UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Written Examination

DIT341 – Web and Mobile Development

Wednesday, April 24th, 2018, 08:30 - 12:30

Examiner: Philipp Leitner

Course Responsible: Grischa Liebel

Contact Persons During Exam:
Philipp Leitner (+46733056914)
Joel Scheuner (+46 733 42 41 26)

Allowed Aides:
None except English dictionary (non-electronic), pen/pencil, ruler, and eraser.

Results:
Exam results will be made available no later than 15 working days after the exam date through Ladok.

Overall Points: 50

Grade Limits: 0 – 24 points: **U**, 25 – 42 points: **G**, >42 points: **VG**

Review:
The exam review will take place latest three weeks after the exam results have been published in Ladok. It will be announced on GUL at least one week in advance.

Part 1 – Concept Definitions & Explanations (17P)

Q1.1: What is the DOM? Describe briefly and give an example. **(3P)**

Q1.2: Describe briefly how HTTP requests and responses are structured, and provide a valid example for a (successful) HTTP exchange. **(5P)**

Q1.3: Name and briefly explain the 5 central properties ("constraints") of REST. **(5P)**

Q1.4: What possibilities do we have to create lists in HTML? Name the tags, describe the difference, and provide a complete example for each. **(2P)**

Q1.5: What is AJAX? What do we need AJAX for? **(2P)**

Q1.6: Explain briefly the difference between processes and threads. **(2P)**

Part 2 – Working with code (19P)

HTML and CSS

Q2.1: Write a minimal HTML document including proper document type information, a page title called `MyPage`, a top-level header called `PageTitle` with the class attribute `class1`, and a paragraph with the content `Text1` with the id attribute `para1`. **(3P)**.

Q2.2: Figure 1 depicts a CSS style definition for the minimal document from Q2.1. Describe in which color the texts `PageTitle` and `Text1` are rendered and explain why. **(2P)**

```
1  <style>
2    .class1 {
3      color: blue;
4    }
5    #class1 {
6      color: yellow;
7    }
8    .para1 {
9      color: red;
10   }
11   #para1 {
12     color: green;
13   }
14 </style>
```

Figure 1: CSS Styles

JavaScript

Q2.3: Figure 2 depicts a part of a JavaScript conditional. What is the console output of this program and explain why this is the case? **(1P)**

```
1 var x = "7";
2 var y = 7;
3 if (x == y) {
4     console.log("True");
5 } else {
6     console.log("False");
7 }
```

Figure 2: JavaScript Code

Q2.4: Name and explain one advantage and one disadvantage of callbacks. **(2P)**

Q2.5: Figure 3 depicts a short JavaScript program. Describe what the console output of the program is and explain why this is the case. **(3P)**

```
1  function fun1() {
2      fun2();
3      console.log("1");
4      setTimeout(function() {
5          console.log("2");
6      }, 0);
7      console.log("3");
8  }
9  function fun2() {
10     console.log("4");
11     setTimeout(function() {
12         console.log("5");
13     }, 0);
14     console.log("6");
15 }
16 function fun3(cb) {
17     console.log("7");
18     cb();
19     console.log("8");
20 }
21
22 fun1();
23 fun3(function() { console.log("9"); });
24 console.log("10")
```

Figure 3: Short JavaScript Program

Frontend Development

Q2.6: Vue.js supports single file components with filenames ending in `.vue`. Name and explain one advantage and one disadvantage of using single file components. **(2P)**

Q2.7: Describe the functionality of the Vue.js Application depicted in Figure 4. What value is displayed before and after pressing the ”ClickMe” button 1, 2, and 3 times? **(2P)**

```
1  <body>
2      <div id="app">
3          <my-comp></my-comp>
4      </div>
5
6      <script>
7          Vue.component('my-comp', {
8          data: function () {
9              return {
10             n1: 3,
11             n2: 2
12         }},
13         template: `<div><button v-on:click="n2++">ClickMe</
               button><br/><p>{{fun1}}</p></div>`,
14         computed: {
15             fun1: function () {
16               return this.n1 + this.n2 * 2;
17             }
18         }
19         });
20
21         var app = new Vue({
22           el: '#app'
23         });
24     </script>
25 </body>
```

Figure 4: Part of a Simple Vue.js Application with a Component

Android Development

Q2.8: Give an example what is executed as a process and what is executed as a thread in the context of Android. **(1P)**

Q2.9: What is the UI thread in an Android application? What is the effect of blocking the UI thread? **(1P)**

Q2.10: Figure 5 shows a method using runOnUiThread() in an Android application. What does the method `runOnUiThread()` do? Describe in which context it is necessary to use this method and give an example. **(2P)**

```
1  private void publishProgress(int randNum) {
2    final String text = randNum.toString();
3
4    runOnUiThread(new Runnable() {
5      @Override
6      public void run() {
7        mainFragment.getResultsTextView().setText(text);
8      }
9    });
10 }
```

Figure 5: Method using runOnUiThread() in an Android Application

Part 3 – REST API Case (8P)

Wordpress is a popular open-source blogging software. Using Wordpress, users can write blog posts, comment on blog posts, and maintain user profiles. In Figure 6, you see an excerpt of possible HTTP requests for a simplified version of the Wordpress REST API.

```
GET /wp/users
GET /wp/users/:user
GET /wp/posts?search=:keyword&page=:page&limit=:limit

PATCH /wp/users/:user/:post
PUT /wp/users/:user/:post
```

Figure 6: Excerpt of Simplified Wordpress REST API

Q3.1: Assuming that the API has been designed following the REST principles introduced in this course, describe what content and status code you would expect the following HTTP request to return in case of a successful execution. What status code would you expect if the user with the provided ID cannot be found? **(3P)**

- GET /wp/users/:user

Q3.2: Assuming that the API has been designed following the REST principles introduced in this course, contrast the following two requests. What is your expectation of how their functionality differs? In which cases would you prefer one over the other? **(2P)**

- PATCH /wp/users/:user/:post
- PUT /wp/users/:user/:post

Q3.3: Investigate the following operation. What do you expect this method to do? What meaning do you think do the parameters “page” and “limit” have? **(2P)**

- GET /wp/posts?search=:keyword&page=:page&limit=:limit

Q3.4: What do you expect to happen if you send the exact same request *twice* to the following endpoint, immediately after each other? **(1P)**

- PUT /wp/users/:user/:post

Part 4 – Reflection (6P)

Q4.1: In one of the guest lectures, we learned about the concept of Continuous Integration (CI). What are the promises of CI (i.e., why do people do it)? How does CI typically work in practice? **(4P)**

Q4.2: Contrast the "smoke testing", "canary testing", and "monkey testing" test strategies that we have encountered in the context of testing Android applications. Are those useful only for mobile apps? **(2P)**