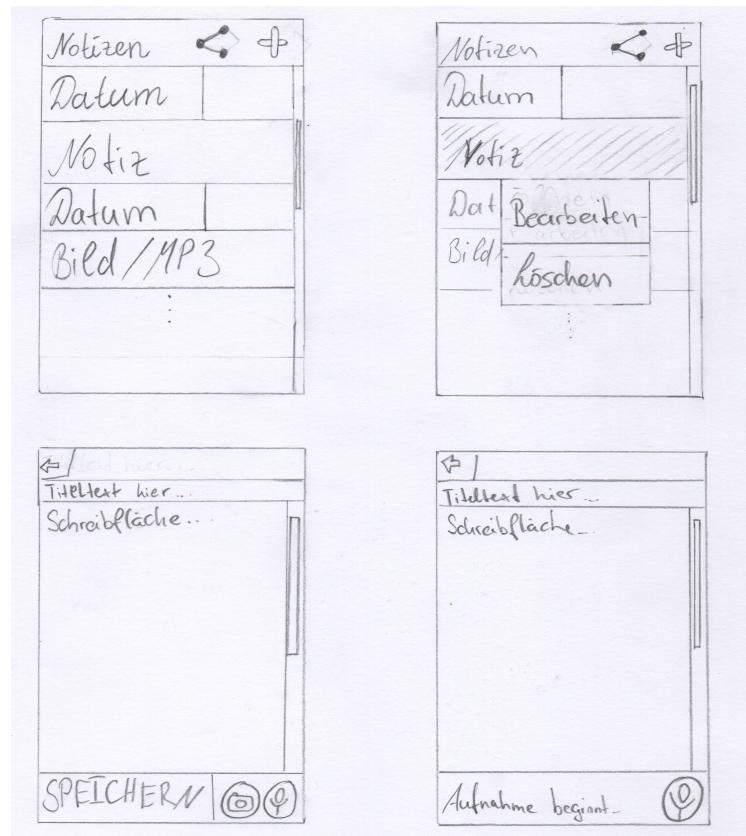
Im Rahmen des Projektes „AUDIO MEMO APP“ sollte eine Android-App mit Möglichkeiten zur persistenten Sprach- und Textaufnahme sowie Speicherung dieser entwickelt werden. Ebenso soll die App die gespeicherten Daten per Android „Share-Funktion“ weitergeben können und bereits ab Android 4 lauffähig sein. Durch die von uns gewählte Erweiterung der Anforderungen einer Notiz-App auf jene einer Tagebuch-App erhöhen sich die benötigten Features um einen Zeitstempel pro Notiz und das Feature mehrere Texte, Videos, Sprachnotizen und Bilder in einem Eintrag abspeichern zu können. Die App soll intuitiv nutzbar und die Einträge editierbar sein.

Die Ideen für solch eine App waren recht schnell gefunden. Wir waren uns alle sehr schnell einig, was diese App später alles können sollte. Wir hatten uns sogar vorgenommen, zusätzliche Funktionen in die App zu programmieren, denn an Einfallsreichtum hat es uns nicht gemangelt. Doch leider und wie zu erwarten, war die Umsetzung unserer Ideen ziemlich schwer. Wie am Anfang jeder Gruppenarbeit, haben wir uns zu Beginn noch recht lange zurückgelehnt und jeder hat sich mit anderen Sachen beschäftigt. Was keine Verwunderung ist, denn die meisten in unserer Gruppe sind Erstsemester Studenten und diese müssen erst einmal in das neue Studentenleben hineinfinden. Nichtsdestotrotz konnten wir eine gerechte Aufgabenverteilung arrangieren, welche jeden Beteiligten zufrieden stellen konnte. So haben sich beispielsweise die Mitglieder der Gruppe, welche in Sachen App Programmierung noch recht Grün

hinter den Ohren sind, meist um Tätigkeiten gekümmert, welche erst einmal weniger mit Programmierung zu tun hatten. Diese waren aber definitiv nicht weniger wichtige Aufgaben. Das betrifft zum Beispiel die Erstellung der MockUps oder das Entwerfen der einzelnen Zwischenpräsentationen und nicht zu vergessen, das Testen der erstellten App. Das soll aber nicht heißen, dass nur die „Profis“ in unserer Gruppe am Programmieren waren. Wenn die Zeit gepasst hat, haben wir uns in der Fakultät zusammengesetzt und gemeinsam an unserer App geschrieben. Ganz reibungslos verlief unsere Arbeit jedoch nicht. Teilweise haben wir wirklich spät angefangen, uns mit den von uns erteilten Aufgaben zu beschäftigen. Andererseits mussten wir auch recht lange im Internet suchen, um herauszufinden wie wir einige Funktionen in unsere App implementieren können. Zudem wurde bereits angesprochen, dass jeder Student mit seinen Eigenen Aufgaben beschäftigt war, was dazu führte, dass wir mit unserem Zeitplan oft zurückhangen. Es kam auch vor, dass wir uns oft nicht alle in der Gruppe treffen konnten, so wie wir es ursprünglich geplant hatten. Deshalb vielen leider einige Aufgaben teilweise einzelnen Personen zu, was heißt, dass wir unsere arrangierte Aufgabenverteilung nicht zu 100% einhalten konnten. Aber glücklicherweise, bestand unserer Gruppe aus kompetenten und netten Leuten, die wussten wie sie mit Problemen zurechtzukommen hatten und keinem wurde vorgeworfen sich nicht am Projekt beteiligt zu haben. Und das ist doch mit einer der wichtigsten Eigenschaften in einer Gruppenarbeit - eine gesunde Atmosphäre im Team.

Zum Schluss kann man sagen, dass alle Mindestanforderungen erfolgreich erfüllt wurden. Trotz der sich als schwieriger herausstellenden Zeit- und Aufgabenplanung konnten selbst komplexe Features realisiert werden. Jedoch sollte an der App weitergearbeitet werden um weitere Funktionen wie ein Customizable User Interface, eine Möglichkeit zur Verschlüsselung der Datenbank oder mehr Appsicherheit durch einen Pin zu ermöglichen. Ebenso ist die App-Performance und die Team Synergie ausbaufähig. Im Großen und Ganzen ist jedoch die Projektarbeit höchst erfolgreich ausgefallen und eine gut nutzbare App mit viel Potenzial entstanden.

Erste Mockups



Protokoll

11.11.2016

- Aufgabenverteilung
- Pflichtenheft
- MockUp
- Grober Zeitplan erstellt

18.11.2016

- Brainstorming
- Pflichtenheft fertig gestellt
 - Tagebuch anstatt Memo App
 - Funktionen der App
 - Bonusfeatures
- Oberfläche entworfen
- Datenbank entworfen
- Kompletter Zeitplan inklusive Aufgabenverteilung im Zeitplan

09.12.2016

- Implementation der Klassen
- Erstellung der Datenbank

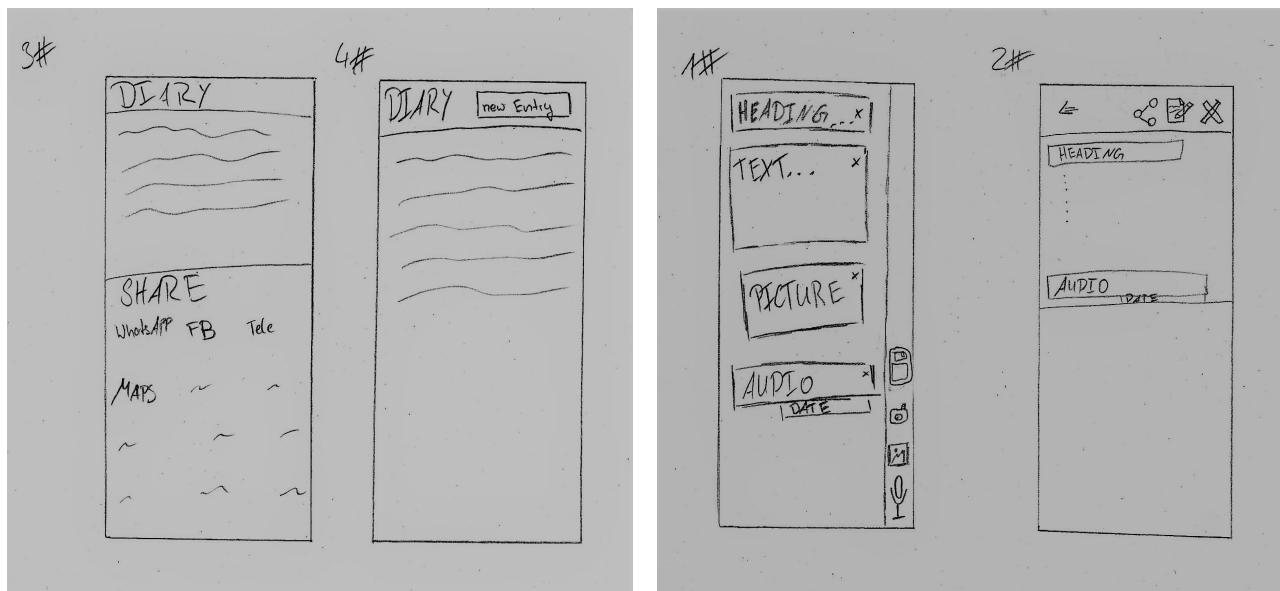
19.12.2016

- Umsetzung des Oberflächenentwurfs der zwei Activities
- Datenbank und Datenstruktur zusammengeführt und eingebunden

13.01.2017

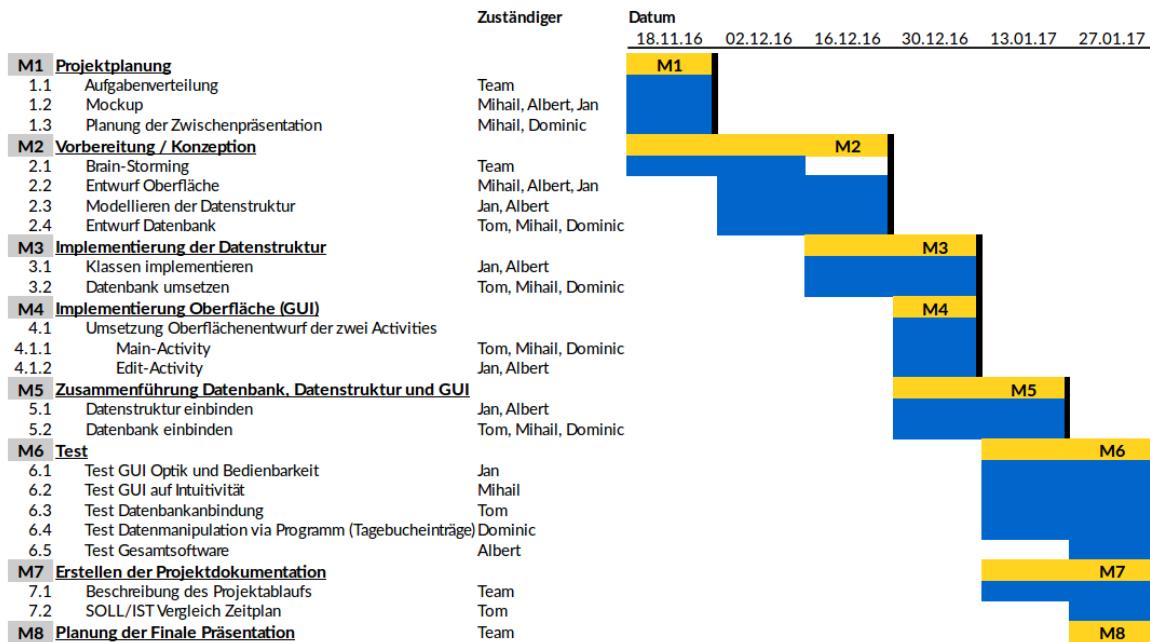
- Update – Stand der Dinge
- Testlauf der App

Finale Mockups

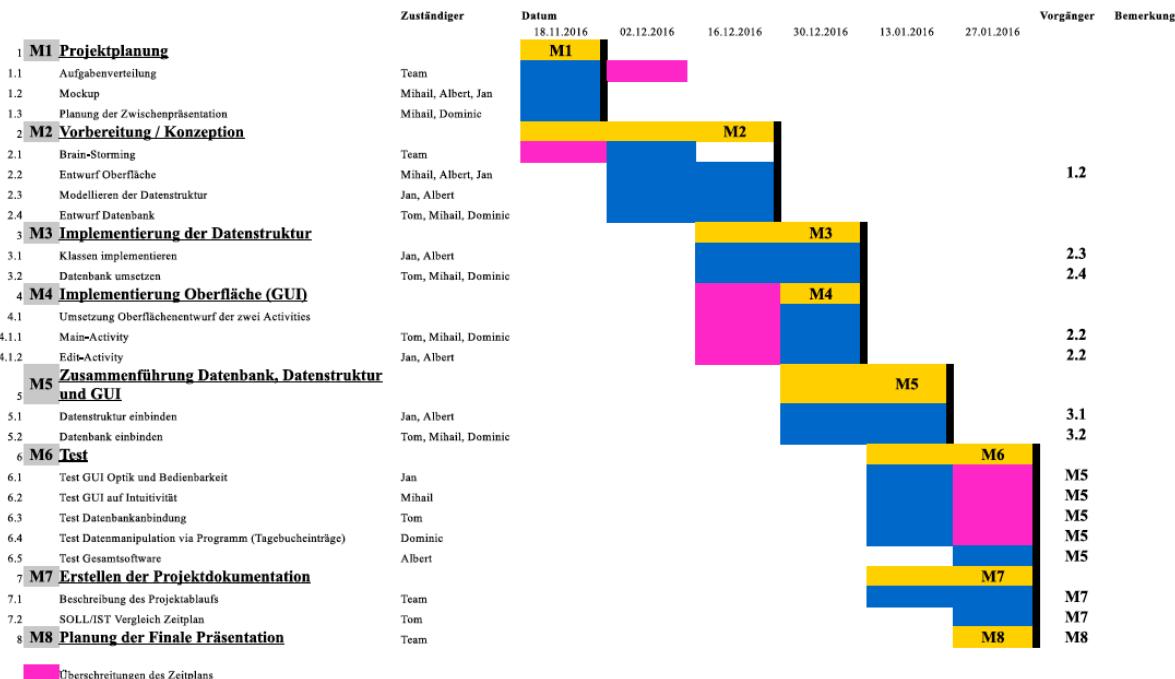


Zeitplanung

SOLL-Zeitplan



IST-Zeitplan



App-Funktionalität

Main Activity:

- Auflistung aller Tagebucheinträge
 - Click auf die drei Punkte oben rechts öffnet Menu zum Erstellen eines neuen Eintrages
- Mit Klick auf den entsprechenden Menüpunkt öffnet sich die **Hinzufügen/Bearbeiten Activity**
- gedrückthalten eines Eintrages, führt zum Öffnen des Optionsmenüs
 - drei Buttons (Share, Bearbeiten)
 - Bearbeiten startet die **Hinzufügen/Bearbeiten Activity**
 - Share startet die **Share Activity**
 - Click auf drei Punkte oben rechts öffnet Menu
 - Möglichkeit zum Löschen oder zum Vorlesen per Sprachsynthese
 - Schütteln des Handys löst hierbei ebenfalls den löschen Dialog aus

Hinzufügen/Bearbeiten Activity:

- Leerer Tagebucheintrag falls neuer Eintrag, sonst werden Text, Bild und Audio des Eintrages hier angezeigt
- fünf Buttons auf der rechten Seite des Bildschirmes (**Vorlesen, Speichern, Kamera, Galerie, Audioaufnahme**)
 - **Vorlesen**: Liest Titel und Textinhalt vor
 - **Speichern**: Sichern der Daten in Datenbank, zurückkehren in die **Main-Activity**
 - → **Kamera**: Starten der Kamera, Möglichkeit eigene Bilder aufzunehmen
 - **Gallerie**: Auswahl Gerät-interner Apps zum Hinzufügen von Bildern aus dem Handyspeicher
 - **Audioaufnahme**: Aufnehmen von Audio solange Button gedrückt gehalten wird, Loslassen führt zur Erstellung von Audioplayer
- Kleine „x“ Buttons führen zur jeweiligen Löschung von Text, Bild, oder Audio

Share Activity:

- Gedrückt halten von Text, Bild oder Audio öffnet ein Optionsmenu
 - Click auf den Share Button oben rechts im Menu löst die jeweilige Share-Action aus

Klassenentwürfe

| <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #f2e0aa; padding: 2px;">PlayButton</th></tr> </thead> <tbody> <tr> <td style="padding: 2px;"> attributes <code>+state : PlayState = PlayState.STOPPED</code> operations <code>«constructor»+PlayButton(context : Context)</code> <code>«constructor»+PlayButton(context : Context, attrs : AttributeSet)</code> <code>«constructor»+PlayButton(context : Context, attrs : AttributeSet, defStyle : int)</code> <code>«setter»+setPlaying() : void</code> <code>«setter»+setPaused() : void</code> <code>«setter»+setStopped() : void</code> </td></tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #f2e0aa; padding: 2px;">EntryAdapter</th></tr> </thead> <tbody> <tr> <td style="padding: 2px;"> attributes <code>-EDIT_ENTRY : int = 1{readOnly}</code> <code>-context : Context</code> <code>-noteList : ArrayList</code> <code>-tableManager : TableManager</code> <code>-selectedTextContent : View = null</code> operations <code>«constructor»+EntryAdapter(context : Context, noteList : ArrayList, tableManager : TableManager)</code> <code>«javaElements»+onEnter(ContentValues values) : void{javaAnnotations = "@Override"}</code> <code>«javaElements»+getEntry(int i : int) : Object{javaAnnotations = "@Override"}</code> <code>«javaElements»+getEntry(int i : int) : long{javaAnnotations = "@Override"}</code> <code>«setter»+setActionBar(mode : ActionBarMode) : void</code> <code>«javaElements»+getView(pos : int, view : View, viewGroup : ViewGroup) : View{javaAnnotations = "@Override"}</code> </td></tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #f2e0aa; padding: 2px;">TableManager</th></tr> </thead> <tbody> <tr> <td style="padding: 2px;"> attributes <code>-table : ListView</code> <code>-context : Context</code> <code>-arrayList : ArrayList</code> <code>-adapter : EntryAdapter</code> <code>-database : Database</code> operations <code>«constructor»+TableManager(context : Context, table : ListView, database : Database)</code> <code>+addEntry(note : Note) : void</code> <code>+removeEntry(note : Note) : void</code> <code>+updateEntry(note : Note) : boolean</code> <code>+notifyAdapter() : void</code> </td></tr> </tbody> </table> | PlayButton | attributes <code>+state : PlayState = PlayState.STOPPED</code> operations <code>«constructor»+PlayButton(context : Context)</code> <code>«constructor»+PlayButton(context : Context, attrs : AttributeSet)</code> <code>«constructor»+PlayButton(context : Context, attrs : AttributeSet, defStyle : int)</code> <code>«setter»+setPlaying() : void</code> <code>«setter»+setPaused() : void</code> <code>«setter»+setStopped() : void</code> | EntryAdapter | attributes <code>-EDIT_ENTRY : int = 1{readOnly}</code> <code>-context : Context</code> <code>-noteList : ArrayList</code> <code>-tableManager : TableManager</code> <code>-selectedTextContent : View = null</code> operations <code>«constructor»+EntryAdapter(context : Context, noteList : ArrayList, tableManager : TableManager)</code> <code>«javaElements»+onEnter(ContentValues values) : void{javaAnnotations = "@Override"}</code> <code>«javaElements»+getEntry(int i : int) : Object{javaAnnotations = "@Override"}</code> <code>«javaElements»+getEntry(int i : int) : long{javaAnnotations = "@Override"}</code> <code>«setter»+setActionBar(mode : ActionBarMode) : void</code> <code>«javaElements»+getView(pos : int, view : View, viewGroup : ViewGroup) : View{javaAnnotations = "@Override"}</code> | TableManager | attributes <code>-table : ListView</code> <code>-context : Context</code> <code>-arrayList : ArrayList</code> <code>-adapter : EntryAdapter</code> <code>-database : Database</code> operations <code>«constructor»+TableManager(context : Context, table : ListView, database : Database)</code> <code>+addEntry(note : Note) : void</code> <code>+removeEntry(note : Note) : void</code> <code>+updateEntry(note : Note) : boolean</code> <code>+notifyAdapter() : void</code> | <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #f2e0aa; padding: 2px;">LocalisationThread</th></tr> </thead> <tbody> <tr> <td style="padding: 2px;"> attributes <code>-TAG : String = "LocalisationThread"{readOnly}</code> <code>-DELAY : int = 1000{readOnly}</code> <code>-LOCATION_FOUND : int = 20{readOnly}</code> <code>-LOCATION_ERROR : int = 21{readOnly}</code> <code>-currentActivity</code> <code>-locHandler : Handler</code> <code>-locationManager : LocationManager</code> <code>-locationListener : LocationListener</code> <code>-tryCounter : int = 0</code> operations <code>«constructor»+LocalisationThread(currentActivity : , locHandler : Handler)</code> <code>«javaElements»+run() : void{javaAnnotations = "@Override"}</code> <code>«getter»-getCityfromLocation(location : Location) : String</code> </td></tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #f2e0aa; padding: 2px;">Database</th></tr> </thead> <tbody> <tr> <td style="padding: 2px;"> attributes <code>-appContext : Context</code> <code>-DATABASE_NAME : String = "entry.db"{readOnly}</code> <code>-TABLE_NAME : String = "entry_table"{readOnly}</code> <code>-ID : String = "ID"{readOnly}</code> <code>-TITLE : String = "TITLE"{readOnly}</code> <code>-DATE : String = "DATE"{readOnly}</code> <code>-CITY : String = "CITY"{readOnly}</code> <code>-CONTEXT_TEXT : String = "TEXT"{readOnly}</code> <code>-CONTEXT_VOICE : String = "VOICE"{readOnly}</code> <code>-CONTEXT_IMAGE : String = "IMAGE"{readOnly}</code> operations <code>«constructor»+Database(context : Context)</code> <code>«javaElements»+onCreate(SQLiteDatabase db : SQLiteDatabase) : void{javaAnnotations = "@Override"}</code> <code>«javaElements»+onUpgrade(SQLiteDatabase db : SQLiteDatabase, i : int, l1 : int) : void{javaAnnotations = "@Override"}</code> <code>+insertData(note : Note) : boolean</code> <code>+updateData(note : Note) : boolean</code> <code>+getters+getAllData() : Cursor</code> <code>+removeAllData() : boolean</code> <code>+removeData(note : Note) : boolean</code> <code>-IDHelper(id : int) : int</code> <code>+getters+getNextFreeID() : int</code> </td></tr> </tbody> </table> | LocalisationThread | attributes <code>-TAG : String = "LocalisationThread"{readOnly}</code> <code>-DELAY : int = 1000{readOnly}</code> <code>-LOCATION_FOUND : int = 20{readOnly}</code> <code>-LOCATION_ERROR : int = 21{readOnly}</code> <code>-currentActivity</code> <code>-locHandler : Handler</code> <code>-locationManager : LocationManager</code> <code>-locationListener : LocationListener</code> <code>-tryCounter : int = 0</code> operations <code>«constructor»+LocalisationThread(currentActivity : , locHandler : Handler)</code> <code>«javaElements»+run() : void{javaAnnotations = "@Override"}</code> <code>«getter»-getCityfromLocation(location : Location) : String</code> | Database | attributes <code>-appContext : Context</code> <code>-DATABASE_NAME : String = "entry.db"{readOnly}</code> <code>-TABLE_NAME : String = "entry_table"{readOnly}</code> <code>-ID : String = "ID"{readOnly}</code> <code>-TITLE : String = "TITLE"{readOnly}</code> <code>-DATE : String = "DATE"{readOnly}</code> <code>-CITY : String = "CITY"{readOnly}</code> <code>-CONTEXT_TEXT : String = "TEXT"{readOnly}</code> <code>-CONTEXT_VOICE : String = "VOICE"{readOnly}</code> <code>-CONTEXT_IMAGE : String = "IMAGE"{readOnly}</code> operations <code>«constructor»+Database(context : Context)</code> <code>«javaElements»+onCreate(SQLiteDatabase db : SQLiteDatabase) : void{javaAnnotations = "@Override"}</code> <code>«javaElements»+onUpgrade(SQLiteDatabase db : SQLiteDatabase, i : int, l1 : int) : void{javaAnnotations = "@Override"}</code> <code>+insertData(note : Note) : boolean</code> <code>+updateData(note : Note) : boolean</code> <code>+getters+getAllData() : Cursor</code> <code>+removeAllData() : boolean</code> <code>+removeData(note : Note) : boolean</code> <code>-IDHelper(id : int) : int</code> <code>+getters+getNextFreeID() : int</code> |
|--|------------|---|--------------|--|--------------|---|--|--------------------|---|----------|--|
| PlayButton | | | | | | | | | | | |
| attributes <code>+state : PlayState = PlayState.STOPPED</code> operations <code>«constructor»+PlayButton(context : Context)</code> <code>«constructor»+PlayButton(context : Context, attrs : AttributeSet)</code> <code>«constructor»+PlayButton(context : Context, attrs : AttributeSet, defStyle : int)</code> <code>«setter»+setPlaying() : void</code> <code>«setter»+setPaused() : void</code> <code>«setter»+setStopped() : void</code> | | | | | | | | | | | |
| EntryAdapter | | | | | | | | | | | |
| attributes <code>-EDIT_ENTRY : int = 1{readOnly}</code> <code>-context : Context</code> <code>-noteList : ArrayList</code> <code>-tableManager : TableManager</code> <code>-selectedTextContent : View = null</code> operations <code>«constructor»+EntryAdapter(context : Context, noteList : ArrayList, tableManager : TableManager)</code> <code>«javaElements»+onEnter(ContentValues values) : void{javaAnnotations = "@Override"}</code> <code>«javaElements»+getEntry(int i : int) : Object{javaAnnotations = "@Override"}</code> <code>«javaElements»+getEntry(int i : int) : long{javaAnnotations = "@Override"}</code> <code>«setter»+setActionBar(mode : ActionBarMode) : void</code> <code>«javaElements»+getView(pos : int, view : View, viewGroup : ViewGroup) : View{javaAnnotations = "@Override"}</code> | | | | | | | | | | | |
| TableManager | | | | | | | | | | | |
| attributes <code>-table : ListView</code> <code>-context : Context</code> <code>-arrayList : ArrayList</code> <code>-adapter : EntryAdapter</code> <code>-database : Database</code> operations <code>«constructor»+TableManager(context : Context, table : ListView, database : Database)</code> <code>+addEntry(note : Note) : void</code> <code>+removeEntry(note : Note) : void</code> <code>+updateEntry(note : Note) : boolean</code> <code>+notifyAdapter() : void</code> | | | | | | | | | | | |
| LocalisationThread | | | | | | | | | | | |
| attributes <code>-TAG : String = "LocalisationThread"{readOnly}</code> <code>-DELAY : int = 1000{readOnly}</code> <code>-LOCATION_FOUND : int = 20{readOnly}</code> <code>-LOCATION_ERROR : int = 21{readOnly}</code> <code>-currentActivity</code> <code>-locHandler : Handler</code> <code>-locationManager : LocationManager</code> <code>-locationListener : LocationListener</code> <code>-tryCounter : int = 0</code> operations <code>«constructor»+LocalisationThread(currentActivity : , locHandler : Handler)</code> <code>«javaElements»+run() : void{javaAnnotations = "@Override"}</code> <code>«getter»-getCityfromLocation(location : Location) : String</code> | | | | | | | | | | | |
| Database | | | | | | | | | | | |
| attributes <code>-appContext : Context</code> <code>-DATABASE_NAME : String = "entry.db"{readOnly}</code> <code>-TABLE_NAME : String = "entry_table"{readOnly}</code> <code>-ID : String = "ID"{readOnly}</code> <code>-TITLE : String = "TITLE"{readOnly}</code> <code>-DATE : String = "DATE"{readOnly}</code> <code>-CITY : String = "CITY"{readOnly}</code> <code>-CONTEXT_TEXT : String = "TEXT"{readOnly}</code> <code>-CONTEXT_VOICE : String = "VOICE"{readOnly}</code> <code>-CONTEXT_IMAGE : String = "IMAGE"{readOnly}</code> operations <code>«constructor»+Database(context : Context)</code> <code>«javaElements»+onCreate(SQLiteDatabase db : SQLiteDatabase) : void{javaAnnotations = "@Override"}</code> <code>«javaElements»+onUpgrade(SQLiteDatabase db : SQLiteDatabase, i : int, l1 : int) : void{javaAnnotations = "@Override"}</code> <code>+insertData(note : Note) : boolean</code> <code>+updateData(note : Note) : boolean</code> <code>+getters+getAllData() : Cursor</code> <code>+removeAllData() : boolean</code> <code>+removeData(note : Note) : boolean</code> <code>-IDHelper(id : int) : int</code> <code>+getters+getNextFreeID() : int</code> | | | | | | | | | | | |
| <h2>Note</h2> <p>attributes</p> <pre>#ID : int #timestamp : long #title : String #city : String #text_note : String = "" #voice_note_path : String = "" #image_note_path : String = "" #bitmap : Bitmap = null #isSelected : boolean = false ...</pre> <p>operations</p> <pre>«constructor»+Note(ID : int, title : String) «getter»+getID() : int «getter»+getTitle() : String «setter»+setTitle(title : String) : void «setter»+setTextNote(text : String) : void «setter»+setVoiceNote(text : String) : void «setter»+setImageNote(image_note_path : String) : void «getter»+getTextNote() : String «getter»+getVoiceNote() : String «getter»+getImageNote() : String +addToDatabase(db : Database) : void >equals(object : Object) : boolean «setter»+setTimestamp(timestamp : long) : void «getter»+getTimestamp() : long «getter»+getDate_Location() : String «getter»+getBitmap() : Bitmap «setter»+setBitmap(bitmap : Bitmap) : void «getter»+getCity() : String «setter»+setCity(city : String) : void «getter»+isSelected() : boolean «setter»+setSelected(selected : boolean) : void ...</pre> | | | | | | | | | | | |