מסמך תכנון

# מבנה כללי

Squeaker היא רשת חברתית לשיתוף הודעות קוליות בין משתמשים. עיקר הפעולות בה הן הקלטות של הודעות קוליות, להלן "ציוצים", ומעקב אחרי משתמשים אחרים וציוציהם. הציוצים נשמרים במסד הנתונים, כאשר משתמש מתחבר הוא מקבל עדכון על הציוצים העדכניים של המשתמשים שהוא עוקב אחריהם.

המערכת מבוססת על מודל Client-Server.

הClient הוא אפליקצית Android המותקנת על טלפונים חכמים של המשתמשים.

הServer הוא אפליקציה הכתובה בשפת JAVA האחראית לקשר בין כלל המשתמשים למסד הנתונים השומר את המידע על כלל המשתמשים.

# מבנה הClient

אפליקציית הלקוח בנויה עפ״י המבנה הסטנדרטי שמקובל בפיתוח אפליקציות אנדרואיד כאשר מחלקות Activity מייצגות מסך בממשק האפליקציה, והן מפעילות מחלקות מסוג AsyncTask על מנת לתקשר עם השרת ברקע ולעדכן את הממשק בקבלת המידע מהשרת.

בנוסף, קיימת שכבת אבסטרקציה לתקשורת עם השרת החושפת מתודות פשוטות לפעולות המוגדרות בדומיין המערכת ואובייקטים המייצגים יישויות בדומיין המערכת (domain objects) ומסתירה את מורכבות התקשורת מעל HTTP וניהול החיבור המתמשך (session). לבסוף, קיימת סדרת מחלקות עזר המגישות שירותי הקלטת והשמעת מידע קולי ואריזתו לצורה הניתנת לשידור אל השרת.

מחלקות עיקריות:

|  |  |  |
| --- | --- | --- |
| אריזה | שם מחלקה | תיאור |
| com.squeaker.app | Activity.java | מחלקת ה-activity הראשית עבור האפליקציה המכילה מתודות עזר למעבר של נתונים בין activities |
| com.squeaker.app | SqueakerActivityProtocol.java | מחלקה המכילה קוד המטפל בשליחה של נתונים בין activities במערכת |
| com.squeaker.app | LoginActivity.java | מחלקת הactivity הראשונה שהמשתמש פוגש, מאפשרת התחברות למערכת |
| com.squeaker.app | NewsFeedActivity.java | המסך הראשי של האפליקציה, מסך החדשות המוצג לאחר התחברות מוצלחת |
| com.squeaker.app | SqueakArrayAdapter.java | מחלקת adapter המתאמת בין מידע על ציוץ להצגתו ב-ListView של אנדרואיד |
| com.squeaker.squeaker | JsonServerApi.java | מחלקה האחראית על הסתרת המורכבות שבתקשורת עם שרת squeaker וחושפת מתודות פשוטות כלפי שאר האפליקציה |
| com.squeaker.squeaker | ServerState.java | אובייקט המנהל את מצב התקשורת עם השרת, מחזיק את ה-Session הנוכחי ואת מחלקת הAPI הנוכחית |
| com.squeaker.squeaker | Session.java | אובייקט המייצג חיבור מתמשך אל השרת (session). מכיל מידע המסופק ע״י השרת ודרוש כדי לקיים מולו פעולות. |
| com.squeaker.utils | AudioPlayer.java | אובייקט המשמש להשמעה של מידע קולי המגיע מן השרת |
| com.squeaker.utils | MicRecorder.java | אובייקט המשמש להקלטה של סאונד למטרות שידור ציוץ אל השרת |
| com.squeaker.utils | CompressedBase64Serializer.java | מחלקה האחראית להפיכת מידע קולי (או כל מידע בינארי אחר) לפורמט base64 deflated, צורה אשר ניתן לשדר ביעילות על גבי בקשות HTTP. |

# מבנה הServer

השרת מבוסס ברוחו על-פי מודל שכבות UI-BL-DAL-DB, לצורך הפרדת השרת לשכבות, השתמשנו בשני מודלי נתונים (Data-Models). מודל ראשון ממומש בשכבת הDAL ואחרי לספק מעין שירות ORM בסיסי לשכבת הBL. מודל שני ממומשת בשכבת הUI והוא מספק שירותים ללקוחות של השרת.

## UI

השכבה החיצונית כלפי העולם מייחצנת REST API מעל פרוטוקול HTTP. דרך ממשק זה כל הClients מקבלים את השירותים הדרושים להם מהServer. כל המתודות מקבלות ומחזירות אובייקטים בפורמט JSON.

פונקציות הREST ממומשות על-פי תקן JAX-RS, מעל שרת HTTP המסופק ע"י תשתית בשם Grizzly. המרות לJSON ובחזרה מסופקות ע"י ספריית Jackson.

המחלקות העקריות:

|  |  |  |
| --- | --- | --- |
| c | שם מחלקה | תיאור |
| Workshop.rest | RestAPI.java | המחלקה הראשית בשכבה הממשת את פונקציות הREST שהשרת מחצין החוצה. |
| Workshop.rest.datamodel | SessionIdInput.java | אובייקט קלט בסיסי ששאר האובייקטים יורשים ממנו, מכיל את מזהה החיבור של משתמש. |
| Workshop.rest.datamodel | FindUserInput.java | אובייקט קלט עבור פעולת חיפוש משתמשים |
| Workshop.rest.datamodel | FollowSqueakerInput.java | אובייקט קלט עבור בקשת מעקב אחרי משתמש |
| Workshop.rest.datamodel | LoginInput.java | אובייקט קלט עבור בקשת התחברות או הרשמה |
| Workshop.rest.datamodel | RecordSqueakInput.java | אובייקט קלט עבור בקשת הקלטת ציוץ |
| Workshop.rest.datamodel | SqueakInput.java | אובייקט קלט עבור בקשת טעינת ציוץ |
| Workshop.rest.datamodel | FindUserOutput.java | אובייקט פלט עבור בקשת חיפוש משתמש, מכיל רשימת משתמשים פוטנציאלית וכמות ציוצים |
| Workshop.rest.datamodel | SqueakInfoOutput.java | אובייקט פלט עבור בקשות מידע על ציוצים, מכיל זמן הקלטה, תאריך וכו'... |
| Workshop.rest.datamodel | SqueakDataOutput.java | אובייקט פלט הייצג את תוכן הציוץ עצמו |
| Workshop.rest.datamodel | IOConverter.java | מחלקת עזר להמרה בין אובייקטים ביצוג פנימי ליצוג חיצוני שמוחצן ללקוחות השרת. |

## Business Logic

שכבת הBL מממשת את הלוגיקה העקרית של השרת, בנוסף היא מחברת בין הקלט מהמשתמש המגיע משכבת הREST לבין שמירת הנתונים ע"י שכבת הDAL שמתחתיה.

בשכבה זו יש מטמון של משתמשים שהתחברו לשרת. כאשר משתמש מתחבר מוגרל עבורו מזהה חיבור, מרגע ההתחברות כל פעולות המשתמש דורשות את המזהה שהוגרל כדי לזהות את המשתמש. המטמון ממומש ע"י HashMap בין מזהה חיבור לבין משתמש.

בנוסף, שכבה זו מכילה אובייקטים מול שכבת הDAL, כל אובייקט אחראי על כתיבת וקריאת חלק מהData-model (יפורט בהמשך).

מחלקות עקריות:

|  |  |  |
| --- | --- | --- |
| c | שם מחלקה | תיאור |
| Workshop | Main.java | המחלקה הראשונית של הסרבר, מאתחלת את שרת הHTTP |
|  | BusinessLogic.java | המחלקה העיקרית של שכבה זו, מממשת את כל פונקציות הלוגיקה העקריות. מכילה את מטמון המשתמשים. |
|  | SessionId.java | המחלקה מייצגת את זיהוי המשתמש אחרי חיבור. מכילה מזהה חד חד ערכי של משתמש. |
|  | StringUtils.java | מחלקת עזר לעבודה מול מחרוזות |

## Data Access Layer

פירוט ...

# UML STUFF

## Class Diagram

קלאסים עיקריים של הסרבר

## Sequence Diagram

חיבור משתמש

שמירת ציוץ

עקיבה אחר משתמש

עוד דברים של העתק-הדבק ..

# Database

חפירה על Cassandra ומבנה טבלאות

חפירה קצרה על Astyanax.

# איך מעלים מערכת