

vMoodle Mobile



Dr. Ming Zhao



Carlos Lacasa

Eduardo Castillo



Gerardo Guijarro

Claudia de Leon



Rachel Chavez

Dr. Peter Clarke
Prof. Tiana Solis



Copyrights © 2012 by

FIU Research Laboratory for Virtualized Infrastructure, Systems, and Applications (VISA)

All rights reserved. No part of this document shall be reproduced, stored in a retrieval system, or transmitted by any means, electronics, mechanical, photocopying, recording, or otherwise, without written permission from the authors. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and authors assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

Trademark

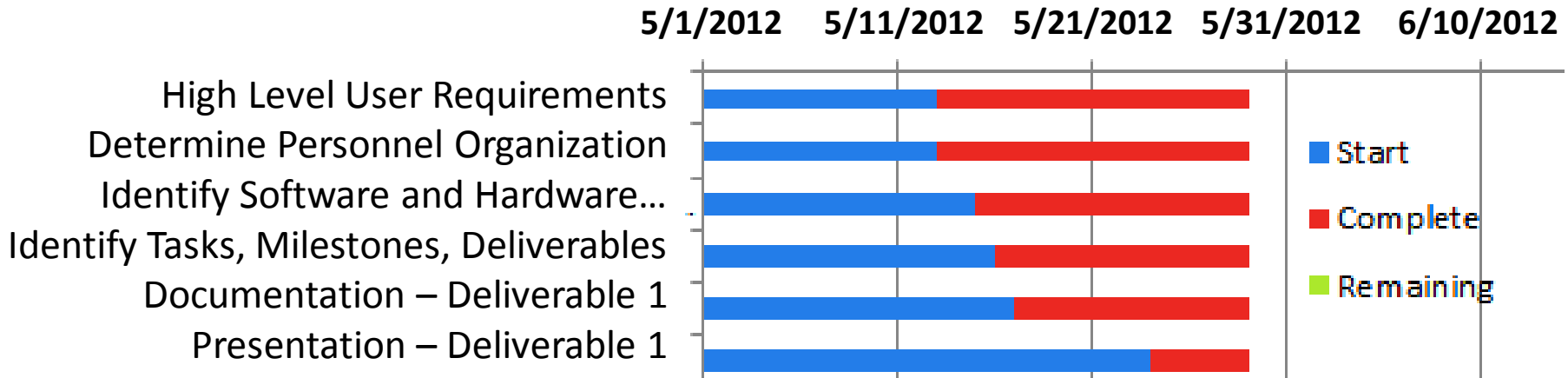
All trademarks, trade names, or logos mentioned or used are the property of their respective owners. Every effort has been made to properly capitalize, punctuate, identify, and attribute trademarks and trade names to their respective owners.

Team Roles

Name	Phase 1	Phase 2	Phase 3	Phase 4
Rachel Chavez	Leader - Business Analyst	Time Keeper - System Arch.	Software Dev.	Minute Taker - Tester
Eduardo Castillo	Project Analyst	Minute Taker - System Analyst	Software Engineer	Leader - Tester
Claudia De Leon	Minute Taker - Functional Analyst	Requirement Validation	Leader - GUI Designer	Time Keeper - Quality Assurance
Carlos Lacasa	Time Keeper - Solutions Arch.	Leader - Testing Analyst	Minute Taker - Dev. Manager	Software Dev. - Tester
Gerardo Guijarro	Software Dev.	Software Analyst	Time Keeper - Quality Assurance	Software Dev. - Tester

Project Management

Deliverable 1

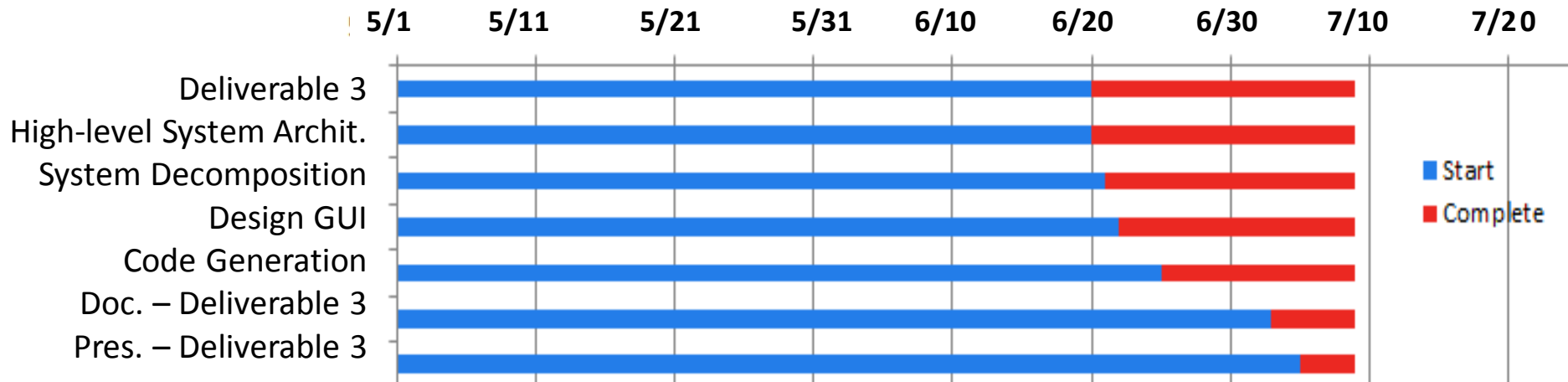


Deliverable 2

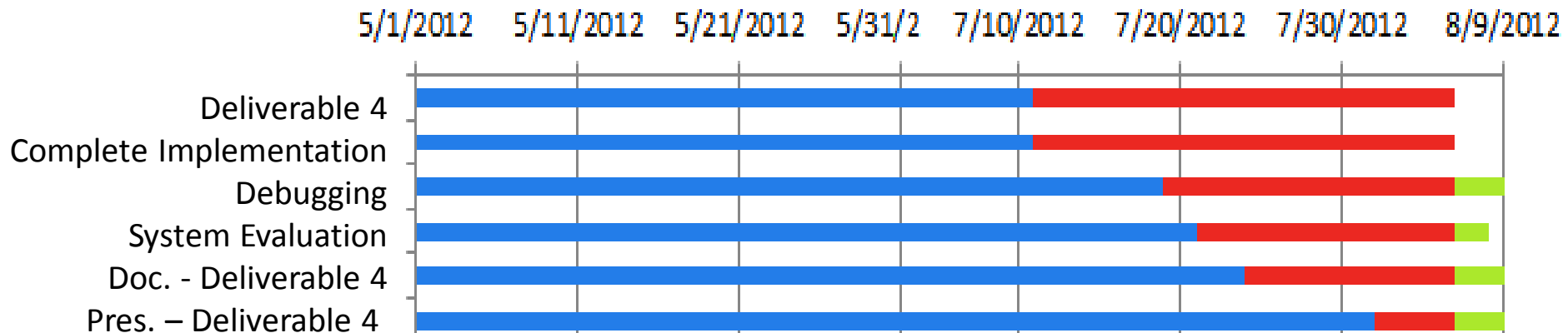


Project Management

Deliverable 3



Deliverable 4



Problem Def. & Limitations

Moodle

- Personalized educational experience
- Access to course materials

vMoodle

- Virtual Machines as valuable tools for portable educational modules
- Easily replicated and deployed

vMoodle Mobile

- Fast data access
- Mobility
- Increased popularity

vMoodle Limitation

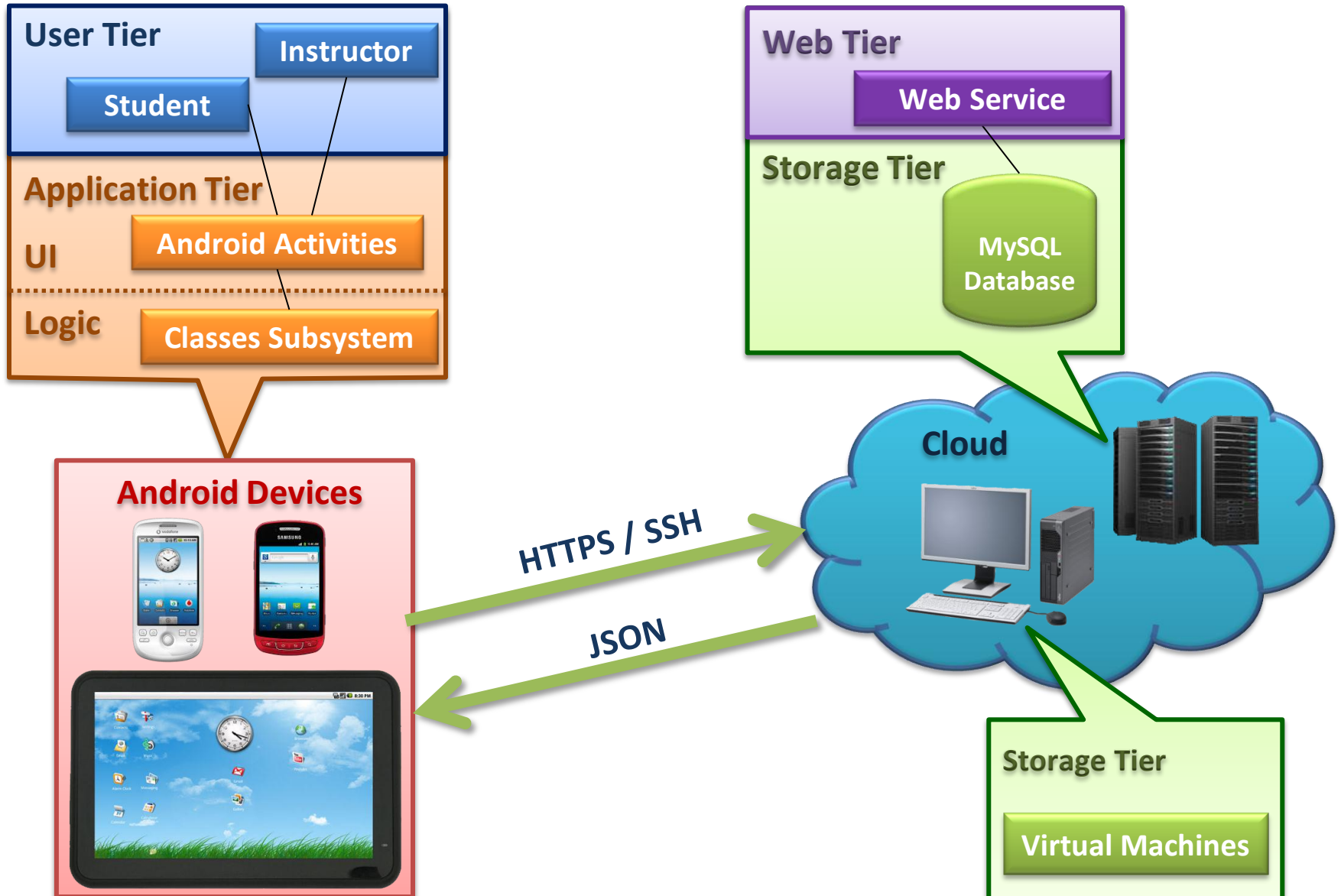
- Does not take advantage of the increasing popularity of mobile devices therefore, it *restricts users to use PCs.*

High Level System Objectives

The system shall:

- Provide the **similar functionality to vMoodle**
- Provide **fast** access to data
- Be **responsive**
- Make **efficient** use of the device screen.
- Have **intuitive UI** and self-explanatory UI elements.
- **Provide error messages** when necessary.

System Design



System Deployment

Hardware:

800MHz single core processor
196MB of RAM
3.5-inch HVGA touchscreen display
3G or Wi-Fi

Software:

Android 2.3 Gingerbread

Hardware:

5 GB of disk space
1 GB of RAM
2 GHz processor

Software:

Linux: Ubuntu 8.1

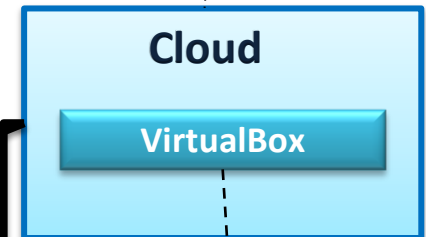
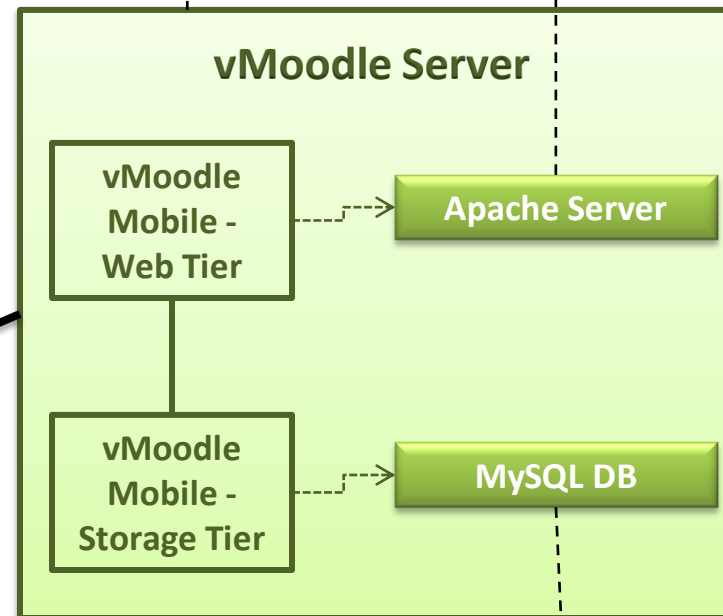
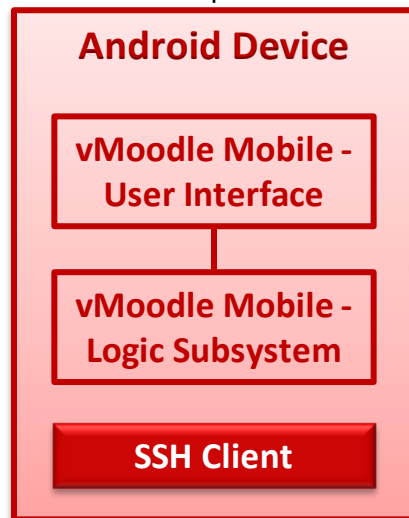
Apache 2.2.21 (Ubuntu)
PHP 5.3.8

Hardware:

1 TB of disk space
4 GB of RAM
2 GHz processor

Software:

Linux: Ubuntu 8.1



Network Speed: 446 kbps

Version: MySQL 5.5.16

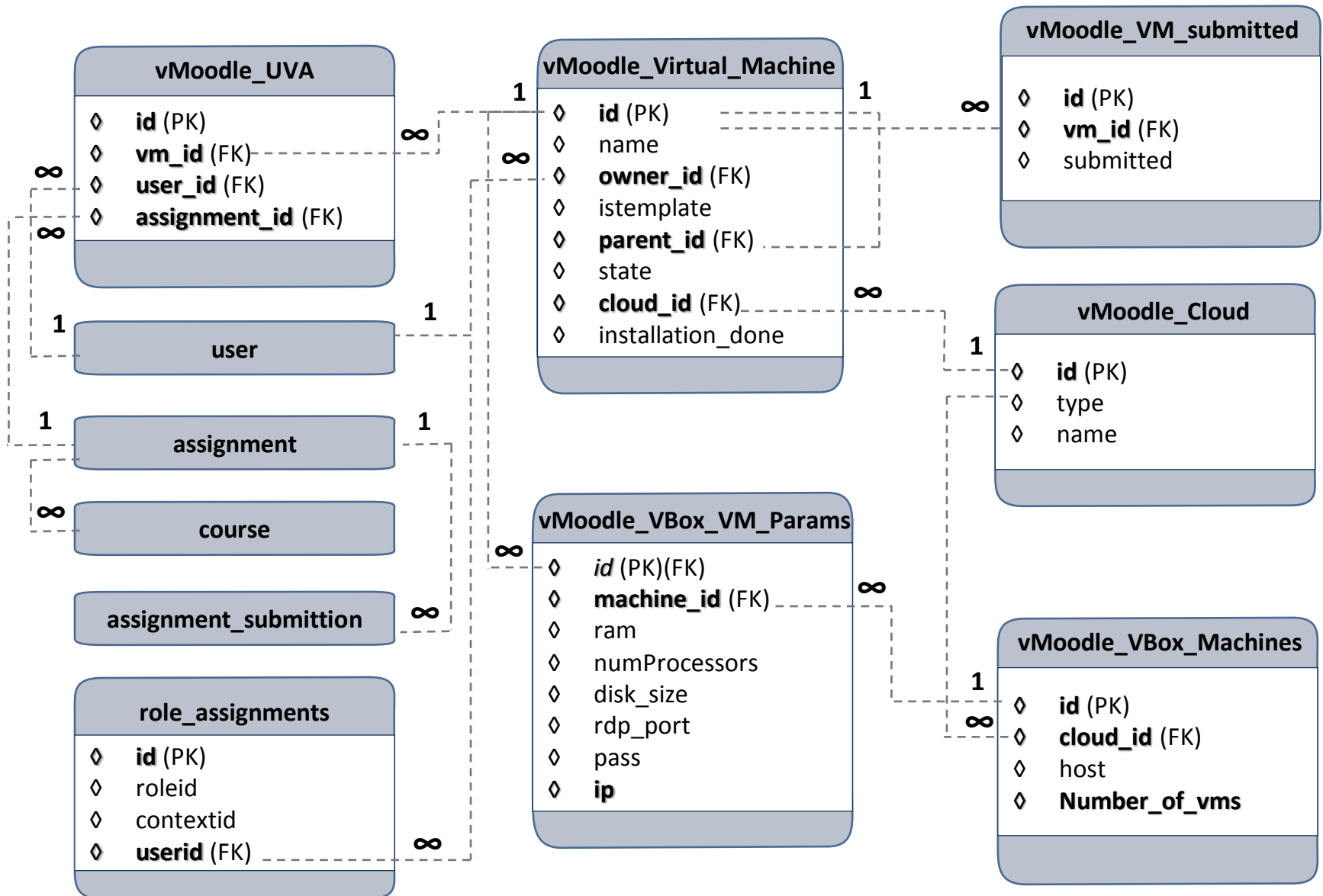
VirtualBox for Linux version : 4.1.18

JSON

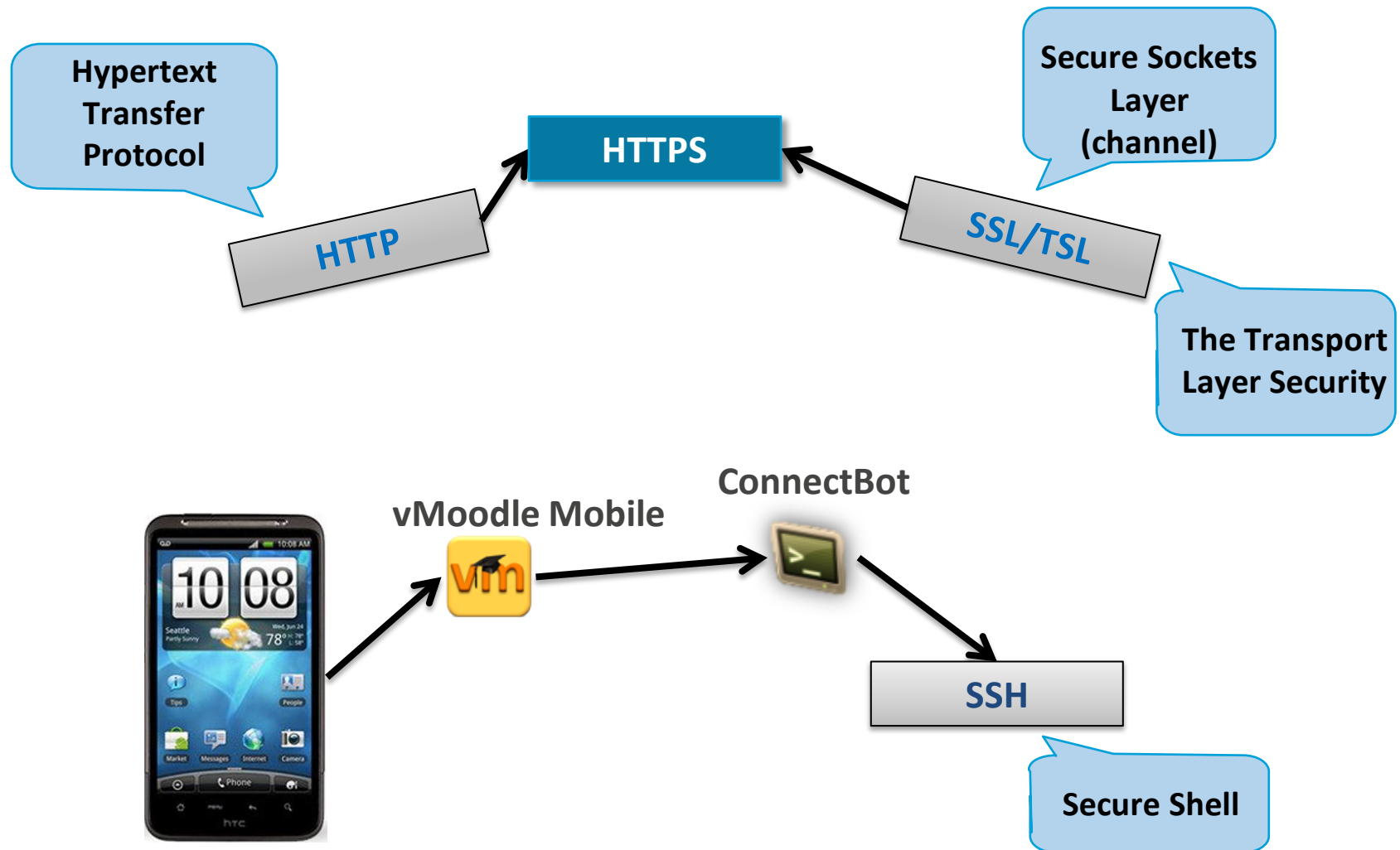
SSH

SSH

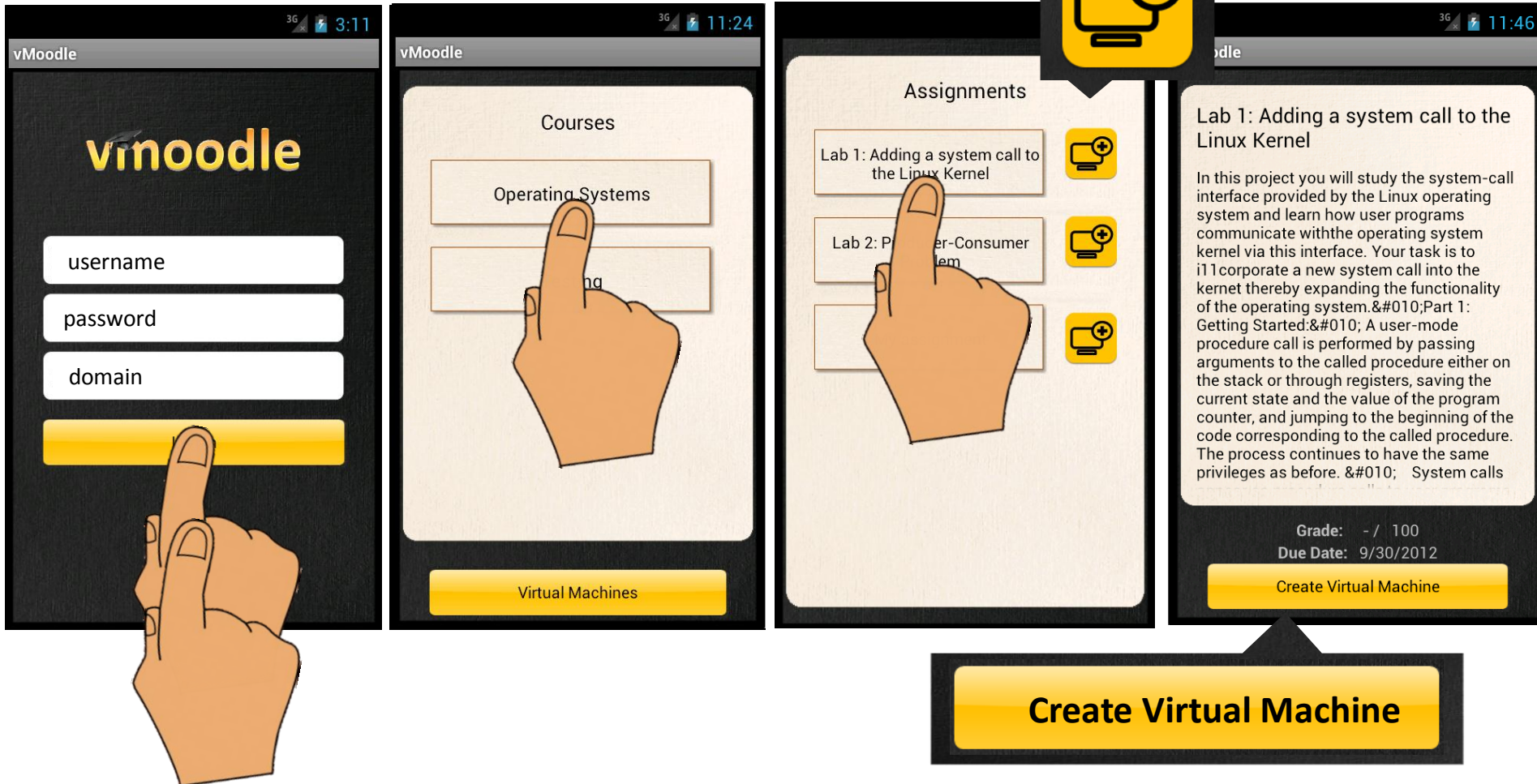
Data Management



Security & Privacy



System Design (Student Path)



System Design (Student Path)



Assignments

Lab 1: Adding a system call to the Linux Kernel



Lab 2: Producer-Consumer

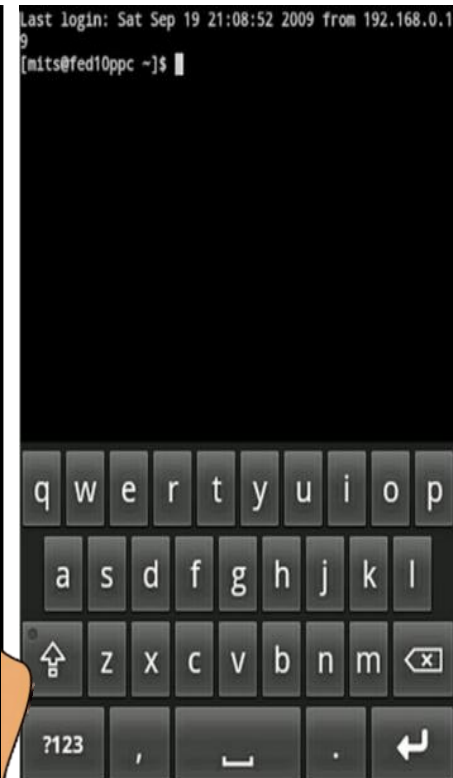
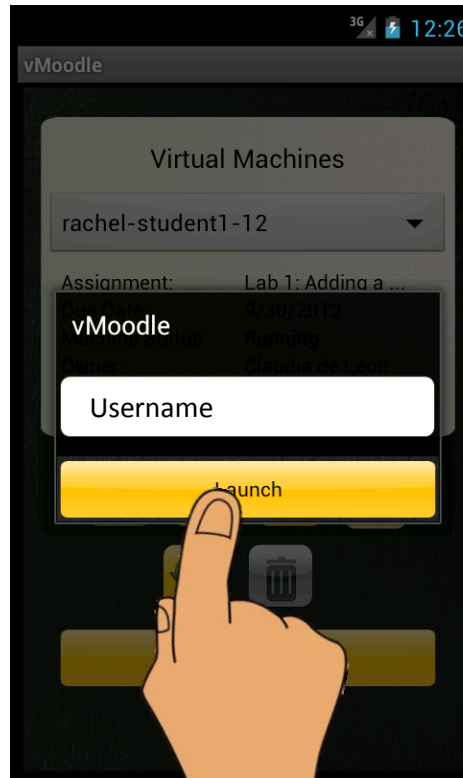


Lab 1: Adding a system call to the Linux Kernel

In this project you will study the system-call interface provided by the Linux operating system and learn how user programs communicate with the operating system kernel via this interface. Your task is to incorporate a new system call into the kernel thereby expanding the functionality of the operating system. Part 1: Getting Started: A user-mode procedure call is performed by passing arguments to the called procedure either on the stack or through registers, saving the current state and the value of the program counter, and jumping to the beginning of the code corresponding to the called procedure. The process continues to have the same privileges as before. System calls

Grade: 89 / 100
Due Date: 9/30/2012

View Virtual Machine



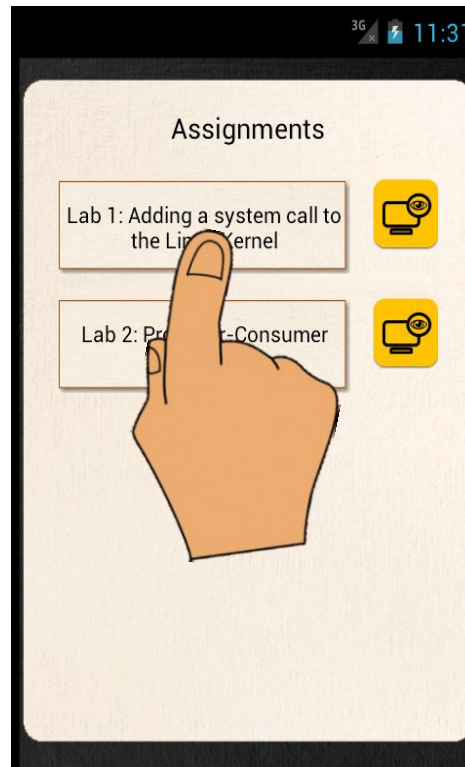
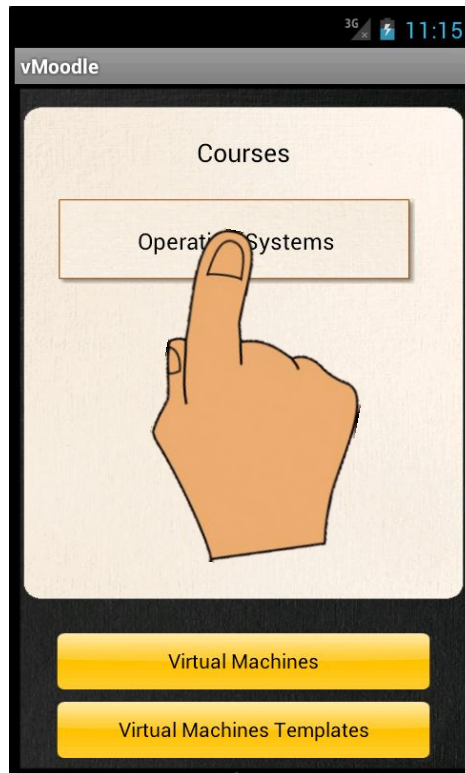
View Virtual Machine

Login / Courses / Assignments

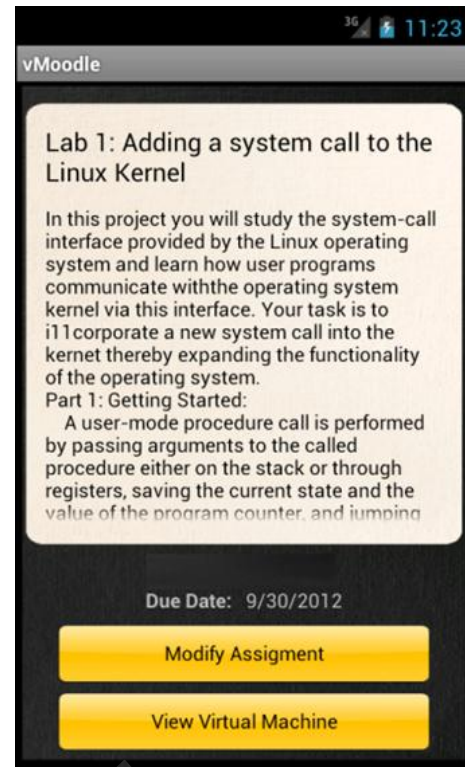
System Design (Instructor Path)



Virtual Machine Templates



Modify Assignment



System Design (Instructor Path)

vMoodle 3G 11:23

Lab 1: Adding a system call to the Linux Kernel

In this project you will study the system-call interface provided by the Linux operating system and learn how user programs communicate with the operating system kernel via this interface. Your task is to incorporate a new system call into the kernel thereby expanding the functionality of the operating system.

Part 1: Getting Started:

A user-mode procedure call is performed by passing arguments to the called procedure either on the stack or through registers, saving the current state and the value of the program counter, and jumping

Due Date: 9/30/2012

Modify Assignment

View Machine

vMoodle 4:22 PM

Lab 1: Adding a system call to the Linux Kernel

In this project you will study the system-call interface provided by the Linux operating system and learn how user programs communicate with the operating system kernel via this interface. Your task is to incorporate a new system call into the kernel thereby expanding the functionality of the operating system.

Part 1: Getting Started:

A user-mode procedure call is

Points: 100

Date: October 01, 2012

Template: template

Update Assignment

vMoodle 3G 12:03

Virtual Machines

student2

Assignment: Lab 1: Adding a ...

Due Date: 9/30/2012

Machine Status: **Stopped**

Owner: student2

Grade: 101 / 101

Play, Pause, Stop, Full Screen, Refresh, Delete, Settings

vMoodle 3G 12:05

Settings

Machine name: student2

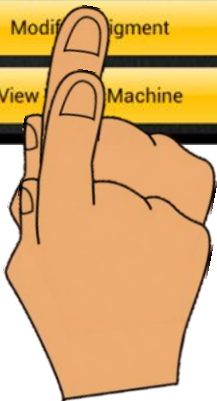
Processors (CPU): 1

System RAM (MB): 1024

Disk size (GB): 10

Save

Cancel



System Implementation

Logic

```
15 private JSONObject parserObj;
16
17 public JSONParser(String response)
18 {
19     try
20     {
21         parserObj = new JSONObject(response);
22     }
23     catch (JSONException e)
24     {
25         e.printStackTrace();
26     }
27 }
28
29 public String getUserId()
30 {
31     return parserObj.optString("user_id");
32 }
33
34 public boolean isValid()
35 {
36     if(this.parserObj == null)
37         return false;
38
39     if(parserObj.optString("isValid").equalsIgnoreCase("TRUE"))
40     {
41         return true;
42     }
43     return false;
44 }
```

Web Service Client

- URL
- Service



JSON Parser

- isValid
- GetUserRoleAndCourses
- GetAssignments
- GetListOfVMs

System Implementation

Web

Web Service

- login
- refresh
- createAssi
- gnment
- submitAss
- ignment
- createVM
- refreshVM
- startVM
- stopVM
- pauseVM
- getResour
- ces
- modifyVM
- deleteVM

Storage



- Installed on Visa lab private cluster
- Allows additional guest operating systems

```
1 function getIP($vm_name, $ssh)
2 {
3     $vm_ip = "";
4     $counter = 20;
5     while($counter > 0)
6     {
7         $vm_ip = nl2br($ssh->exec("VBoxManage guestproperty get " . $vm_name .
8         " \"/VirtualBox/GuestInfo/Net/0/V4/IP\" | awk '{ print $2 }'"));
9         if($vm_ip && !preg_match ("/VALUE/i", $vm_ip))
10        {
11            return substr($vm_ip, 0, -7);
12        }
13        sleep (5);
14        $counter--;
15    }
16    return false;
17 }
```

- MySQL(RDBMS)



System Evaluation

Experiment Testbed

Hardware

Device	RAM	CPU	Hard Disk	Network Speed	Screen Size
Cellphone	512 MiB	800 MHz	125 MB	446 kbps	3.5 inch
PC	4 GB	2 x 2.2 GHz	320 GB	11 Mps Wi-Fi (fiu-scs)	N/A

Device

OS

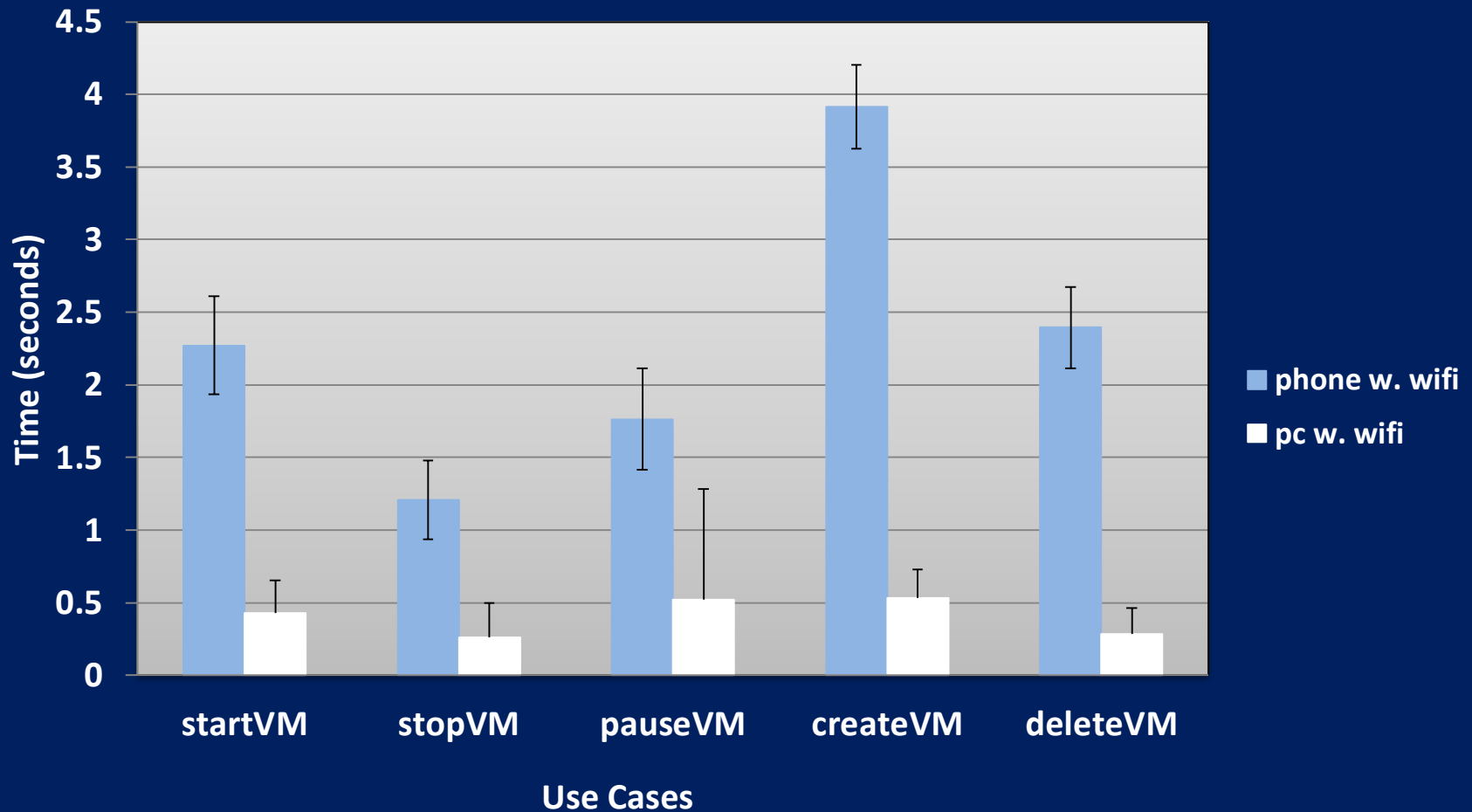
Additional Soft.

Software

Cellphone	Android 2.3 Gingerbread	ConnectBot
PC	Windows 7 Professional	N/A

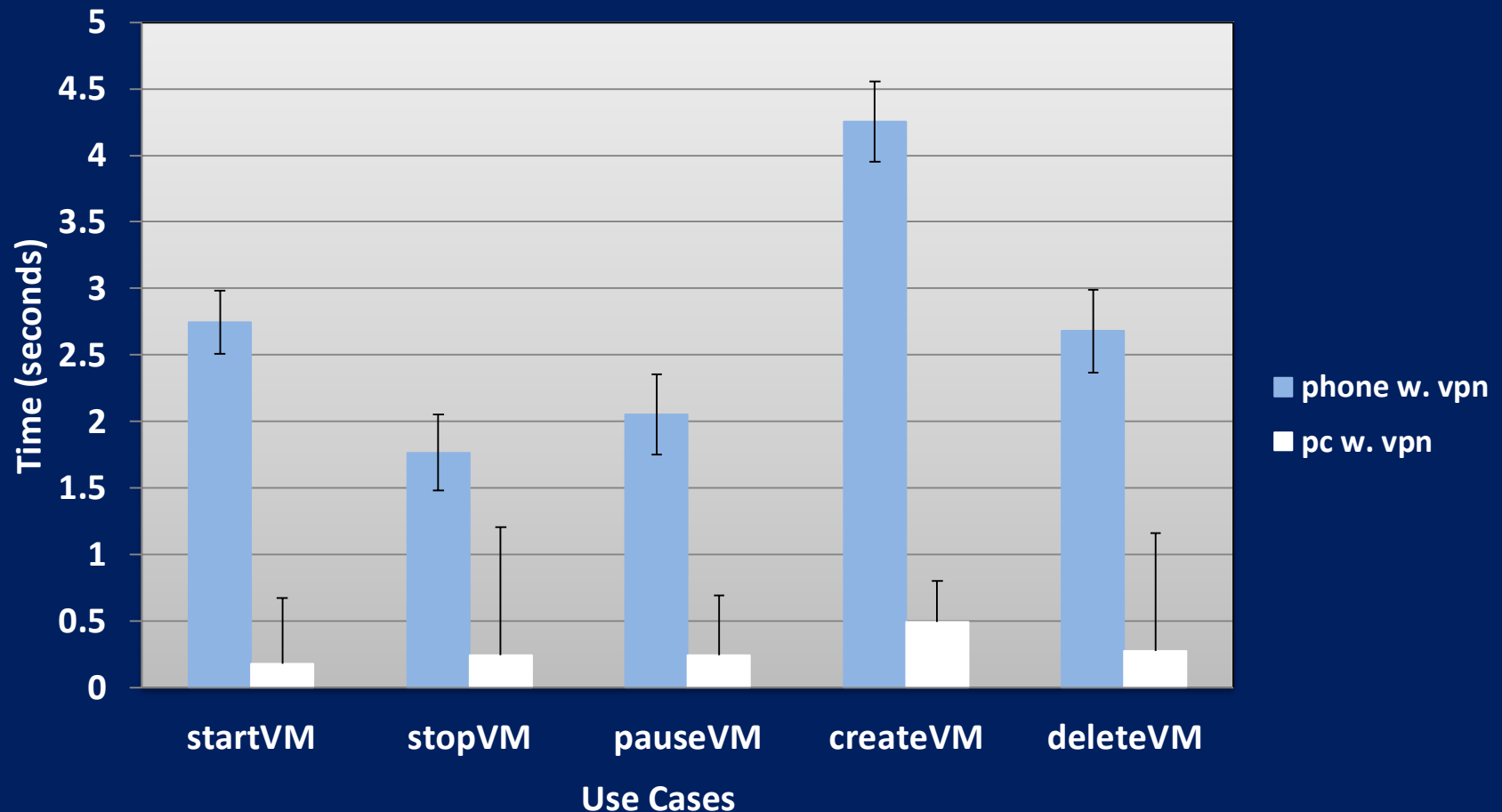
System Evaluation

Use Cases Performance Using Wi-Fi

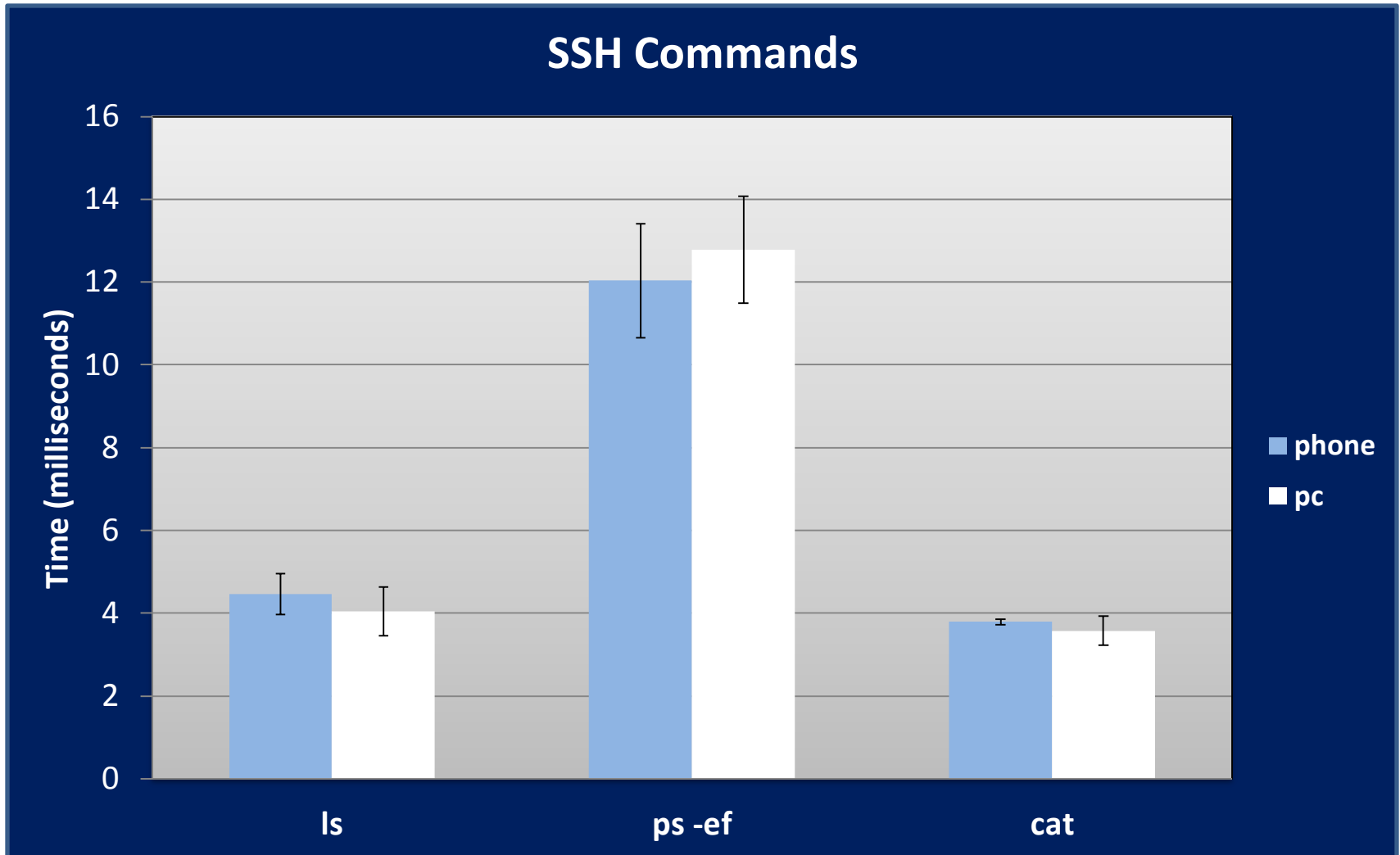


System Evaluation

Use Cases Performance Using VPN



System Evaluation



System Evaluation

Additional Performance Statistics

Action	Average Time
startNewVM	1.5 min.
Transition between activities	0.0592 sec.
SSH commands	2 millisec.

Use Cases Demo

Student Create VM

1. Login
2. Select a course from the list
3. Select assignment from the list
4. Tap Create Virtual Machine

Student View VM

Student is on the Courses view

- 1. Tap Virtual Machine button**
- 2. Select the machine from the list**
- 3. Tap the View button**
- 4. Enter username**
- 5. Enter password**

Instructor Edit Settings

- 1. Login**
- 2. Tap Virtual Machine Templates
button**
- 3. Tap on the Settings button.**
- 4. Adjust machine's settings**
- 5. Save**

Instructor Delete VM

Instructor is on the Courses view

- 1. Tap Virtual Machine button**
- 2. Select machine to delete from list**
- 3. Press Delete button**
- 4. Confirm action**

Performance Evaluation Demo

Start Virtual Machine

User is on VM Controls screen

1. Tap start VM
2. Press stop VM
3. Repeat 3 times
4. Open file 'startVMDemo.txt' located on SD card

System SSH Commands

User is on VM Controls screen

1. Tap view VM
2. Enter username
3. Enter password
4. Type 'date +%N;(any cmd) ;date +%N'
5. Subtract to last from first to get execution time.

Questions?