SAFE Development & Scalability Testing
M. Tech. Project Phase I

Ayush Singh
163050014

Guide: Prof. Kameswari Chebrolu
Computer Science & Engineering
Indian Institute of Technology, Bombay
Outline

1 Introduction

2 Development

3 Testing
Introduction

This Project is broadly divided into two parts:

- **Development**: Feature development of SAFE V2 android app.
- **Testing**: Setup & Planning for scalability testing of Bodhitree and SAFE.
**Problem:** Access point causing bottleneck & "No Network Connection" error in app.
## Development

**Figure:** Wireshark Trace before
Development

Design & Implementation

- OKHttp library is being used for all its network calls.
- To make HTTP request and reading their response an instance of OKHttpClient is required.
- OKHttpClient is instantiated in singleton pattern.
Figure: Wireshark Trace after
Development

Problem: Implement a quiz navigation screen on android side.

Design

- Color code assigned to each question.
- Marks for each question will be displayed.
- Some part of the question can be viewed.
- Tag is used to easily identify questions.
Development

Navigate Quiz

Q. 1  What is the height of tree?
    #course #topic #type
    1.0 Marks

Q. 2  What are the prime numbers?
    #course #topic #type
    1.0 Marks

Q. 3  Who discovered tree
    #course #topic #type
    1.0 Marks

Q. 4  Trees are found near
    #course #topic #type
    1.0 Marks

Q. 5  How many leaves a tree can
    #course #topic #type
    1.0 Marks

Q. 6  B stands for
    #course #topic #type
    Attempted Marked for Review

Figure: Quiz Navigation Activity
**Problem:** Implement an option to download the quiz.

**Design**

- Get Quiz ID from Instructor.
- Download the quiz with its unique Quiz ID.
- Quiz will be encrypted compressed file.
- Can only be decrypted by entering pass-code.
- Saved in internal device memory cannot be detected to other apps.
**Development**

**Problem:** Implement image pass-code to validate users for quiz.

**Design**

- User will be given a set of images.
- Select images in correct order to be get access to the quiz.
- Instructor will display the pass-code on screen.
Development

**Figure:** PassCode Activity
Development

**Problem:** Provide facility for user to change password for their account.

**Design**

- Login with current credentials
- Get Authenticated by sever.
- Create new password.
- Password’s minimum length 6 character.
- User have to enter new password two times.
**Problem:** List of previously submitted quizzes can become quite large.

**Design**
- List of enrolled courses is displayed.
- Selection of a course.
- Displaying previous submissions of single course.
- Use of Recycler View for showing both list.
- Use of single layout file for multiple activity.
### Development

#### (a) Course List

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
<th>Marks</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Graphics Basics</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Machine Learning Foundation only</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Introduction to Computer Science Basic</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### (b) Submissions of a Course

<table>
<thead>
<tr>
<th>Course</th>
<th>Submission Type</th>
<th>Date</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>BQMG1B7G</td>
<td>Binary Search Tree</td>
<td>1-8-2017</td>
<td>0.0</td>
</tr>
<tr>
<td>T47FLCMT</td>
<td>Binary Search Tree</td>
<td>12-6-2017</td>
<td>2.0</td>
</tr>
<tr>
<td>X1RA4D1M1</td>
<td>Binary Search Tree</td>
<td>12-6-2017</td>
<td>2.0</td>
</tr>
<tr>
<td>YTC2BS5M</td>
<td>Binary Search Tree</td>
<td>12-6-2017</td>
<td>0.0</td>
</tr>
<tr>
<td>1AN4WMKC</td>
<td>Binary Search Tree</td>
<td>12-6-2017</td>
<td>NA</td>
</tr>
<tr>
<td>BH56W1C8</td>
<td>Binary Search Tree</td>
<td>12-6-2017</td>
<td>NA</td>
</tr>
<tr>
<td>6F8YP5E8</td>
<td>Binary Search Tree</td>
<td>12-6-2017</td>
<td>NA</td>
</tr>
</tbody>
</table>

**Figure:** Previous Submission Screens
Scalability Testing
How application scales as it is deployed on larger system or as more load is applied to current system.
Testing to be done for:

- Bodhitree
  - **Backend** Python based Django framework.
  - **Frontend** HTML, CSS, JQuery etc.

- SAFE
  - **Backend** Python based Django framework.
  - **Frontend** Android.
Server Setup

- Setup an instance of Bodhitree and Safe on different server.
- Testing server having same specification as that of original.
- Update of the testing server and upgrade of softwares for no compatibility issues to appear.
Nginx Configuration

- Nginx a reverse proxy server.
- All the request comes first to Nginx, it sits behind the firewall in server.
- Configured the nginx to send incoming request to correct backend server.
- Configured the nginx logging to our requirements.
**Setup**

**Database Setup**

- Created dummy courses for the sites.
- Populated it with essentials contents which were required during testing.
- Created some instructor account and 1000 student account.
- Enrolled students account in course.
Designing Test Plan

BlazeMeter

- Tool that can be integrated with browser to create test plan.
- Easily record plan by visiting web pages.
- Extra component also recorded like javascript, JSON etc. creating accurate test plan.
- Export test to JMeter for extra tweaking.
Designing Test Plan

**JMeter**

- Open source java application for performance measurement.
- Simulating multiple users by sending multiple request.
- Configure the test plan for managing cookies and session.
- Collects responses to calculate various server parameters such as throughput and CPU utilization.
Designing Test Plan

Figure: JMeter test plan for video page
Designing Test Plan

Test Plan

- Check the current throughput of server when video is being played.
- Plan consists of:
  - Login to Bodhitree.
  - Navigating to a video in course.
  - Playing the video.
Running Test

**Figure:** Request Made to the server
Running Test

Figure: Response from server
Thank You...