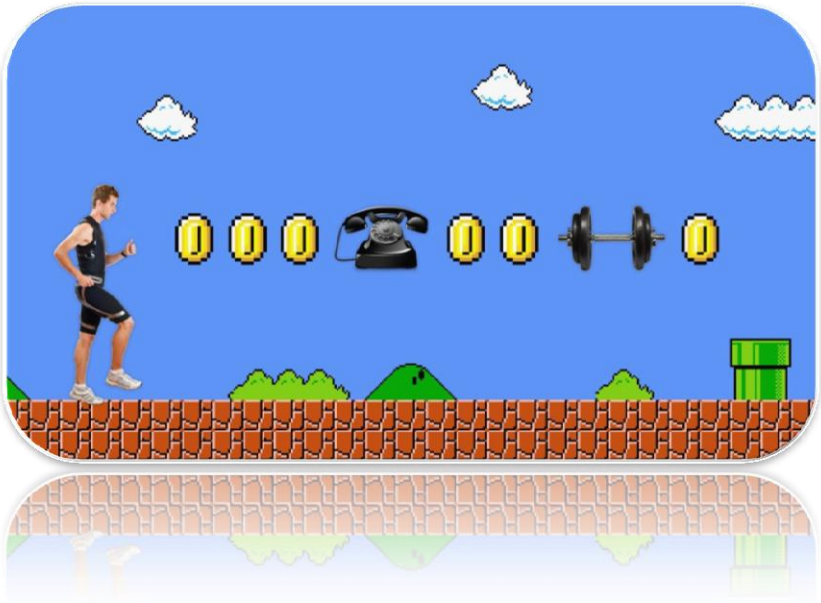


GAMIFY LIFE

CS315 Project Report

Sakaar Khurana - 10627

Department of Computer Science and Engineering, IIT Kanpur

T: 9560391587 **E:** sakaar@iitk.ac.in

Goal of the Project

Project 'Gamify Life' builds an application, with an efficient Database and a convenient command line interface, of immense daily utility which can be used as a great tool to incentivize daily jobs and, aid in cultivating good habits and can help keep track of goals and milestones.

Description of the Application

The application for each user is a personalized system that operates along a daily exchange of productivity and reward. The basic idea is to fill life with incentives to make you do the things you don't want to do. Seeing that habits develop when one is consistently doing small things on a regular basis, the application also provides an aid in developing good habits. It also keeps track of goals and milestones rooted at performing on daily basis.

A detailed description on the idea and concept of the application can be found on following link:
http://www.lifehacker.com/5975824/gamify-your-life-a-guide-to-incentivizing-everything

Implementation Details

Database: PostgreSQL

Application Programming Language: Python

Libraries used: psycopg2 (PostgreSQL library for python), argparse (formatting command line arguments) and prettytable (outputting the query results in user-friendly manner)

(**Snapshot**: help menu for the record and user feature of the app. Similar interface for other features like activities goals and milestones.)

```
usage: app.py record [-h]
                     [-dl | -sl SPECIFICLOG | -al ACTIVITYLOG ACTIVITYLOG]
                     [-sd SHOWDATE] [-st] [-ss SHOWSUMS SHOWSUMS] [-hs] [-ls]

optional arguments:
  -h, --help            show this help message and exit
  -dl, --dailylog       Use it to set quantities for various activities which
                        are marked active
  -sl SPECIFICLOG, --specificlog SPECIFICLOG
                        Use it to set quantities for various activities which
                        are marked active for a specified date(mm-dd-yyyy)
  -al ACTIVITYLOG ACTIVITYLOG, --activitylog ACTIVITYLOG ACTIVITYLOG
                        Use it to set quantities for various activities which
                        are marked active for a specified date(mm-dd-yyyy)
  -sd SHOWDATE, --showdate SHOWDATE
                        use it to show the points of various activities on
                        that date
  -st, --showtoday      use it to show the points of various activities today
  -ss SHOWSUMS SHOWSUMS, --showsums SHOWSUMS SHOWSUMS
                        use it to show the total points on a range of days
  -hs, --highscore      use it to know the highest score till now
  -ls, --lowscore       use it to know the lowest score till now
```

```
usage: app.py user [-h] [-su | -lo | -da]

optional arguments:
  -h, --help        show this help message and exit
  -su, --signup     Use this option to signup
  -lo, --logout     Use this option to logout
  -da, --deleteacc  Use this option to delete an account
```

Features and Functionality Implemented

1. **Account for each user:** Each user can create an account and have a password.

2. **Make activity list and associate points with them:** User can make a list of all activities he doesn't like doing or stop doing or like to control it etc. and then associate points with them. Points can be given on an interval or quantity/amount of activity done.

(**Snapshot**: adding an activity.)

```
What Type of Activity would you like to create?
1. Interval activity (Different points for qty in different intervals)
2. Scaled activity (Points are automatically scaled from the value and qty you will specify)
3. Yes/No activity (You either get the points for doing the activity or not doing it)
1
Adding Interval activity:
Give Activity Name: demo
Give a nickname to refer to the activity: demo
Give a default qty for the activity: 3
Should we add this to list of active activities(to be prompted for daily)?(y/n): y
Adding interval for: demo
Enter the Starting quantity of Interval: 5
Enter the Ending quantity of Interval: 4
Starting qty can't be greater than ending quantity.
Please enter again.
Enter the Starting quantity of Interval: 4
Enter the Ending quantity of Interval: 5
Enter Points associated with this interval: 1
Do you want to enter more intervals for this activity?(y/n): y
```

3. **Support for different types of Activities:** Considering the different types of activities a user may do application provides interface for creating activities of different types, each of which provides different mechanisms for logging its points.

(**Snapshot**: example of list of different activities for a user.)

```
Welcome Sakaar Khurana

Interval Activities for sakaar are:
+-------+----------+-----------+---------+-------+
| aname | nickname | qty_start | qty_end |  pts  |
+-------+----------+-----------+---------+-------+
| sleep |  sleep   |    0.0    |   3.99  |  0.0  |
| sleep |  sleep   |    4.0    |   7.0   |  10.0 |
| sleep |  sleep   |    7.01   |   8.0   |  3.0  |
| sleep |  sleep   |    8.01   |   10.0  | -10.0 |
| sleep |  sleep   |   10.01   |   24.0  | -40.0 |
+-------+----------+-----------+---------+-------+

Scaled Activities for sakaar are:
+---------------+---------------+-----+-------+
|     aname     |    nickname   | qty |  pts  |
+---------------+---------------+-----+-------+
| Hours Studied | hours_studied | 1.0 |  15.0 |
|  Time Wasted  |     twaste    | 1.0 | -15.0 |
+---------------+---------------+-----+-------+

YesNo Activities for sakaar are:
+-------------+-------------+---------+--------+
|    aname    |   nickname  | yes_pts | no_pts |
+-------------+-------------+---------+--------+
|     Bath    |     bath    |   5.0   | -10.0  |
|  breakfast  |  breakfast  |   5.0   | -10.0  |
|   medicine  |   medicine  |   15.0  |  0.0   |
| dhanur_talk | dhanur_talk |   10.0  |  0.0   |
+-------------+-------------+---------+--------+
```

4. **Daily and date specific records/logs:** User can then, on a daily basis record which activity he did and in what quantity and points will automatically be logged for that day.

(**Snapshot**: Sample adding a daily log.)

```
Welcome Sakaar Khurana
Last entry at 2013-04-11 10:51:50.522588

Log for 2013-04-11
+------------+---------------+-----+--------+
|    date    |    activity   | qty | points |
+------------+---------------+-----+--------+
| 2013-04-11 |   breakfast   | 1.0 |  5.0   |
| 2013-04-11 |    medicine   | 0.0 |  0.0   |
| 2013-04-11 |      bath     | 1.0 |  5.0   |
| 2013-04-11 | hours_studied | 2.0 |  30.0  |
| 2013-04-11 |  dhanur_talk  | 0.0 |  0.0   |
| 2013-04-11 |     twaste    | 0.0 |  -0.0  |
| 2013-04-11 |     sleep     | 7.5 |  3.0   |
+------------+---------------+-----+--------+

Log for 2013-04-11
+------------+--------+
|    date    | points |
+------------+--------+
| 2013-04-11 |  43.0  |
+------------+--------+

Entering daily log for: sakaar
(enter qty to add to existing, $qty to set qty for that acttivity, leave blank to keep it same(default if not enetered))
sleep: 3
hours_studied: 1
twaste: 0
bath: 1
dhanur_talk: 1
breakfast: 1
medicine:
```

4. **High scores(and low scores):** User can see his/her high scores for days/weeks/months.

(**Snapshot**: high scores.and interval scores)

```
Log from 2013-04-04 to 2013-04-13
+------------+--------+
|    date    | points |
+------------+--------+
| 2013-04-10 | 110.0  |
| 2013-04-09 |  53.0  |
| 2013-04-08 |  40.0  |
| 2013-04-11 |  25.0  |
+------------+--------+

The highscore(s) till now is(are):
+------------+--------+
|    date    | points |
+------------+--------+
| 2013-04-10 | 110.0  |
+------------+--------+
```

5. **Cultivate Habits:** A key feature would be an aid in cultivating a habit(for a particular activity), in which user will specify the condition under which habit would be said to be made and the application will track if that condition is met or not.

(**Snapshot**: Adding a habit)

```
Enter an activity(nickname) from following activities for which you want to make a habit:
['sleep', 'hours_studied', 'twaste', 'bath', 'dhanur_talk', 'breakfast', 'medicine'] : breakfast
Enter the description for the habit (if any):
Give a habit name(unique for each habit): brfast
Enter the date for Habit to start (in MM-DD-YYYY format (eg. 7-19-2012)): 4-9-2013
Enter the type of habit from the following options:
(daily, weekly, monthly)
daily
Is this habit inverse type? (i.e. more you do worse it is)(y/n): n
Enter the duration to monitor the making of habit (eg if type is weekly then enter 7 for 7 weeks): 5
Enter the quantity for which you you would consider the habit to be formed (qty per habit type): 1
Enter the relaxed quantity for the habit to be made(optional):
Enter the relaxes(per habit type) for the habit allowed(optional):
Enter the misses allowed for the habit(optional):
Habit Successfully added!
Do you want to enter another habit?(y/n):
```

6. **Review progress:** User will have access to progress of habits, faliures in cultivating habits and accomplished habits.

(**Snapshot**: Showing Habit Progress and status.)

```
Progressing Habits:
+------------+---------------+------------+--------+------+-----------+--------------+-------------+------+-----------+
| Habit Name |    Activity   | Start Date |  Type  | Done | Remaining | Relaxes Left | Misses Left | Amt  | Amt Left  |
+------------+---------------+------------+--------+------+-----------+--------------+-------------+------+-----------+
|   bfast    |   breakfast   | 2013-04-09 | daily  |  2   |     21    |      0       |      1      | 1.0  |    0.0    |
|   twaste   |     twaste    | 2013-04-08 | weekly |  0   |     10    |      1       |      0      | 12.0 |    3.0    |
|   sleep    |     sleep     | 2013-04-08 | daily  |  3   |     57    |      8       |      1      | 7.0  | Exc: 0.5  |
|   hstudy   | hours_studied | 2013-04-08 | weekly |  0   |     10    |      1       |      0      | 60.0 |    35.0   |
|    bath    |      bath     | 2013-04-09 | daily  |  3   |    177    |      0       |      10     | 1.0  |    0.0    |
+------------+---------------+------------+--------+------+-----------+--------------+-------------+------+-----------+

Failed Habits:
+------------+-----------+------------+-------+------------+--------------+
| Habit Name |  Activity | Start date |  Type | Failed on  | Periodic Amt |
+------------+-----------+------------+-------+------------+--------------+
|   brfast   | breakfast | 2013-04-09 | daily | 2013-04-09 |     1.0      |
+------------+-----------+------------+-------+------------+--------------+
```

7. **Set goals:** User can set milestones in terms of total points scored per week/day/month or activity milestones (in terms of quantity).

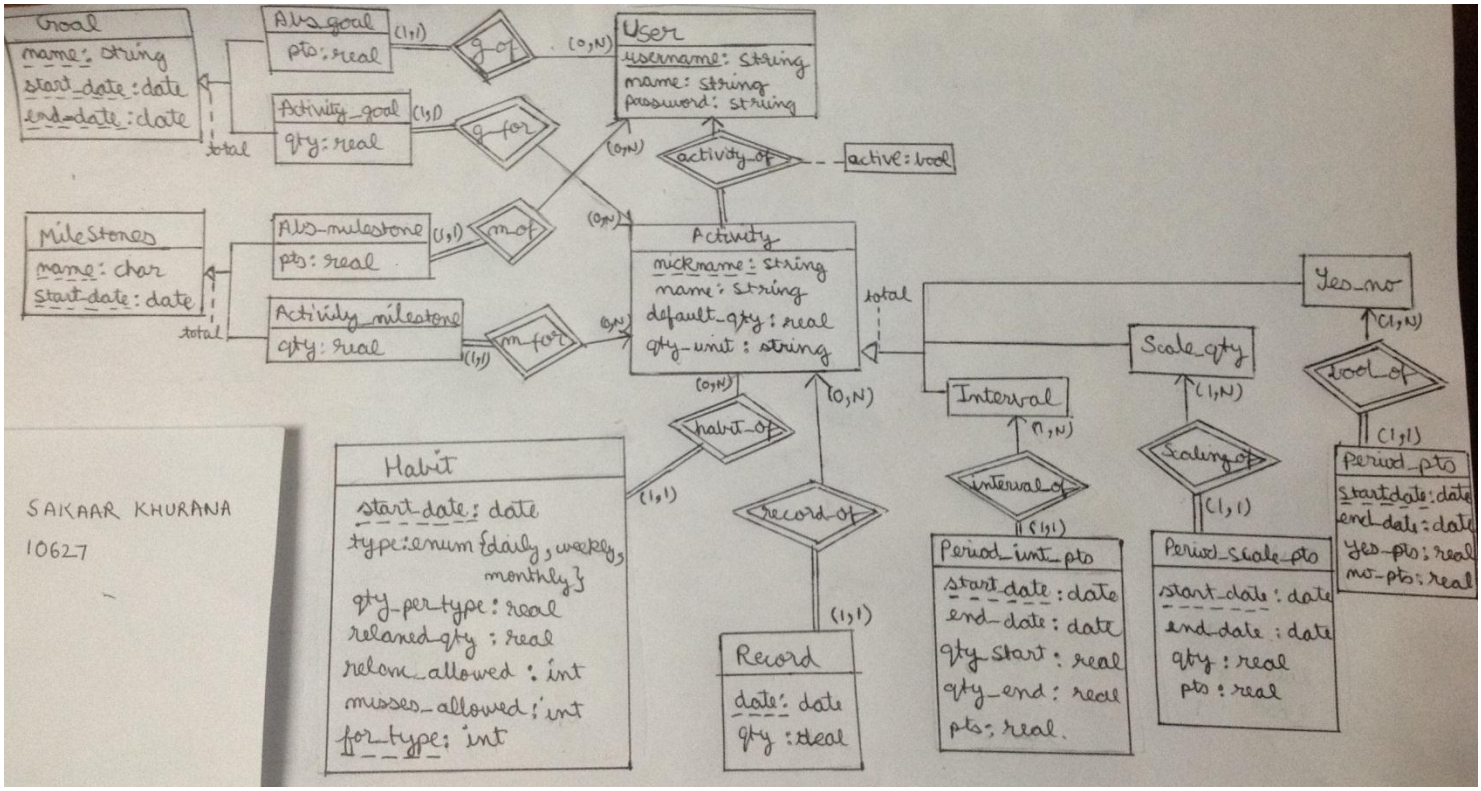
Progressing Goals:

Goal Name	Activity	Start Date	End Date	Goal	Current Total
hrs_studied	hours_studied	2013-04-08	2013-04-14	85.0	26.0
bfast	breakfast	2013-04-08	2013-05-01	20.0	3.0
twaste	twaste	2013-04-09	2013-04-20	8.0	7.5

8. **Set milestones:** User can also set milestones similar to goals.

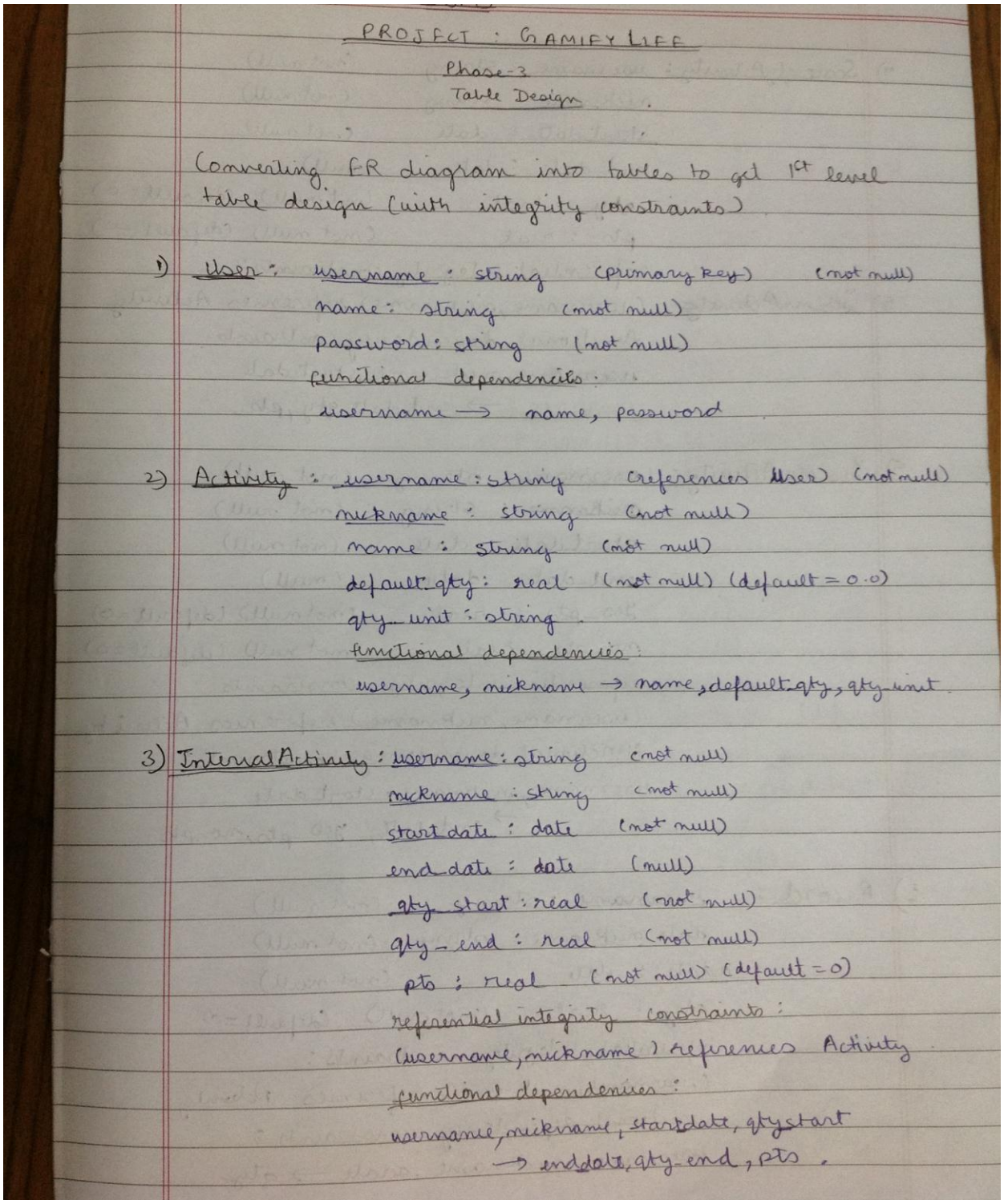
10. **Flexibility in giving points to activities:** User can change the points associated with an activity and all the scores and milestones achieved will be accordingly changed.

Extended Entity Relationship Model



Database Design

Converting ER Diagram into tables and list of all the functional dependencies and integrity contraints:



PROJECT : GAMIFY LIFE

Phase-3

Table Design

Converting ER diagram into tables to get 1st level table design (with integrity constraints)

1) User : username : string (primary key) (not null)
name : string (not null)
password : string (not null)
functional dependencies :
username → name, password

2) Activity : username : string (references User) (not null)
nickname : string (not null)
name : string (not null)
default_qty : real (not null) (default = 0.0)
qty_unit : string
functional dependencies :
username, nickname → name, default_qty, qty_unit

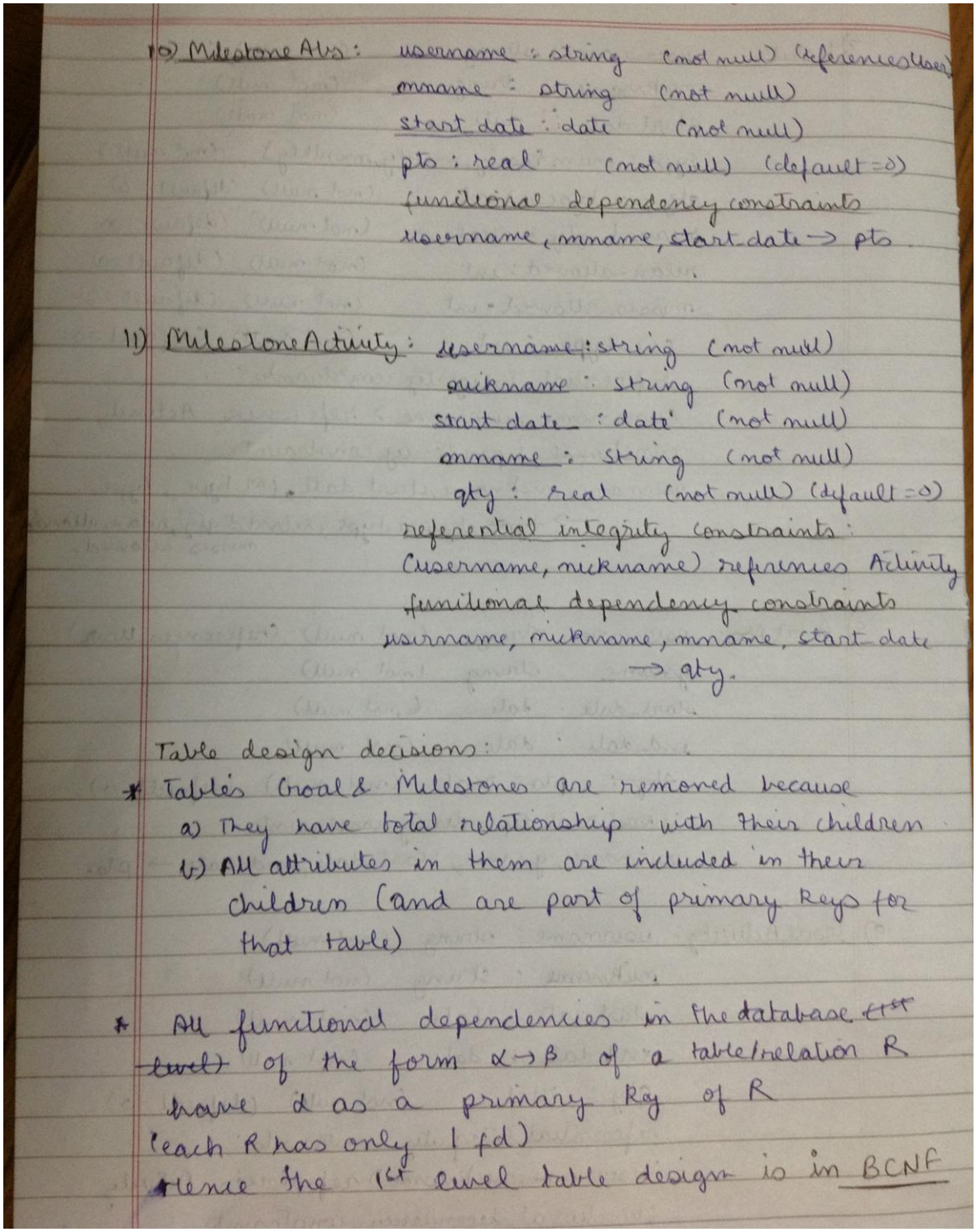
3) InternalActivity : username : string (not null)
nickname : string (not null)
start_date : date (not null)
end_date : date (null)
qty_start : real (not null)
qty_end : real (not null)
pts : real (not null) (default = 0)
referential integrity constraints :
(username, nickname) references Activity
functional dependencies :
username, nickname, start_date, qty_start
→ end_date, qty_end, pts

4) Scale_qtyActivity : username : string (not null)
nickname : string (not null)
start_date : date (not null)
end_date : date (null)
qty : real (not null) (default = 0)
pts : real (not null) (default = 0)
referential Integrity constraints:
(username, nickname) references Activity
functional dependency constraints:
username, nickname, start_date
→ end_date, qty, pts.

5) ~~Yes_no Activity~~ :

5) Yes_no Activity : username : string (not null)
nickname : string (not null)
start_date : date (not null)
end_date : date (null)
Yes_pts : real (not null) (default = 0)
no_pts : real (not null) (default = 0)
referential integrity constraints :
(username, nickname) references Activity
functional dependency constraints:
username, nickname, start_date
→ end_date, yes_pts, no_pts.

6) Record : username : string (not null)
date nickname : string (not null)
adate : date (not null)
qty : real (not null) (default = 0)
referential integrity constraints:
(username, nickname) references Activity.
functional dependency constraints :
username, nickname, adate → qty


10) Milestone Acts: username : string (not null) (references User)
mname : string (not null)
start_date : date (not null)
pts : real (not null) (default = 0)
functional dependency constraints
username, mname, start_date → pts

11) MilestoneActivity: username: string (not null)
nickname : string (not null)
start_date : date (not null)
mname : string (not null)
qty : real (not null) (default = 0)
referential integrity constraints:
(username, nickname) references Activity
functional dependency constraints
username, nickname, mname, start_date → qty.

Table design decisions:
* Tables Goal & Milestones are removed because
a) They have total relationship with their children
b) All attributes in them are included in their children (and are part of primary keys for that table)

* All functional dependencies in the database ~~level~~ of the form α → β of a table/relation R have α as a primary key of R
(each R has only 1 fd)
Hence the 1st level table design is in BCNF

Normalization

The LHS of all the functional dependencies in the above design are candidate keys of their respective relations. Hence the database is in BCNF.