Lucas Dachman
Jasmine Rethmann
Nihar Nandan
Jonathan Taing
Ningtian Ruan

Vision Statement

Making it easier for CU students to find and participate in events on campus

Methodologies

Agile and Peer Code Review

- Tried to work sequentially/iteratively, but would respond to a particular issue if it was a dependency
- If issue was not a dependency or didn't have high priority, would skip it and come back
- Helped each other fix bugs

Tools

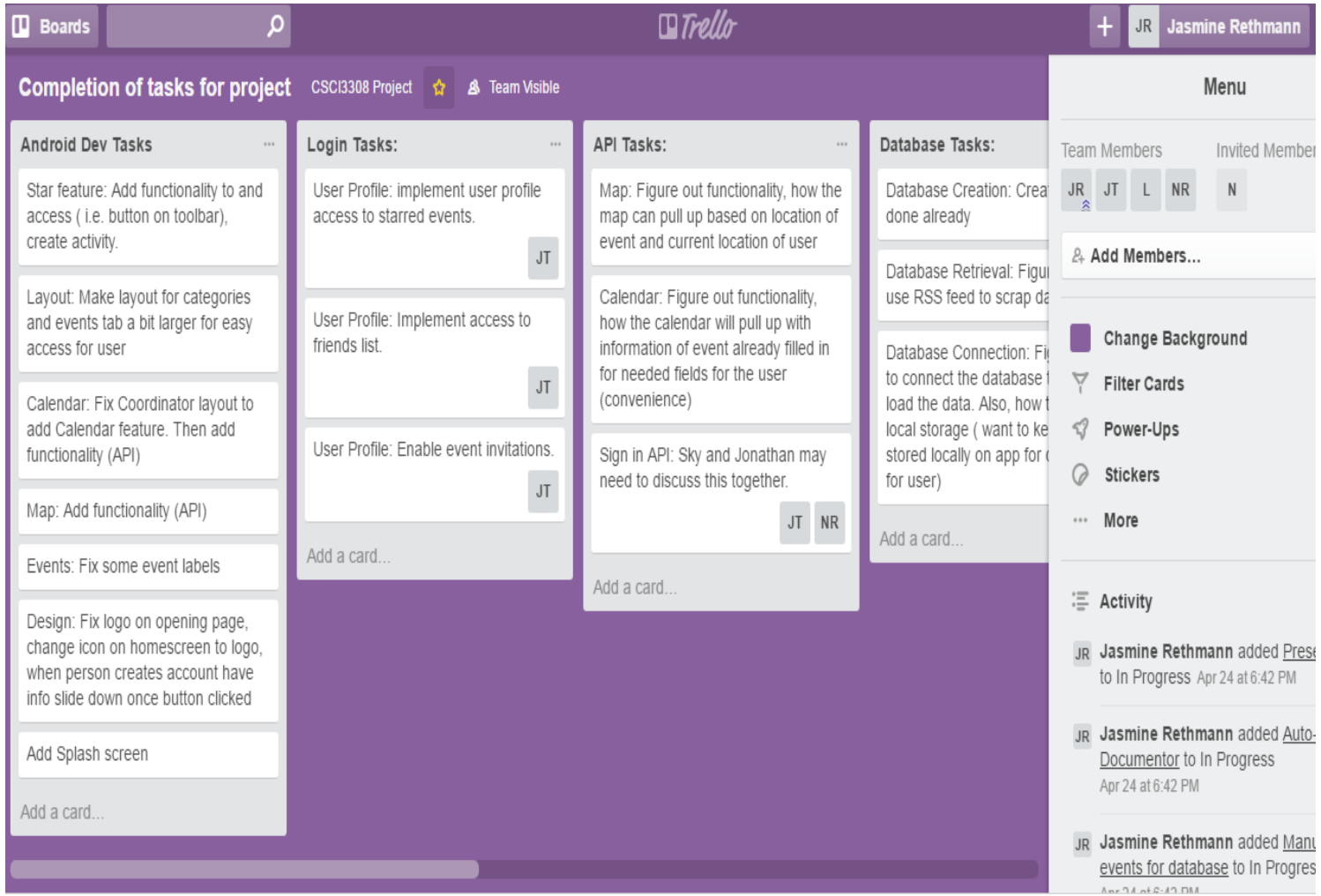
Project Tracker

Trello

Purpose:

Kept track of parts of the project that needed to be finished

3.5



VCS Repository

Purpose:
Save project-related work for group to easily access

4

Computer-Girl / 3308Project

Unwatch 4 | Star 0 | Fork 1

Code | Issues 0 | Pull requests 1 | Projects 0 | Wiki | Pulse | Graphs | Settings

Branch: master | Find file | Copy path

3308Project / LetsGoApp2 / app / src / main / java / dachman / lucas / letsgoapp2 / EventGenerator.java

LucasDachman Fixed bug where events are shown multiple times 6e3f219 4 days ago

1 contributor

81 lines (64 sloc) | 2.53 KB | Raw | Blame | History

```
1  package dachman.lucas.letsgoapp2;
2
3  import android.content.Context;
4
5  import java.util.ArrayList;
6  import java.util.Date;
7  import java.util.Random;
8  import java.util.concurrent.ThreadLocalRandom;
9
10 import dachman.lucas.letsgoapp2.Models.Category;
11 import dachman.lucas.letsgoapp2.Models.Event;
12
13 /**
14  * Created by lucas on 3/8/17.
15  * Badly written event generator class for testing
16  */
17
18 public class EventGenerator {
```

Database

Purpose:

To store information for events

4

```
public class CreateDatabase extends SQLiteOpenHelper{

    public static final String TABLE_EVENTS = "events";
    public static final String COLUMN_ID = "id";
    public static final String COLUMN_NAME = "name";
    public static final String COLUMN_ORGANIZER = "organizer";
    public static final String COLUMN_CATEGORY = "category";
    public static final String COLUMN_DESCRIPTION = "description";
    public static final String COLUMN_DATE = "date";
    public static final String COLUMN_STAR = "star";
    public static final String COLUMN_LOCATION = "location";

    private static final String DATABASE_NAME = "events.db";
    private static final int DATABASE_VERSION = 1;

    private static final String DATABASE_CREATE = "create table "
            + TABLE_EVENTS + "( " +
            COLUMN_ID + " integer primary key autoincrement, " +
            COLUMN_NAME + " text , " +
            COLUMN_ORGANIZER + " text, " +
            COLUMN_CATEGORY + " text , " +
            COLUMN_DESCRIPTION + " text , " +
            COLUMN_DATE + " text , " +
            COLUMN_STAR + " integer , " +
            COLUMN_LOCATION + "text );";

    public CreateDatabase(Context context){
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
```

Online Framework

Purpose:

To store information for events

4

D:\Dropbox\backend\3308Project\app.py - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Graph.hpp | Graph.cpp | config | djent.rp355p | REDNECK.rp355p | tsh.cc | README.txt | tsh.cc | new 9 | EventGenerator.java | app.py

```python
import os
import psycopg2
import psycopg2.extras
from flask import Flask, request, jsonify
from flask.ext.sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = os.environ['DATABASE_URL']
db = SQLAlchemy(app)

class events(db.Model):
    __tablename__ = 'events'
    id = db.Column('id', db.Integer, primary_key = True)
    name = db.Column('name',db.Text)
    category = db.Column('category',db.Text)
    location = db.Column('location',db.Text)
    organizer = db.Column('organizer',db.Text)
    description = db.Column('description',db.Text)
    date = db.Column('date',db.Text)
    star = db.Column('star',db.Integer)

listevents = events.query.all()

final = {"events":[]}

for k in listevents:
    f_list = {"id":[],"name":[],"category":[],"location":[],"organizer":[],"description":[],"date":[],"star":[]}
    f_list["id"].append(k.id)
    f_list["name"].append(k.name)
    f_list["category"].append(k.category)
    f_list["location"].append(k.location)
    f_list["organizer"].append(k.organizer)
    f_list["description"].append(k.description)
    f_list["date"].append(k.date)
    f_list["star"].append(k.star)
    final["events"].append(f_list)
#COMMENT

@app.route('/', methods=['GET','POST'])
def index():
    return jsonify(final)

if __name__ == "__main__":
    app.run()
```

Python file | length : 1,311 lines : 45 | Ln : 13 Col : 57 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

Testing

JUnit

Purpose:

Used these frameworks to develop unit tests for the code

5

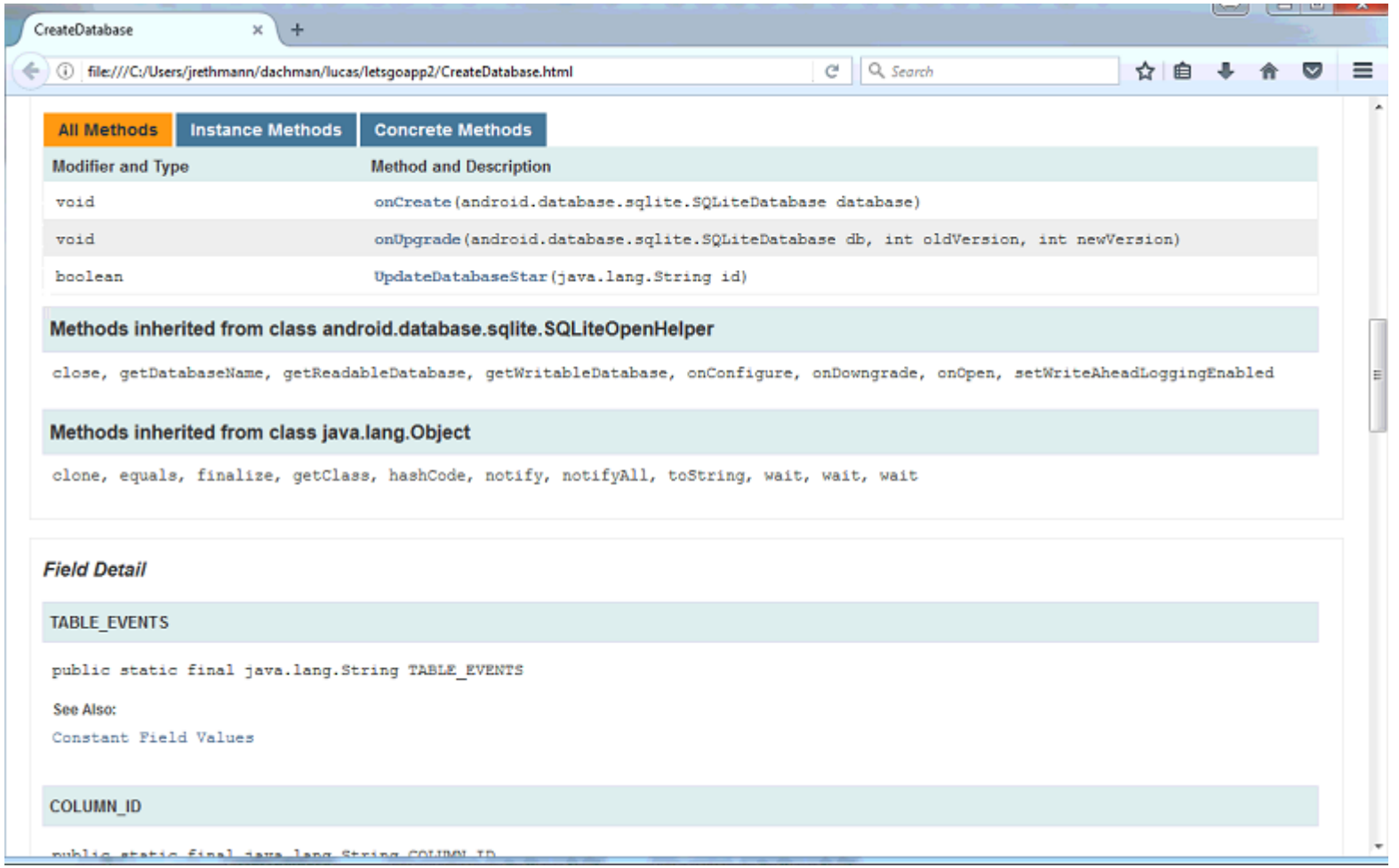
```
30  @RunWith(AndroidJUnit4.class)
31  public class EventViewActivityTest {
32
33      @Rule
34      public ActivityTestRule<EventViewActivity> eventViewActivityActivityTestRule = new
35              ActivityTestRule<EventViewActivity>(EventViewActivity.class);
36
37
38      @Test
39      public void testCase2UI()
40      {
41  //checks for UI, such as text views, the buttons toolbar for the user. .
42          Activity activity = eventViewActivityActivityTestRule.getActivity();
43
44          assertNotNull(activity.findViewById(R.id.Google_plus));
45
46          assertNotNull(activity.findViewById(R.id.star_button));
47
48          assertNotNull(activity.findViewById(R.id.content_event_view));
49
50          TextView nameView = (TextView) activity.findViewById(R.id.event_view_organizerName);
51          TextView dateView = (TextView) activity.findViewById(R.id.event_view_date);
52          TextView locationView = (TextView) activity.findViewById(R.id.event_view_location);
53          TextView depView = (TextView) activity.findViewById(R.id.event_view_description);
54
55          assertTrue(nameView.isShown());
56          assertTrue(dateView.isShown());
57          assertTrue(locationView.isShown());
58          assertTrue(depView.isShown());
59
```

Auto-Documenter

Purpose:

To gather all comments and summaries of code

4

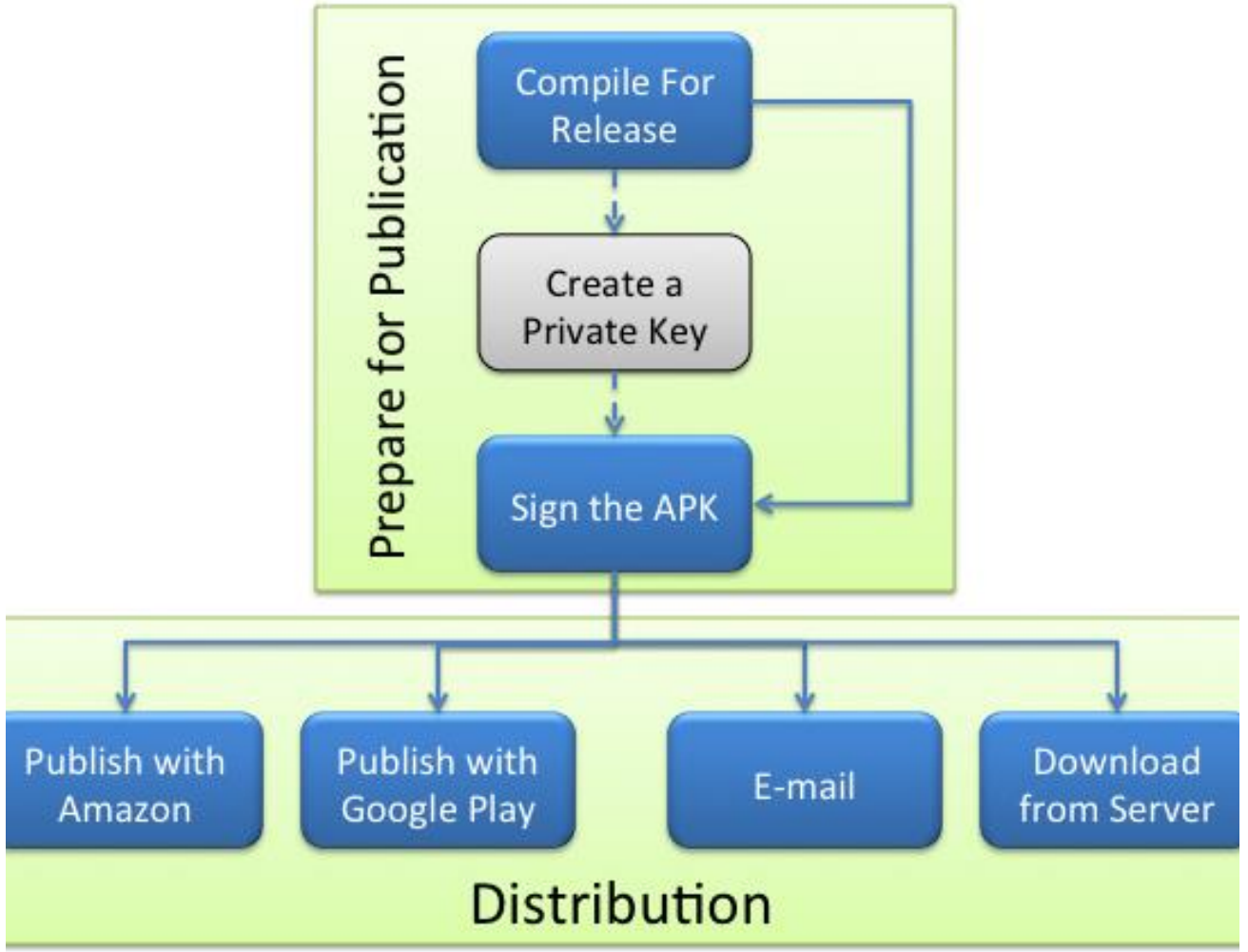


Deployment Environment

Purpose:

To test and run the app

4



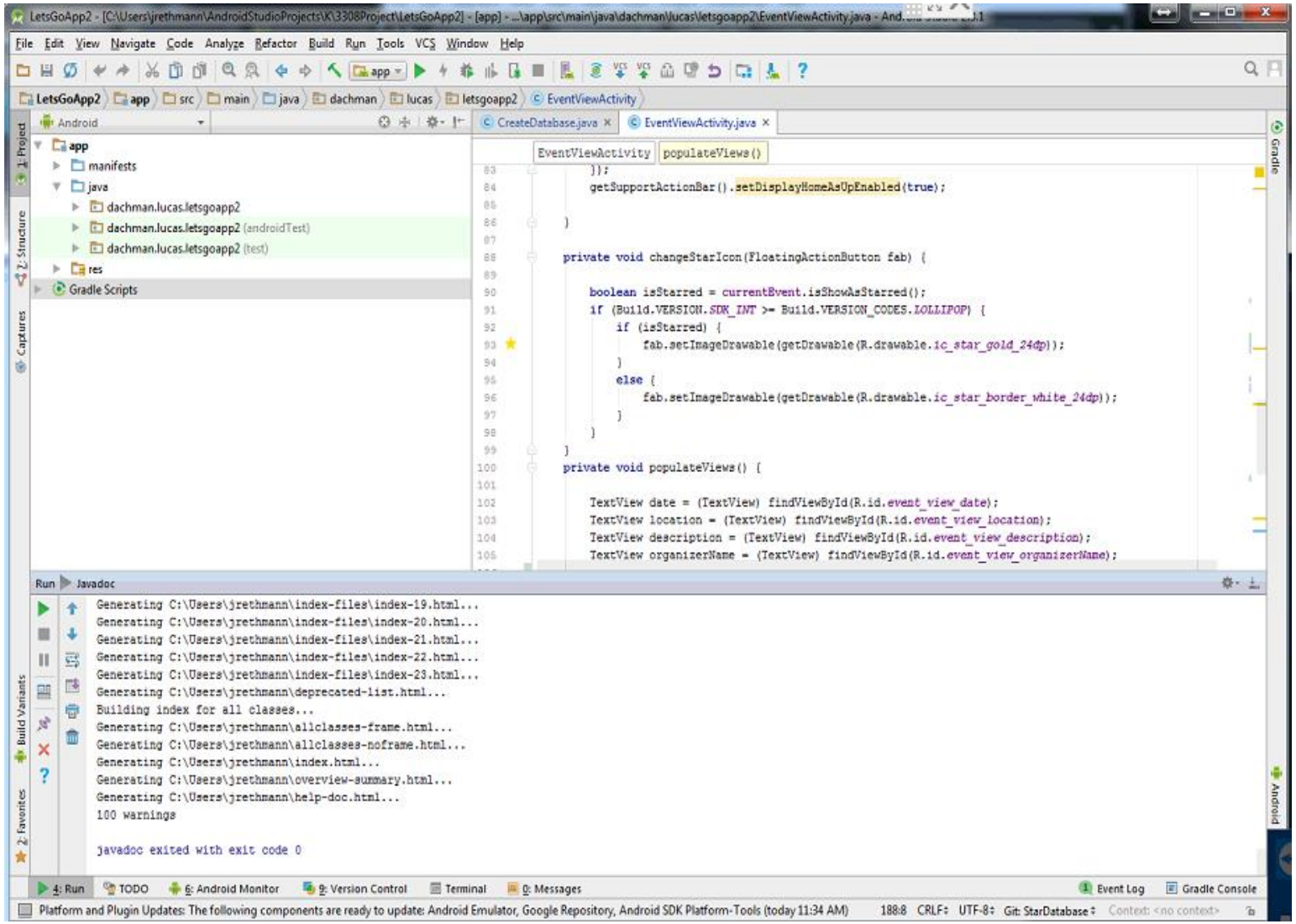
Framework

Android Studio

Purpose:

We used this to develop the code needed for the app

3



Challenges

Challenges

- Integration
  - We took care of different parts separately, so integrating them caused conflicts/bugs
  - Used peer review in person/team viewer to fix
- Friends feature
  - More complex to develop
  - Decided that it wasn't needed for the purpose of the app
- Off campus events
  - More concerned with on campus events
  - Didn't deem it to be necessary for original purpose

Future Improvements

Future Improvements

- Add a friends feature
- Try a different Framework for development
- Use the project tracker more efficiently
- Add an option for off campus events

Questions so far?

Demo