# Please see the notes and update flow diagram below.

Client (Sanjeev)

1. classes to access Comcast API

1. Browse show

2. Search show

2. classes to access server API

3. UI

4. intercept the invite SMS, accept/deny invite

5. intent to go to group conversation

6. reminder and alarm

7. send a fling and play a fling

8. play the fling to Aruba (Aruba SDK)

9. Manage list of events???

Server

1. DB, data structure and data access setup (Dongliang)

2. API for client (Fred)

1. create watch group event

2. list events??

3. edit event??

4. delete event??

5. accept invite

6. deny invite

7. upload a fling video

8. download a fling video

3. classes to access VZW/comcast API (Fred)

1. send SMS message as invite

2. tune to a channel at certain time

3. check who change channel

4. monitor changing channel and send MMS (Dongliang)

Presentation (Jeff)

# Design

Objects

Event - channels, etc

Group - invitee

Fling - name, file location, id

End points:

* Create event (client -> server)
* handle invite (server)
* delete event (client -> server)
* add to group (client -> server)
* list events (client)

REST

Events

Groups

Fling

# Database

## Derby DB (10.10.1.1)

**Download**:

[http://mirrors.gigenet.com/apache/db/derby/db-derby-10.10.1.1/db-derby-10.10.1.1-bin.tar.gz](http://mirrors.gigenet.com/apache//db/derby/db-derby-10.10.1.1/db-derby-10.10.1.1-bin.tar.gz)

**Install:** Untar to $INSTALL\_DIR

(e.g. /Applications/db-derby-10.10.1.1-bin)

**Modify $HOME/.bash\_profile by adding path:**

PATH=${PATH}:/Applications/db-derby-10.10.1.1-bin/bin

**Create Database Directory:** $DB\_DIR

(e.g. mkdir -p $HOME/hackathon/db)

**Create startup script (in your PATH):** startdb

# default port is 1527, use -p <port> to change

DERBY\_OPTS="-Dderby.system.home=$HOME/hackathon/db"

export DERBY\_OPTS

nohup startNetworkServer -h 0.0.0.0 > /dev/null & 2>&1

**Create stop script (in your PATH):** stopdb

stopNetworkServer

**Start the database**

startdb

**Use “ij” tool (like sqlplus, in $INSTALL\_DIR/bin)**

<http://db.apache.org/derby/papers/DerbyTut/ij_intro.html>

**Create database “madhack”, schema “madhack”, and create tables**

**username: madhack**

**password: madhack**

ij

ij> connect 'jdbc:derby://localhost:1527/madhack;user=madhack;password=madhack;create=true';

ij> create schema madhack;

ij> run '00\_setup.sql';

ij> exit;

Note: 00\_setup.sql is located at server/install directory.

**JDBC Connection URL:** jdbc:derby://localhost:1527/madhack;user=madhack;password=madhack

**Reference:**

<https://builds.apache.org/job/Derby-docs/lastSuccessfulBuild/artifact/trunk/out/ref/index.html>

## Table GROUP\_EVENT

Columns may be added if more information is needed

CREATE SEQUENCE GROUP\_EVENT\_PK\_SEQ START WITH 1 MAXVALUE 100000000;

CREATE TABLE GROUP\_EVENT (

GROUP\_EVENT\_ID INTEGER PRIMARY KEY NOT NULL,

SHOW\_ID VARCHAR(128) NOT NULL,

CHANNEL\_ID VARCHAR(128) NOT NULL,

SHOW\_TIME TIMESTAMP NOT NULL,

MASTER\_MDN VARCHAR(128) NOT NULL,

CREATE\_TIME TIMESTAMP DEFAULT CURRENT TIMESTAMP NOT NULL

);

CREATE INDEX IDX1\_GROUP\_EVENT ON GROUP\_EVENT(SHOW\_ID);

CREATE INDEX IDX2\_GROUP\_EVENT ON GROUP\_EVENT(CHANNEL\_ID);

CREATE INDEX IDX3\_GROUP\_EVENT ON GROUP\_EVENT(SHOW\_TIME);

CREATE INDEX IDX4\_GROUP\_EVENT ON GROUP\_EVENT(MASTER\_MDN);

-------------------------------------------

INSERT INTO GROUP\_EVENT (

GROUP\_EVENT\_ID,

SHOW\_ID,

CHANNEL\_ID,

SHOW\_TIME,

MASTER\_MDN

) VALUES (

NEXT VALUE FOR GROUP\_EVENT\_PK\_SEQ,

?,

?,

?,

?

);

## Table GROUP\_MEMBER

CREATE TABLE GROUP\_MEMBER (

GROUP\_EVENT\_ID INTEGER NOT NULL,

MDN VARCHAR(128) NOT NULL,

MEMBER\_STATUS VARCHAR(128) NOT NULL

);

ALTER TABLE GROUP\_MEMBER ADD PRIMARY KEY (GROUP\_EVENT\_ID, MDN);

ALTER TABLE GROUP\_MEMBER

ADD CONSTRAINT FX1\_GROUP\_EVENT FOREIGN KEY (GROUP\_EVENT\_ID) REFERENCES GROUP\_EVENT(GROUP\_EVENT\_ID);

ALTER TABLE GROUP\_MEMBER ADD CONSTRAINT CST1\_GROUP\_MEMBER

CHECK (MEMBER\_STATUS IN ('MASTER', 'INVITED', 'ACCEPTED', 'DECLINED'));