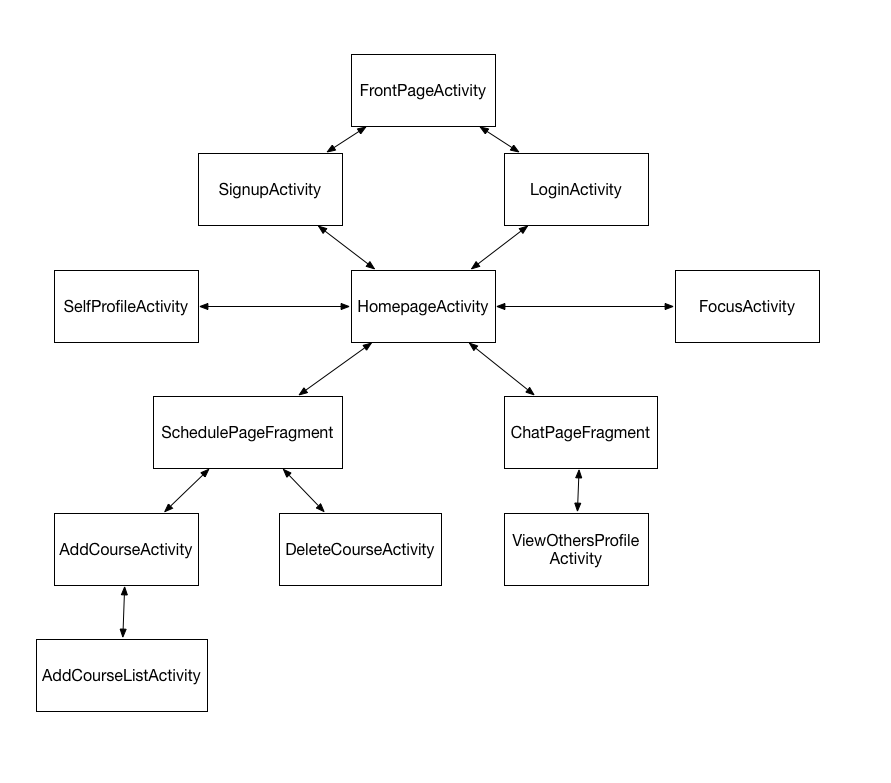
1. Designing Screens
2. Our application supports two resolutions (one for 7’’, 1200\*1920:xhdpi, one for 8.9’’, 2048\*1536:xhdpi or 10.1’’, 2560\*1600:xhdpi). It can be seen at /src/main/res/layout, and /src/main/res/layout-sw700dp.
3. All Android Resources are identified by their id in Java source code.
4. Designing Presentation Tier

Intents are used for all activities and fragments



FrontPageActivity:



SignupActibity:



LoginActivity:



SchedulePageFragment:



AddCourseActivity:



SelfProfileActivity:



FocusActivity:



ChatPageFragment:



1. Designing Content Provider
2. -Database schema:



1. Table create commands

User Table:

CREATE TABLE user (

id INT NOT NULL AUTO\_INCREMENT,

email varchar(64) NOT NULL,

password varchar(32) NOT NULL,

name varchar(64) NOT NULL,

avatar MEDIUMBLOB,

college varchar(128),

major varchar(128),

focus INT NOT NULL,

PRIMARY KEY (id)

);

Course Table:

CREATE TABLE course (

id INT NOT NULL AUTO\_INCREMENT,

num varchar(64) NOT NULL,

instructor varchar(64) NOT NULL,

name varchar(128) NOT NULL,

time varchar(128) NOT NULL,

PRIMARY KEY (id)

);

Registration Table:

CREATE TABLE registration (

id INT NOT NULL AUTO\_INCREMENT,

c\_id INT NOT NULL,

s\_id INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (c\_id) REFERENCES course(id),

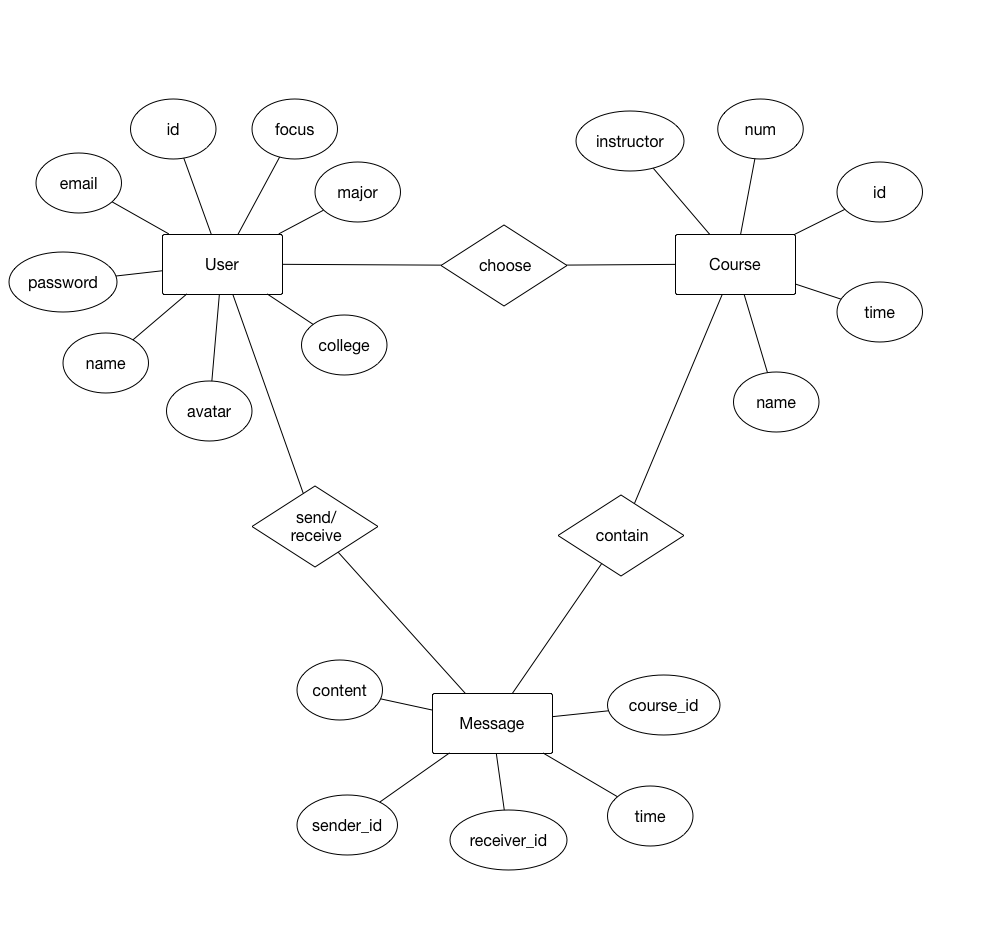
FOREIGN KEY (s\_id) REFERENCES user(id)

);

1. Java codes for entities CRUD

Please see /src/main/ws/remote. Please note that we implemented database operations on server (backend) site. The client (android app) will send API requests to server for database operation CRUD.

1. Designing Application Tier
2. Entities: user, course, chatMessage
3. Relationship: choose(user-course), send/receive (user-chatMessage), contain(course-chatMessage)
4. E-R Diagram:



1. Designing Integration Tier
2. Please see /src/main/ws/remote. Please note that we implemented database operations on server (backend) site. The client (android app) will send API requests to server for database operation CRUD.
3. Server(backend) API:

/signup

Post “email”,”password”,”name” to this page can sign up a user to the backend. If email already exist in database then return null. If success then return a new user object in json format.

/login

Post “email”,”password” to this page. If email and password match the data in database, then return a user in json format.

/allcourses

Return all courses info in json format.

/editprofile

Post user’s profile info to this page in order to update the user’s info in database

/id

Post user’s id to this page. Return the corresponding user’s info.

/registered

Post user’s id to this page. Return courses this user’s registered in json format.

/regordrop

Post user’s id and course id to this page. Register a course for this user.

/updatefocus

Post new focus value to this page. Update the value of user table.

/usersincourse

Post course id to this page. Return all users who registered this course.