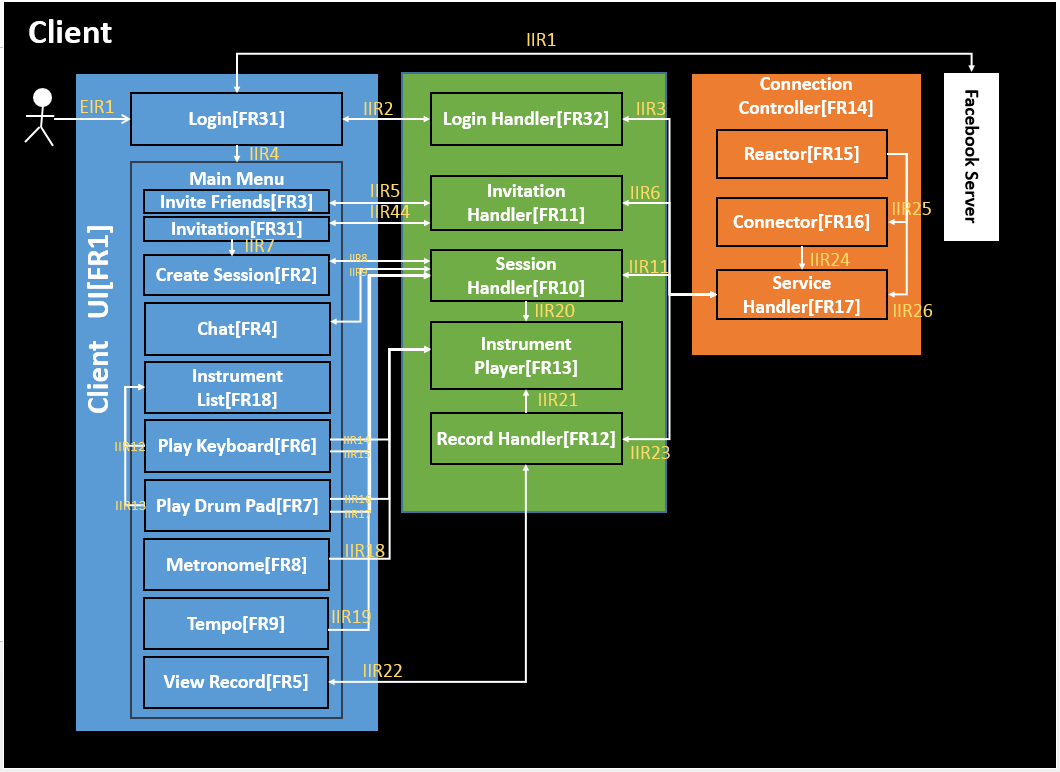
# Finger Band

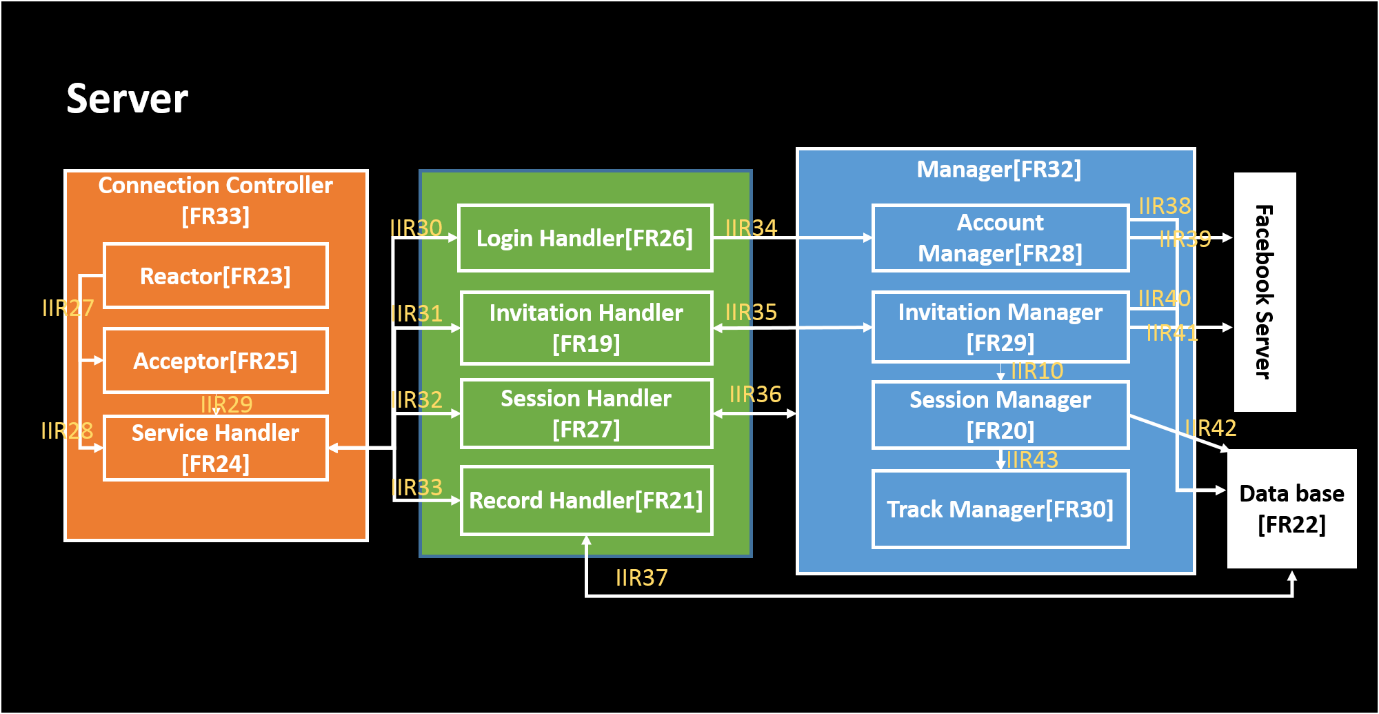
# System Requirements Specification

* **Introduction**

Finger band is a mobile service designed for people who enjoy playing music with others. With Finger band, one can co-work with friends to produce creative music without limited by instrument and environment setting. We provide a portable, time-saving and interesting jam session enviroment for players. Additionally, players can invite their facebook friends to join a session and have fun together!

* **System Architecture**





* **Requirement**
  + **Functional Requirement**

|  |  |
| --- | --- |
| Client: |  |
| CFR01 | **Client UI** |
|  | There are two activities.  Activity 1 = Login Screen  Activity 2 = Main Screen  Main Screen is divided into two areas, Menu Area and Content Area.  Each button in menu has a corresponding content. The relationship between contents and buttons are described below.  Menu Area contains the following buttons:   1. [C] create session 2. [I] invite others 3. [S] invitation list 4. [R] record list 5. [K] keybord |
| CFR02 | **CreateSession** |
|  | Press [C] button to create a session. If the session is successfully created, creators will enter a session with keyboard layout. |
| CFR03 | **InviteFriends** |
|  | Press [I] button to select friends one would like to invite. The layout contains a list of the user’s Facebook friends who has already downloaded our app. |
| CFR04 | **ViewRecord** |
|  | Press [R] button to display the <recordListUIFragment> in the Content Area, which contains a list of records from sessions that the user has participated. |
| CFR05 | **PlayKeyboard** |
|  | 1. Press [K] button to display the <keyboardUIFragment> in the Content Area, which contains a keyboard with 88 keys. 2. Each key is represented a button on the screen. If pressed, a particular sound of the selected note will be produced. 3. Each pressed key will also generate a set of data (i.e. channel, program, volume, duration, notes) to be sent to the server and be rebroadcasted immediately by the server to the rest of the session member. |
| CFR06 | **Session Handler** |
|  | 1. After login, the user can create a Session. 2. User can join a session, it will generate a track to manage one’s own music. |
| CFR07 | **Invitation Handler** |
|  | 1. Once one click [I] button, it will show one’s facebook friend list. 2. Invitation Manager can get facebook friend list with OAuth via Invitation Handler. 3. User use check box to select who he/she wants to invite. 4. After User decides the invitee list, Invitation Manager will send it to Server via Invitation Handler. 5. Friend in friend list contain one's name, portrait Url. 6. Once refresh the invitation page, Invitation Manager will get Invitation list from server via Invitation Handler. 7. Invitation contains creator, session number and session name. 8. User can accept or ignore invitation. 9. Once User accept an invitation, one will go to join a session. |
| CFR08 | **Record Handler** |
|  | 1. Once refresh the record page, Record Handler will get record list from server. 2. It will show simple information of record, for example, record play time, record create time, participants. 3. User can click Record to get MidiFile from Server via Record Handler then 4. MidiFIle contain timestamp with note. |
| CFR09 | **Login Manager** |
|  | Login Manager can let user to login with Facebook account. |
| CFR10 | **Connection Controller** |
|  | Connection controller can connect to server and manage connections for corresponding functions. |

**Server:**

|  |  |
| --- | --- |
|  |  |
| SFR01 | **Invitation Handler** |
|  | 1. **Initiate an invitation**   User can initiate an invitation in the system.  Server will create an invitation with the port number and user's account. When the invitation is initiated, sever (Invitation Manager) will record the invitation.   1. **Request friend list**   Server (Invitation Manager) can request user's friends from Facebook server.   1. **Check App availability**   Server (Invitation Manager) can check if one user has installed the App or not   1. **Record Invitation**   Server (Invitation Manger) holds all the invitation information   1. **Return User's Invitation**   Server will return users' own Invitation list   1. **Remove an invitation**   Once user accept or reject an invitation, sever will remove the invitation from user's own list   1. **Respond to user's acceptance**   After server remove the invitation from user's own list, server will send the session port back to user |
| SFR02 | **Session Handler** |
|  | 1. SessionManager will create a Session after a client send a create request. 2. After the session is created, the session will create a TrackManager for the creator. 3. The Session can recognize a TrackManager by the client FBID. |
| SFR03 | **Database** |
|  | 1. The server will be connected to the database via a DBConnector class, which contains a set of java APIs that the server can call to manipulate data in various field within the database. 2. The APIs consists of four methods for each database field:   <1> create()  <2> delete()  <3> update/set()  <4> find/get() |

* + **Internal Interface Requirement**

**Client**

|  |  |
| --- | --- |
| IIR1 | Login Module sends Facebook Server on request (access token) and receives the information of acknowledgment from Facebook. |
| IIR2 | Login Handler Module receives the information from Login Module to handle. |
| IIR3 | Login Handler can register a login service on Service Handler to allow users to login with facebook. |
| IIR4 | If user login successfully, then it can display the main menu. |
| IIR5 | Invite Friends send request Friend List to Invitation Handler and get Friend List from Invitation Handler. After user choosing, Invite Friends will send the invitee List to Invitation Handler |
| IIR6 | Inviation Handler can ask Service Handler to initiate an invitation or respond invitation. |
| IIR7 | After User accept one Invitation, user will enter a session. |
| IIR8 | 1. Create Session can create a Session Handler, 2. Session Handler will create Service Handler and register itself to the Service Handler. 3. Create Session will make Session Handler to send a request with creator FB ID and session name to Service Handler. Manager will add this <ID, Session Manager> to the Map. |
| IIR9 | When the tempo setting is changed by the creator, the session handler will tell Tempo to change |
| IIR10 | Invitation Handler can get the list of members and user being invited from session manager |
| IIR11 | 1. Session Handler can ask Service Handler to create a Session for the user. 2. Session Handler can send a sound packet to Service Handler. 3. Whenever there is a sound by other member in the session, Service Handler will send a sound packet to Session Handler. 4. Whenever the tempo in the session is changed, Service Handler will tell Session Handler. |
| IIR12 | Notify Instrument List to show |
| IIR13 | Notify Instrument List to show |
| IIR14 | Play keyboard send sound to Instrument Player |
| IIR15 | Play keyboard can send a sound info to Session Handler. |
| IIR16 | Play drum pad send sound to Instrument Player |
| IIR17 | Play drum pad can send a sound info to Session Handler. |
| IIR18 | Send tempo increase/decrease to Session Handler |
| IIR20 | Session Handler can send a sound info to Instrument Player to play a sound. |
| IIR44 | Invitation can request Invitation List from Invitation Handler and get Invitation List from Invitation Handler. |

**Server**

|  |  |
| --- | --- |
| IIR27 | Initiate three Acceptors to listen for specific ports. |
| IIR29 | Initiate ServiceHandler and assign a connection to it. |
| IIR30 | LoginHandler will use ServiceHandler to communicate to Client. |
| IIR31 | InvitationHandler will use ServiceHandler to communicate to Client. |
| IIR32 | 1. Service Handler can tell the Session Handler a user is leaving. 2. Service Handler can tell the Session Handler a user is joining. 3. Service Handler can send a sound packet to Session Handler. |
| IIR33 | RecordHandler will use ServiceHandler to communicate to Client. |
| IIR34 | Login Handler Module can ask Account Manager which Facebook account to use. |
| IIR35 | 1. Invitation Handler can send user's request to Invitation Manager for retrieving his/her own invited list. 2. Invitation Handler can also call Invitation Manager to add participants to a session. After Invitation Manager creation, Manager will return “OK”. 3. Invitation Handler can send user's response to specific Invitation to Invitation Manager. Invitation Manager will remove them from the user's invited list and return “OK”. |
| IIR36 | 1. Session Handler can call Manager to create a Session Manager. 2. Session Handler will register itself to the Session Manager. 3. Session Handler will return ACK. 4. Service Handler send request with Session ID and FB ID to Manager. 5. The Manager will find the Session Manager by Session ID, and make the Session Manager create a Track Manager for the FB. S 6. ession Manager will tell database to add the FB id in the session row. |
| IIR37 | Once the session is finished, SessionManager will merge all the records of Track Managers into a record file and store it into database. |
| IIR38 | Account manager can record users’ information in database. |
| IIR39 | Account manager can allow users to login with facebook. |
| IIR40 | Invitation Manager can look up the Database to find if one has logined or not. |
| IIR41 | Invitation Manager can send user's authorization token to Facebook Server in order to take their Facebook's friends, photos and FB name. |
| IIR42 | Session Manger will tell Database to create a row with Session name and Creator FB ID. Session Manager knows the row Index. |
| IIR43 | When the Track Manager receive a sound or chat info, it will tell Session Manager to send to other members in the session. |

**External Interface Requirement**

**Client**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Server**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |