

Collaborative Labs

Rubén Heradio, Luis de la Torre

May 25, 2011

Contents

1	Requirements Specification	2
1.1	A snapshot of the collaborative system	2
1.1.1	System specification in Alloy	2
1.1.2	Random scenarios	2
1.2	System interaction according to the user roles	2
1.2.1	Master User	2
1.2.2	Non-master User	2
2	Entity-relationship model	6
3	Relational Model	6
4	Proposals	6
4.1	Eliminating most input parameters for the applet	6
4.2	Eliminating <code>createHtmlStudent</code>	7

1 Requirements Specification

1.1 A snapshot of the collaborative system

1.1.1 System specification in Alloy

Figures 1 and 2. Note we have defined the role *Master User* (i.e., *Master User* is a specific kind of *User*). This way, we can constraint the users who have permission to start a collaborative session (e.g., everyone, only teachers...).

```
1 sig User {
2     participates: lone CollaborativeSession,
3     is_invitated: set CollaborativeSession
4 }
5 sig MasterUser extends User {
6     starts_and_finishes: lone CollaborativeSession
7 }
8
9 abstract sig Lab {}
10 sig VirtualLab, RemoteLab extends Lab {}
11
12 sig CollaborativeSession {
13     shares: one Lab
14 }
15
16 fact every_session_is_started_by_some_master_user {
17     all s: CollaborativeSession | some u: MasterUser |
18         s in u.starts_and_finishes
19 }
20
21 fact starts_and_finishes_is_a_subset_of_participates {
22     starts_and_finishes in participates
23 }
24
25 fact the_user_who_starts_a_session_is_not_invited_to_it {
26     all u: User | all s: CollaborativeSession |
27         s in u.starts_and_finishes => s not in u.is_invitated
28 }
29
30 fact every_user_who_participates_in_a_session_has_also_been_invited {
31     (participates - starts_and_finishes) in is_invitated
32 }
33
34 fact a_remote_lab_cannot_be_shared_among_sessions {
35     all rl: RemoteLab | no disj s', s'': CollaborativeSession |
36         (rl in s'.shares) and (rl in s''.shares)
37 }
```

Figure 1: System specification in Alloy

1.1.2 Random scenarios

Figures 3 and 4.

1.2 System interaction according to the user roles

1.2.1 Master User

Figures 5 and 6.

1.2.2 Non-master User

Figure 7 and 8.

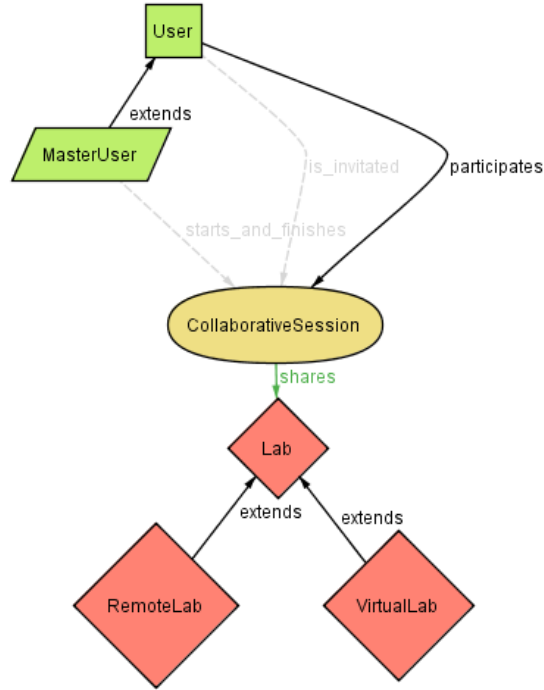


Figure 2: Graphical representation of the specification

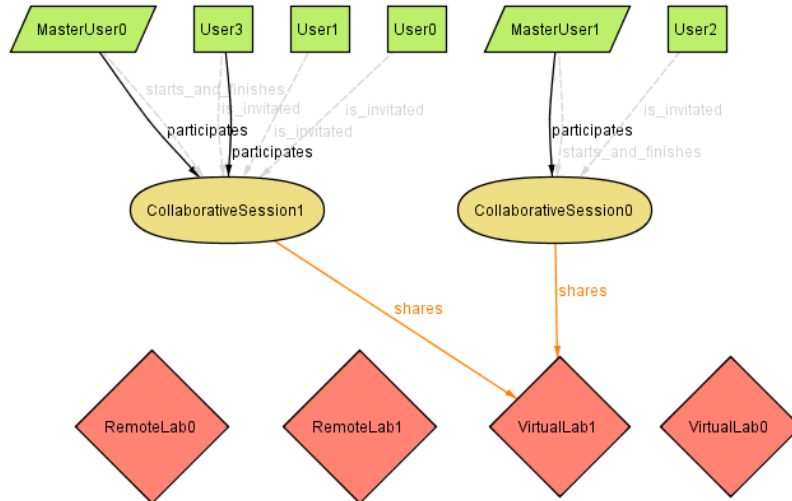


Figure 3: Random scenario according to the specification (1)

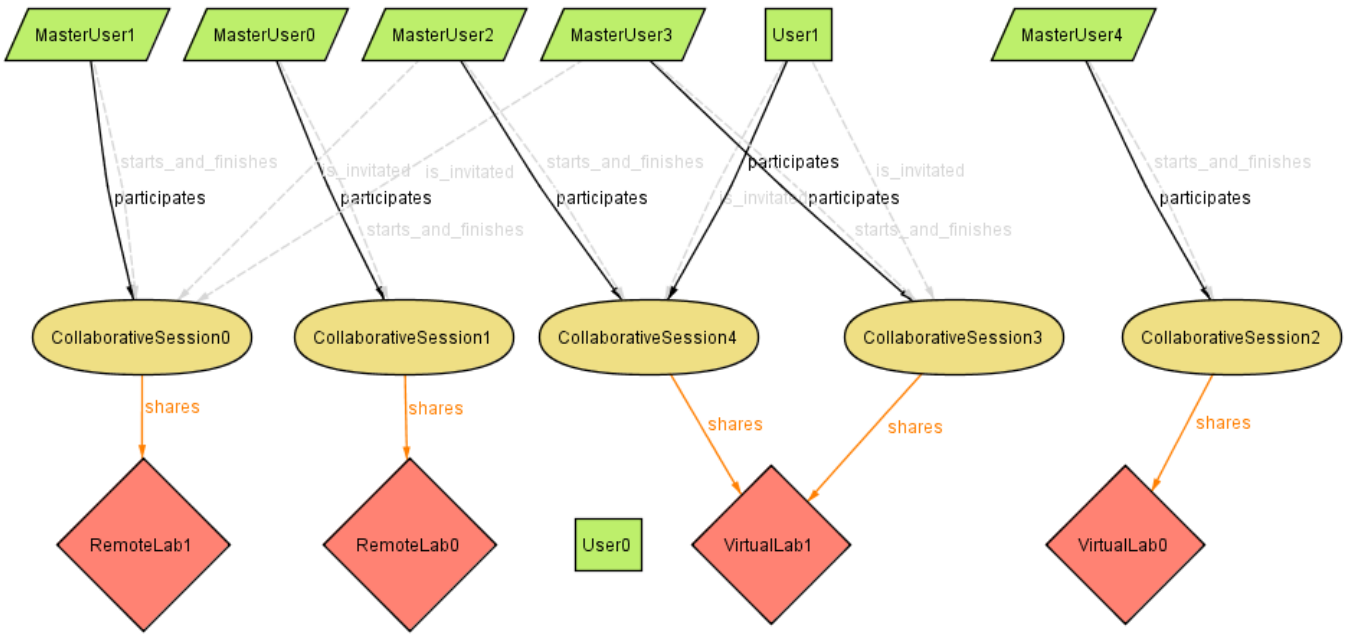


Figure 4: Random scenario according to the specification (2)

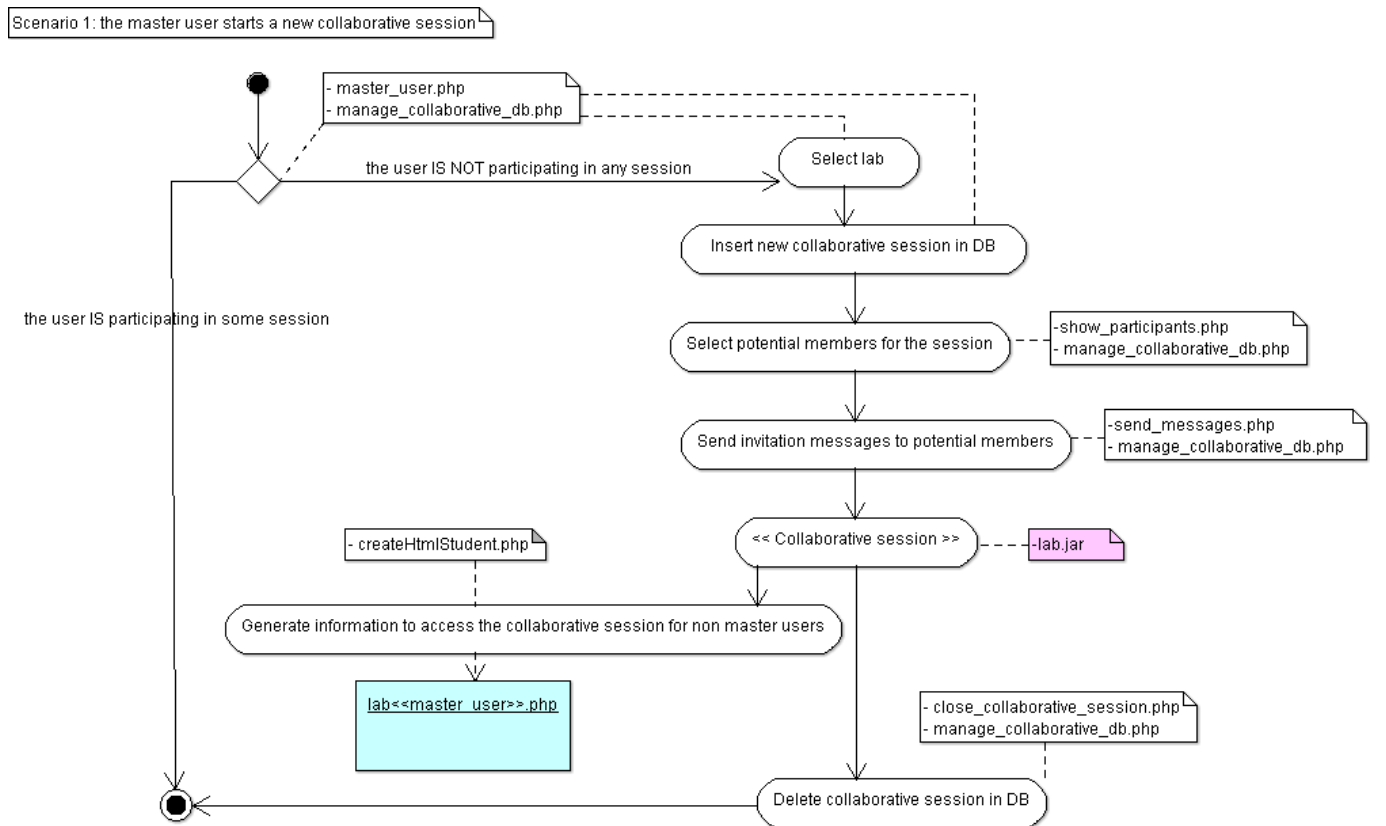


Figure 5: Activity diagram that depicts how the master user interacts with the system

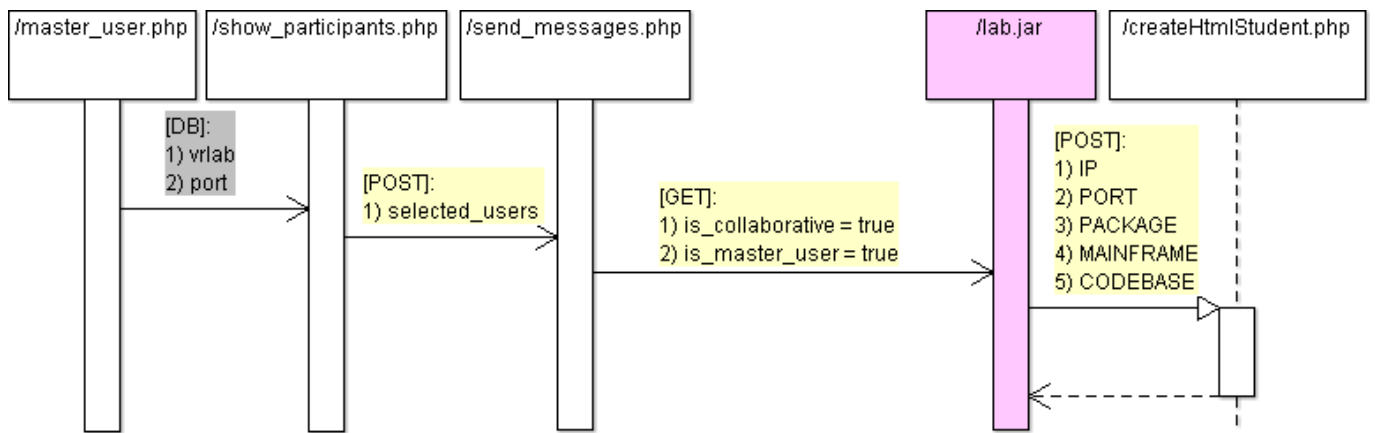


Figure 6: Sequence diagram that depicts how the master user interacts with the system

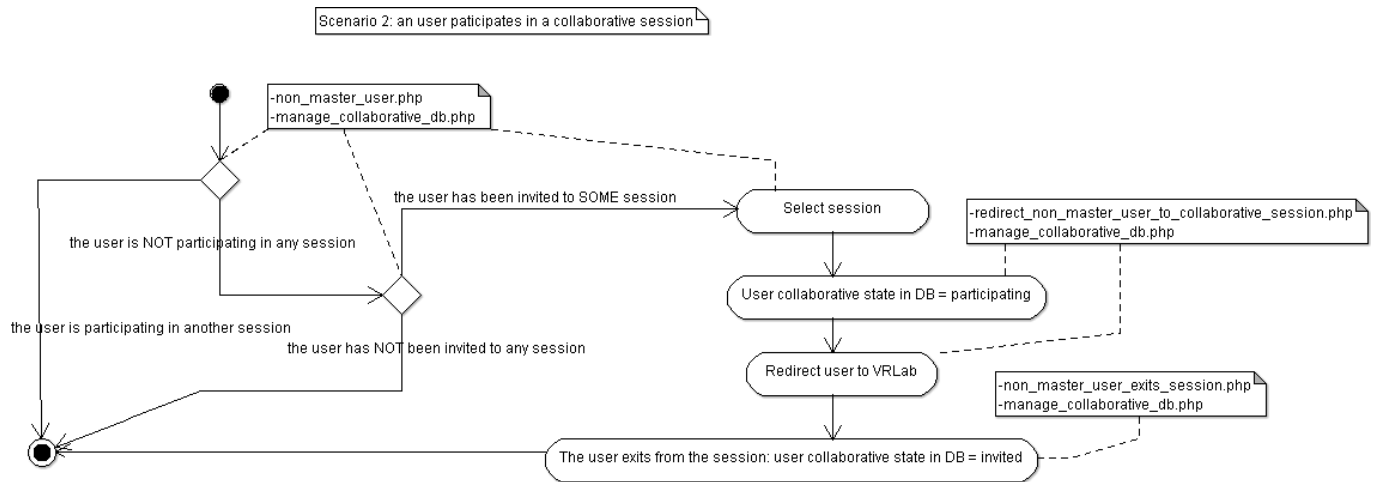


Figure 7: Activity diagram that depicts how the non-master user interacts with the system

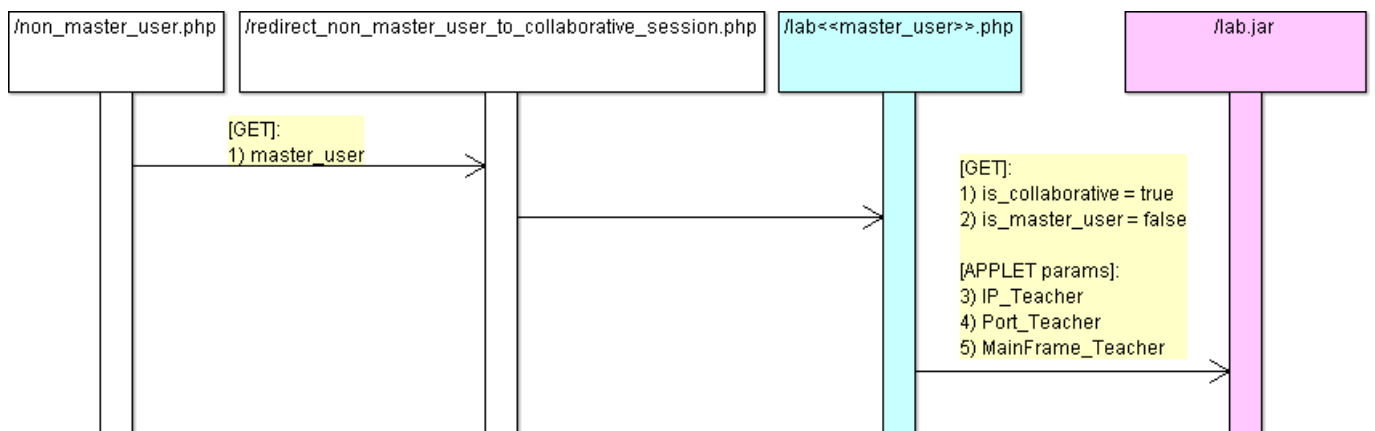


Figure 8: Sequence diagram that depicts how the non-master user interacts with the system

2 Entity-relationship model

Figure 9.



Figure 9: Entity-relationship model

3 Relational Model

There are 3 tables:

1. collaborative_sessions (id, port, vrlab, master_user)
2. collaborative_users (id, ip, collaborative_session_where_user_participates)
3. collaborative_invitations (id, invited_user, collaborative_session)

```

1 create table collaborative_sessions (
2     id serial not null,
3     port int not null,
4     vrlab varchar(1000) not null,
5     master_user int not null,
6     primary key(id),
7     foreign key (master_user) references collaborative_users(id)
8 )
9
10 create table collaborative_users (
11     id int not null,
12     ip varchar(100),
13     collaborative_session_where_user_participates int not null,
14     primary key(id),
15     foreign key(collaborative_session_where_user_participates)
16         references collaborative_sessions(id)
17 )
18
19 create table collaborative_invitations (
20     id serial not null,
21     invited_user int not null,
22     collaborative_session int not null,
23     primary key(id),
24     foreign key(invited_user) references collaborative_users(id),
25     foreign key(collaborative_session) references collaborative_sessions(id)
26 )

```

Figure 10: SQL code to create the tables

4 Proposals

4.1 Eliminating most input parameters for the applet

In Figure 6, the applet `lab.jar` should not pass any parameter to `createHtmlStudent.php` due to:

1. `createHtmlStudent.php` can get IP with the sentence `$ip=$_SERVER['REMOTE_ADDR'];`
2. PORT will always be 50000.
3. `PACKAGE`, `MAINFRAME` and `CODEBASE` are applet attributes. So, instead of being passed as parameters to `createHtmlStudent.php` (see Figure 6), they should be stored in a new table named `vrllabs` (see Figure 11); and the relational model would have a new table `vrllabs`:
 - (a) `collaborative_sessions` (`id`, `port`, `vrllab`, `master_user`)
 - (b) `collaborative_users` (`id`, `ip`, `collaborative_session_where_user_participates`)
 - (c) `collaborative_invitations` (`id`, `invited_user`, `collaborative_session`)
 - (d) `vrllabs` (`id`, `name`, `course`, `applet_name`, `class_file`, `codebase`, `mainframe`, `is_collaborative`)

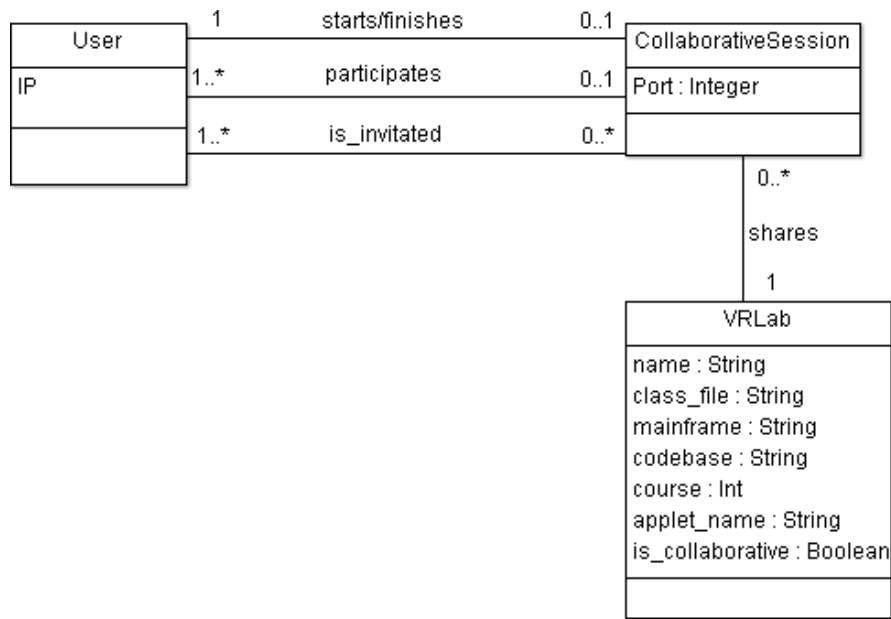


Figure 11: Entity-relationship model

4.2 Eliminating `createHtmlStudent`

Figures 5, 6 and 8 are replaced by Figures 12, 13 and 14 respectively.

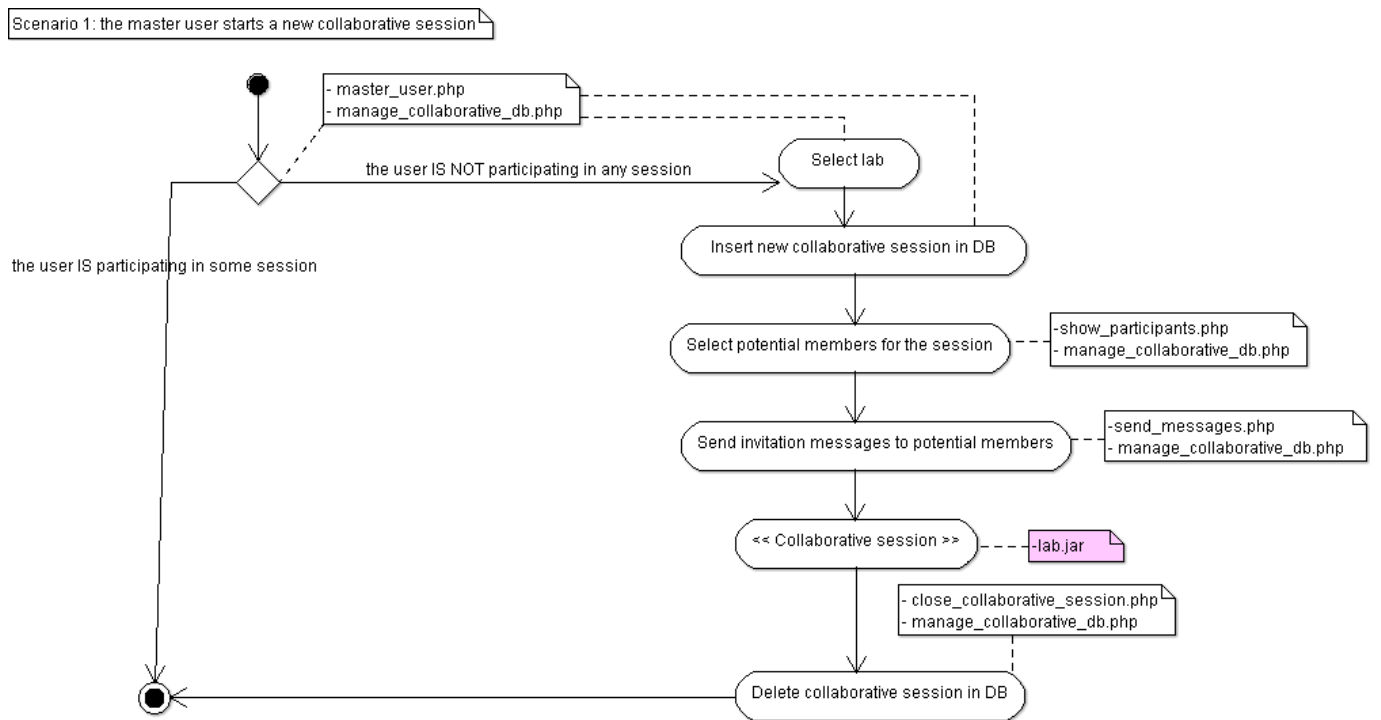


Figure 12: Activity diagram that depicts how the master user interacts with the system

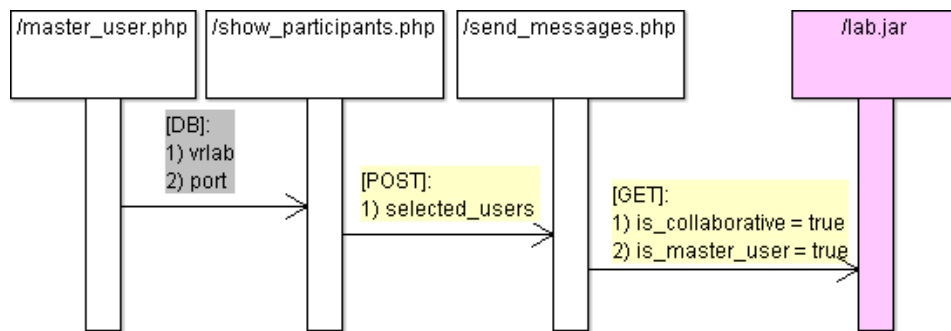


Figure 13: Sequence diagram that depicts how the master user interacts with the system

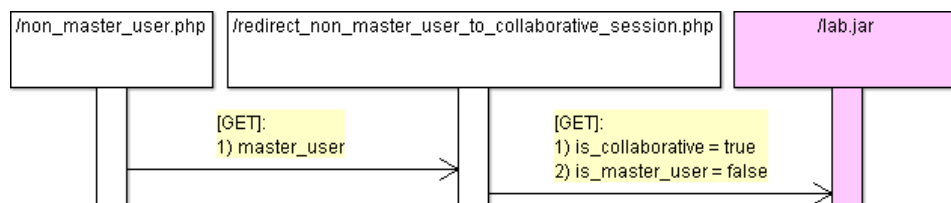


Figure 14: Sequence diagram that depicts how the non-master user interacts with the system