

# **Anforderungsdokumentation für die mobile App SC Lerchenberg e.V. TrainerAssist**

*Mobile Computing-Praktikum  
Johannes Gutenberg-Universität Mainz  
Wintersemester 2016/17*

Brand  
Niedballa  
Porth  
Reinhart  
Schulz  
Skronn

## **I. Ziel und Zweck der App**

## **II. Technische Anforderungen**

### **a) Systemverhalten**

- 1) Einloggen
- 2) nach Übung suchen
  - i) Filtern nach Altersklasse
  - ii) Filtern nach Gruppengröße
  - iii) Filtern nach Schwerpunkt und Schwierigkeitsgrad
  - iv) Filtern nach Schlagwörtern
- 3) Übung/Trainingseinheit herunterladen
- 4) Übung/Trainingseinheit erstellen
  - i) Name der Übung/Trainingseinheit
  - ii) Beschreibung der Übung/Trainingseinheit
  - iii) (Gesamt-)Dauer der Übung
  - iv) Gruppengröße der Übung
  - v) Altersklasse der Übung
  - vi) Materialbedarf der Übung/Trainingseinheit
  - vii) Bearbeiten einer Graphik
  - viii) Schwerpunkt und Schwierigkeitsgrad
  - ix) Schlagwörter
  - x) Videolink
  - xi) Übungen zu Trainingseinheit hinzufügen
- 5) Übung/Trainingseinheit speichern
- 6) Übung/Trainingseinheit hochladen
- 7) Datenbank aktualisieren
- 8) Einstellungen ändern
- 9) Betätigen des Videolinks
- 10) Daten exportieren

### **b) Systemstruktur**

- 1) Ladezeiten beim Seitenwechsel
- 2) Zuverlässigkeit
- 3) Benutzerfreundlichkeit
- 4) Datenschutz und -sicherheit

## **III. Nutzungsszenarien**

- a) Einloggen
- b) Übung herunterladen
- c) Übung erstellen
- d) Übung hochladen
- e) Benutzernamen ändern

## **IV. Aufbau und Komponenten**

3 UMLs

## **V. Glossar**

- a) Altersklassen nach DFB-Standard
- b) Schlagwörter-Katalog

## **I) Ziel und Zweck der App**

Die mobile App "SC Lerchenberg e.V. TrainerAssist" wurde im Rahmen des Mobile Computing-Praktikums an der Johannes Gutenberg-Universität Mainz im Wintersemester 2016/17 für den SC Lerchenberg e.V. entworfen und dient der Unterstützung (ehrenamtlicher) Trainer bei der Erstellung und Organisation von Übungen und Trainingseinheiten, sowie der (terminlichen) Verwaltung verschiedener Trainingsgruppen und soll somit auch von Laien ohne weitere Einweisung benutzbar sein.

## **II) Technische Anforderungen**

### II.a) Systemverhalten

#### II.a.1) Einloggen

Beim erstmaligen Nutzen der App wird der Benutzer um die Eingabe seines Nutzernamens gebeten. Nach korrekter Eingabe ist die App dann mit der auf dem Server hinterlegten Datenbank verbunden, ohne dass eine erneute Authentifizierung notwendig ist.

#### II.a.2) nach Übung suchen

Der Benutzer kann verschiedene Filter nutzen, um eine für seine Trainingsgruppe passende Übung zu finden.

##### II.a.2.i) Filtern nach Altersklasse

Der Benutzer kann die Suchergebnisse eingrenzen, indem er nach der gewünschten Altersklasse filtert. Die App orientiert sich dabei an den Altersklassen-Einteilungen des DFB. (Näheres im Glossar.)

##### II.a.2.ii) Filtern nach Gruppengröße

Der Benutzer kann die Suchergebnisse eingrenzen, indem er nach der gewünschten Gruppengröße filtert.

##### II.a.2.iii) Filtern nach Schwerpunkt und Schwierigkeitsgrad

Der Benutzer kann die Suchergebnisse eingrenzen, indem er nach dem gewünschten Schwerpunkt filtert und dann nach Schwierigkeitsgrad.

Die App unterscheidet drei Schwerpunkte (Taktik, Technik, Physis), deren Schwierigkeitsgrad im 7-Punkte-System angegeben wird (1 = leicht, 7 = schwer), wobei jeder Schwerpunkt mit mindestens Schwierigkeitsgrad 1 angegeben ist.

##### II.a.2.iv) Filtern nach Schlagwörtern

Der Benutzer kann die Suchergebnisse eingrenzen, indem er nach Schlagwörtern filtert.

Dazu enthält die App einen Schlagwörter-Katalog (siehe Glossar), aus dem der Benutzer wählen kann.

#### II.a.3) Übung/Trainingseinheit herunterladen

Der Benutzer kann Übungen/Trainingseinheiten herunterladen, um diese offline (ohne Internetverbindung) zu nutzen.

#### II.a.4) Übung/Trainingseinheit erstellen

Der Benutzer kann eigene Übungen/Trainingseinheiten erstellen.

Er kann der erstellten Übung/Trainingseinheit einen Namen geben, die Übung/Trainingseinheit beschreiben, eine Dauer eintragen (nur Übung), Gruppengröße und Altersklasse empfehlen (nur Übung), einen Spielplan erstellen, aus dem sich der Materialbedarf berechnet (nur Übung), Schwerpunkt(e) und dessen/deren Schwierigkeitsgrad setzen (nur Übung), eine Liste an Schlagwörtern angeben (nur Übung), einen Videolink angeben (nur Übung) und Übungen zu einer Trainingseinheit hinzufügen.

#### II.a.4.i) Name der Übung/Trainingseinheit

Der Benutzer kann der zu erstellenden Übung/Trainingseinheit einen Namen geben.

Beleidigungen, Telefonnummern oder das Veröffentlichen von privaten Daten sind nicht erlaubt.

#### II.a.4.ii) Beschreibung der Übung/Trainingseinheit

Der Benutzer kann eine kurze Beschreibung seiner erstellten Übung/Trainingseinheit schreiben.

Beleidigungen, Telefonnummern oder das Veröffentlichen von privaten Daten sind nicht erlaubt.

#### II.a.4.iii) (Gesamt-)Dauer der Übung/Trainingseinheit

Der Benutzer kann eine Empfehlung für die Dauer seiner erstellten Übung angeben.

Das Setzen der Dauer beim Erstellen einer Trainingseinheit entfällt, da hier die Gesamtdauer aus den einzelnen Zeiten der in der Trainingseinheit enthaltenen Übungen berechnet wird.

#### II.a.4.iv) Gruppengröße der Übung/Trainingseinheit

Der Benutzer kann angeben, für welche Gruppengröße seine erstellte Übung/Trainingseinheit ausgelegt ist. Dabei ist die voreingestellte Anzahl 1.

Das Setzen der Gruppengröße beim Erstellen einer Trainingseinheit entfällt, da sich hier die Gruppengröße an der größten Mindestanzahl der in der Trainingseinheit enthaltenen Übungen orientiert.

#### II.a.4.v) Altersklasse der Übung/Trainingseinheit

Der Benutzer kann in seiner erstellten Übung/Trainingseinheit eine Altersklassen-Empfehlung angeben.

Dabei ist das voreingestellte Mindestalter "Bambini".

Das Setzen der Altersklasse beim Erstellen einer Trainingseinheit entfällt, da sich hier die Altersklasse am kleinsten Mindestalter der in der Trainingseinheit enthaltenen Übungen orientiert.

#### II.a.4.vi) Materialbedarf der Übung/Trainingseinheit

Beim Bearbeiten der Graphik werden die Materialien automatisch berechnet und angegeben.

Das Setzen des Materialbedarfs beim Erstellen einer Trainingseinheit entfällt, da hier der Materialbedarf aus den in der Trainingseinheit enthaltenen Übungen berechnet wird.

#### II.a.4.vii) Bearbeiten einer Graphik

Der Benutzer kann zur Veranschaulichung seiner beschriebenen Übung einen zur Übung gehörenden, anfangs leeren Spielplan bearbeiten, indem er Materialien und Pfeile auf einem Spielfeld positioniert.

Anzahl der Objekte und Länge der Pfeile können vom Benutzer angepasst werden.

#### II.a.4.viii) Schwerpunkt und Schwierigkeitsgrad

Der Benutzer kann bis zu drei Schwerpunkte auswählen (Taktik, Technik und Physis) und diesen einen Schwierigkeitsgrad (1 = leicht, 7 = schwierig) zuordnen. Dabei ist der voreingestellte Mindestwert 1.

Das Setzen von Schwerpunkten und deren Schwierigkeitsgraden beim Erstellen einer Trainingseinheit entfällt, da sich hier die Schwerpunkte und deren Schwierigkeitsgrade aus den in der Trainingseinheit enthaltenen Übungen berechnen.

#### II.a.4.ix) Schlagwörter

Der Benutzer kann eine Liste von Schlagwörtern angeben, um den anderen Trainern das Finden seiner Übung zu erleichtern.

Das Setzen von Schlagwörtern beim Erstellen einer Trainingseinheit entfällt, da hier die Schlagwörter aus den in der Trainingseinheit enthaltenen Übungen angegeben werden.

Beleidigungen, Telefonnummern oder das Veröffentlichen von privaten Daten sind nicht erlaubt.

#### II.a.4.x) Videolink

Der Benutzer kann zur Verdeutlichung seiner Übung einen Videolink angeben.

Das Angeben eines Videolinks beim Erstellen einer Trainingseinheit entfällt.

#### II.a.4.xi) Übung zu Trainingseinheit hinzufügen

Der Benutzer kann ausgewählte Übungen zu einer Trainingseinheit hinzufügen.

#### II.a.5) Übung/Trainingseinheit speichern

Der Benutzer kann Übungen/Trainingseinheiten auf seinem Mobilgerät speichern, um diese auch offline (ohne Internetverbindung) nutzen zu können.

#### II.a.6) Übung/Trainingseinheit hochladen

Der Benutzer kann seine erstellten Übungen/Trainingseinheiten hochladen und sie somit anderen Trainern zur Verfügung stellen.

Dazu muss er seine Email-Adresse freigeschaltet haben (siehe II.a.8)

#### II.a.7) Datenbank aktualisieren

Der Benutzer kann die auf seinem Mobilgerät gespeicherten Daten mit denen in der Datenbank synchronisieren und aktualisieren.

#### II.a.8) Einstellungen ändern

Der Benutzer kann die Einstellungen der App ändern oder nach seinen Wünschen anpassen.

Er kann den Benutzernamen ändern, seine Email-Adresse freischalten lassen, um erstellte Übungen/Trainingseinheiten in die Datenbank hochladen zu können und die Datenbank zurücksetzen.

#### II.a.9) Betätigen des Videolinks

Der Benutzer wird durch Betätigen des Videolinks bei bestehender Internetverbindung auf die im Link angegebene Internetseite weitergeleitet.

#### II.a.10) Daten in Kalender exportieren

Der Benutzer kann seine Übungen und Trainingseinheiten zur besseren Verwaltung seiner Trainingsgruppen in den Google-Kalender eintragen lassen.

### II.b) Systemstruktur

#### II.b.1) Ladezeiten beim Seitenwechsel

Der Benutzer muss nicht länger als 2 Sekunden auf das Laden einer neuen Seite warten.

#### II.b.2) Zuverlässigkeit

Die App läuft größtenteils problemlos und nicht erlaubte Eingaben werden teilweise abgefangen..

#### II.b.3) Benutzerfreundlichkeit

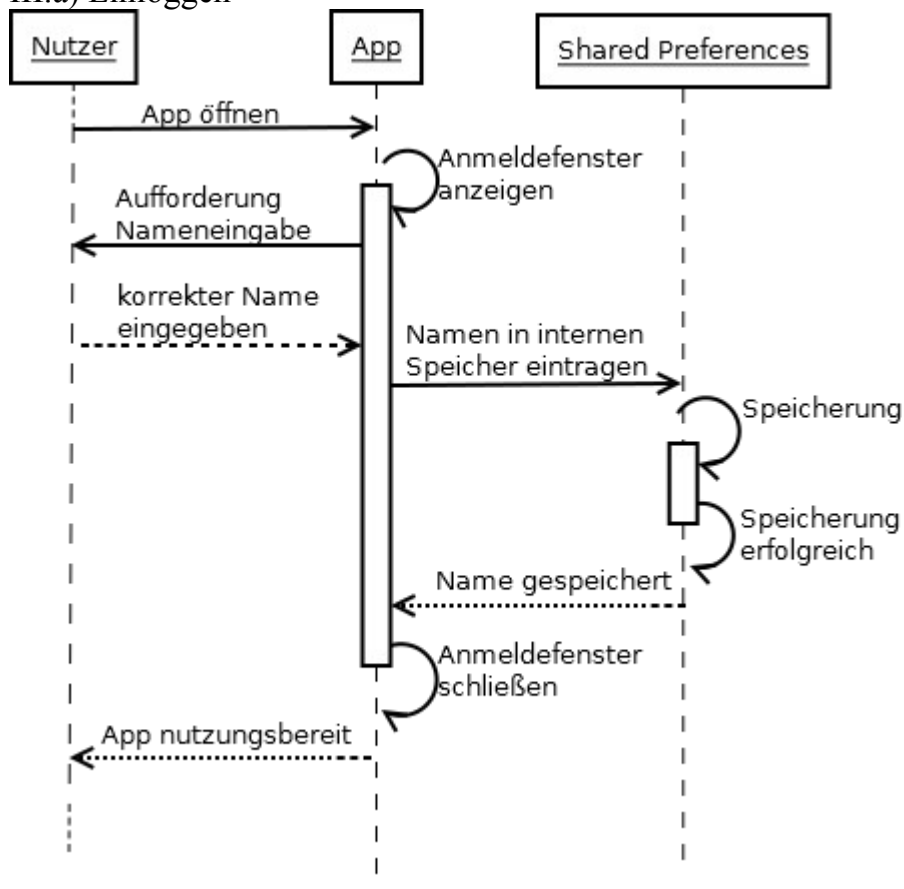
Die App kann auch von Laien ohne weitere Einarbeitung gut genutzt werden.

#### II.b.4) Datenschutz und -sicherheit

Die App gibt keine Daten an Dritte weiter. Name und Email-Adresse werden nicht veröffentlicht, statt dessen werden vom Benutzer angegebene Pseudonyme angezeigt.

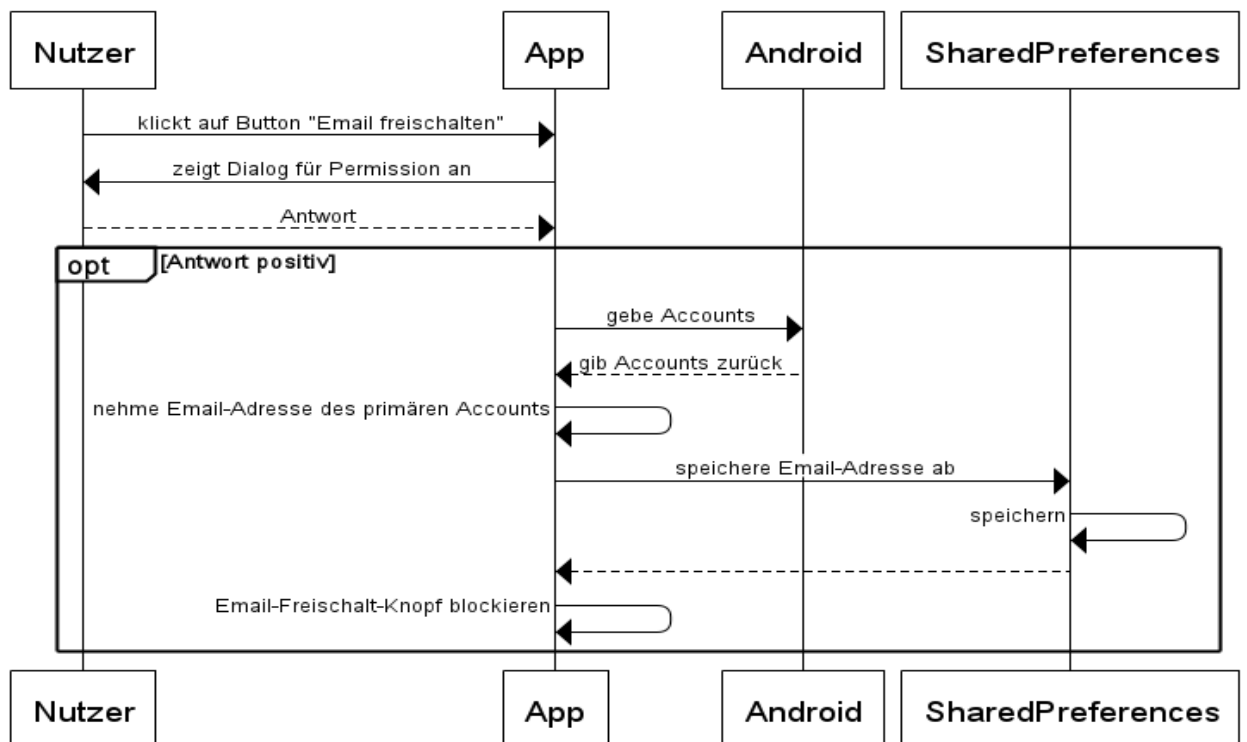
### III. Nutzungsszenarien

#### III.a) Einloggen

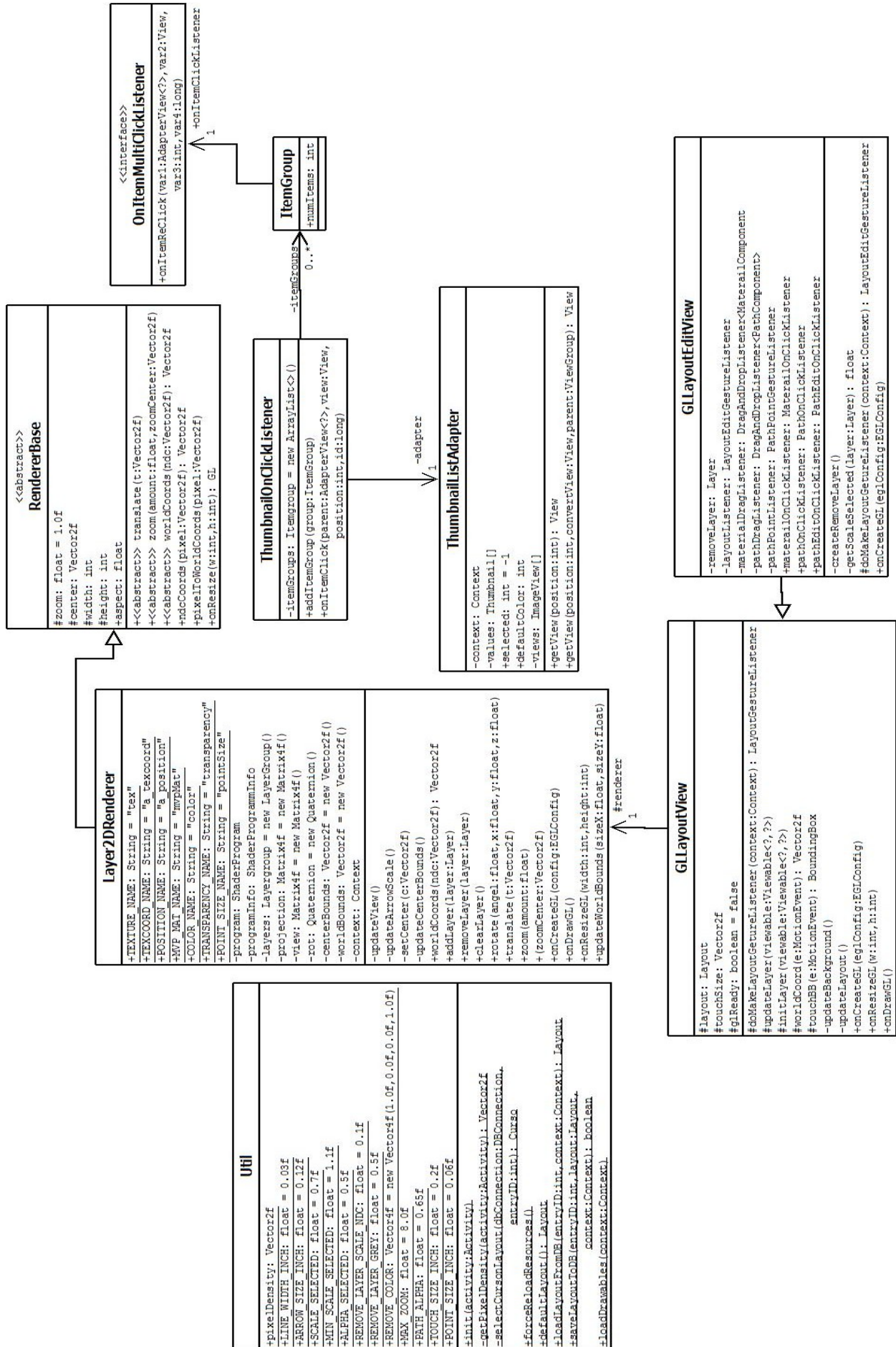


#### III.e) Email freischalten

##### Email-Adresse (für Uploadfunktionalität) freigeben



## IV) Aufbau und Komponenten



Shader
+id: int
-compiled: boolean
+setShaderSource(source:String)
+compile()
+loadFromResource(resid:int, context:Context)
+delete()

← -attachedShaders  
0..\*

ShaderProgram
+id: int
+name: String
-attachedShaders: List<Shader> = new ArrayList<>()
-linked: boolean
+attachShader(shaders:Shader[])
+compile()
+link()
+compileAndLink()
+bind()
+unBind()
+isLinked(): boolean
+delete()

ProgramLinkException
-serialVersionUID: long = -8475803770305624351L

ShaderProgramInfo
-locations: Map<String, Integer> = new HashMap
+location(name:String): int
+getLocation(name:String, location:int): ShaderProgramInfo

TriangleVBO
+id: int
+unitTriangle: TriangleVBO
+reset()
+draw(posLocation:int, texLocation:int)
+delete()

← -triangle  
1

Path
-arrowScale: float = 1.0f
+getArrowRotation(): float
-modelMatrix(): Matrix4f
#doDraw(FV:Matrix4f, M:Matrix4f, info:ShaderProgramInfo)
#localBoundingBox(): BoundingBox
+getPoint(i:int): Vector2f
-size(): int

Texture2D
+id: int
+width: int
+height: int
+aspect: float
+bind()

← +texture  
1

QuadVBO
-unitQuad: QuadVBO
-id: int
+reset()
+draw(posLocation:int, texLocation:int)
+delete()

← +quadVBO  
1

TexturedQuad
+phi: float
+scaleX: float = 1
+scaleY: float = 1
+drawInNDCSpace: boolean
-modelMatrix(): Matrix4f
+scaleToTexture()
#doDraw(FV:Matrix4f, M:Matrix4f, info:ShaderProgramInfo)
#localBoundingBox(): BoundingBox

← +texture  
1

Layer
+pos: Vector2f
+color: Vector4f = new Vector4f(1,1,1,0)
+transparency: float = 1.0f
+scale: float
#doDraw(FV:Matrix4f, M:Matrix4f, info:ShaderProgramInfo)
#localBoundingBox(): BoundingBox
+draw(FV:Matrix4f, M:Matrix4f, info:ShaderProgramInfo)
+boundingBox(): BoundingBox

← layers  
0..\*

GLRendererListener
<<interface>>
+onCreateGL(eglConfig:EGLOnfig)
+onDraw()
+onResizeGL(w:int, h:int)

listeners  
0..\*

GLRenderer
-listeners: GLRendererListener = new ArrayList<>()
+onSurfaceCreated(gl10:GL10, eglConfig:EGLOnfig)
+onDrawFrame(gl10:GL10)
+onSurfaceChanged(gl10:GL10, w:int, h:int)
+addListener(listener:GLRendererListener)

GLRendererAdapter
<<interface>>
+onCreateGL(eglConfig:EGLOnfig)
+onDrawGL()
+onResizeGL(w:int, h:int)

LayerUpdate
<<interface>>
+run()

TextureFactory
+texMap: Map<Integer, Texture2D> = new HashMap<>
+loadTexture(context:Context, resourceId:int): Texture2D
+reset()

BoundingBox
-min: Vector2f
-max: Vector2f
-center: Vector2f
-extent: float
+update()
-enclosePoint()
+reset()
+contains(v:Vector2f): boolean
+intersect(b:BoundingBox): boolean
+intersect(start:Vector2f, end:Vector2f): boolean
+enclose(points:Vector2f[])
+enclose(points:Collection<Vector2f>)
+enclose(boundingBox:BoundingBox)
+translate(t:Vector2f)

← -boundingBox  
1

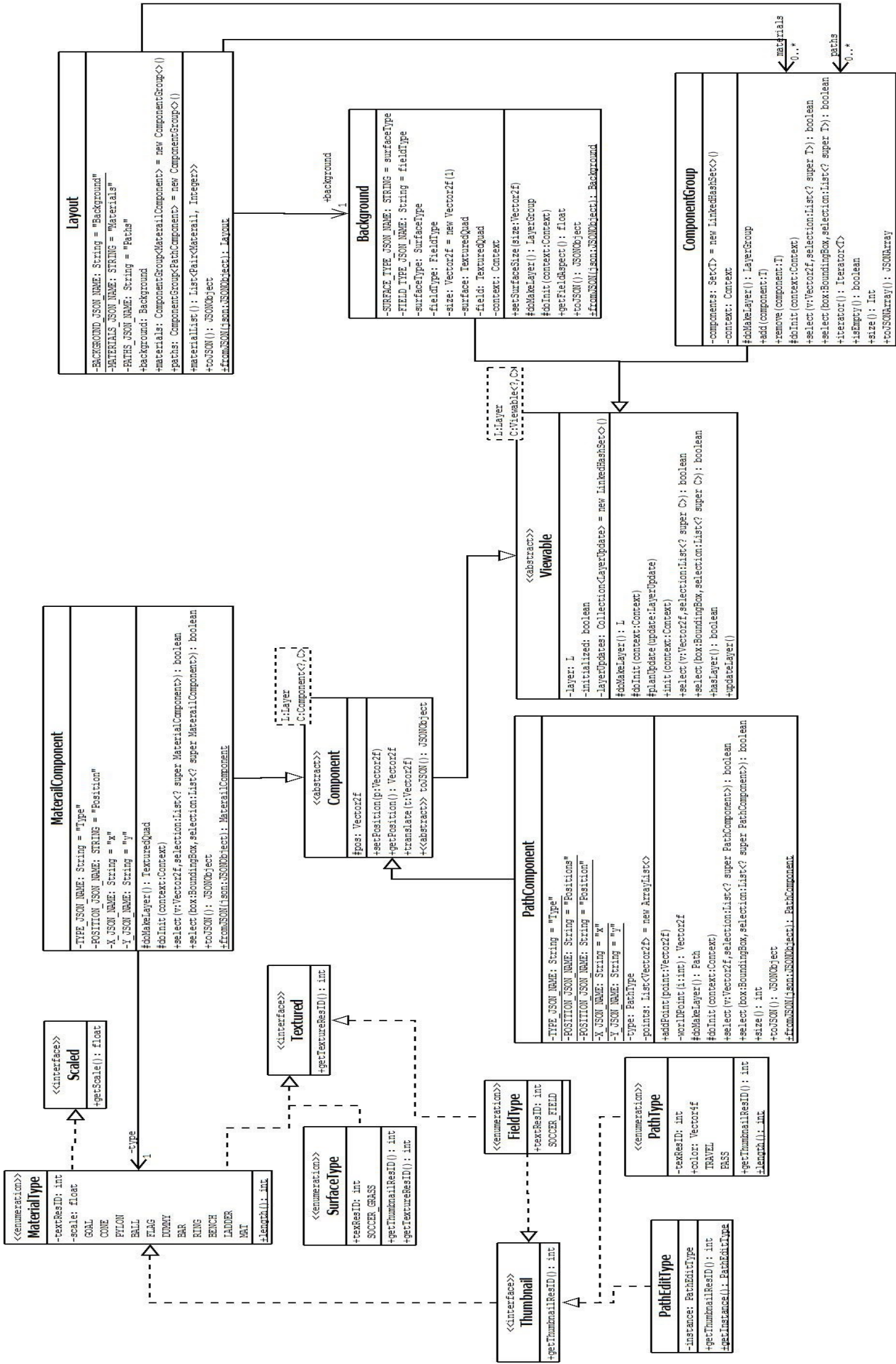
Line
-points: List<Vector2f>
-buffer: ByteBuffer
+showPoints: boolean
+updateBuffer()
+updateBoundingBox()
+update()
+getPoint(i:int): Vector2f
-end(): int
-runPoints(): int
+draw(posLocation:int, texLocation:int)
+iterator(): Iterator<Vector2f>

← +line  
1

ShaderCompileException
+serialVersionUID: long = 4444002156721040877L

ProgramNotLinkedException
-serialVersionUID: long = -6023047330211526448L





## V) Glossar

### V.a) Altersklassen nach DFB-Standard

Bambini = G-Junioren = U7  
F-Junioren = U8, U9  
E-Junioren = U10, U11  
D-Junioren = U12, U13  
C-Junioren = U14, U15  
B-Junioren = U16, U17  
A-Junioren = U18, U19, U20  
Aktive = Ü20  
Senioren = Ü35

### V.b) Schlagwörter-Katalog

#### TECHNIK:

Ballan-/mitnahme  
Dribbeln  
Finten(Ausspielen)  
Köpfen  
Passen  
Schiessen

#### FITNESS:

Ausdauer  
Koordination  
Schnelligkeit

#### TAKTIK:

Gruppe(Angriff)  
Gruppe(Ball erobern)  
Gruppe(Ballzirkulation)  
Gruppe(Tore verhindern)  
Gruppe(Umschalten)  
Team(als Mannschaft angreifen)  
Team(als Mannschaft verteidigen)  
Team(kompakt verteidigen)  
Team(Pressing)  
Team(Spielaufbau/Angriff)  
Team(Torchancen herausspielen)  
Team(Umschalten)  
Zweikampf(Angriff)  
Zweikampf(Verteidigung)

#### TORWARTTRAINING