



UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Written Examination

DIT341 – Web and Mobile Development

Monday, October 29th, 2018, 8:30 - 12:30

Examiner: Philipp Leitner

Course Responsible: Grischa Liebel

Contact Persons During Exam:

Grischa Liebel (+46 31 772 56 46)

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)

General Terminology

Q1.1: What does the term *fullstack development* describe? Explain the term. (1P)

Q1.2: Web applications should nowadays be implemented in a *responsive* way. What does it mean to do *responsive web design*? Explain the term and give an example. (1.5P)

Q1.3: Give an example of a Uniform Resource Locator (URL) and point out which parts of the URL make up the port, the protocol, the path, and the host name. (2P)

Q1.4: What is the DOM? Explain the term and describe how JavaScript can interact with the DOM. (1.5P)

JavaScript

Q1.5: In JavaScript, variables can be compared using the `==` operator and the `===` operator. Explain the difference between these two operators. (1P)

Q1.6: *AJAX* is an essential part of creating modern web applications with JavaScript. Describe what the term stands for and for which purpose *AJAX* is used. (1.5P)

RESTful Systems

Q1.7: A RESTful system follows a number of constraints. One of them is the *stateless* constraint. Explain what it means that a REST system is *stateless*. (1P)

Frontend Development

Q1.8: Several frontend (web) development frameworks use *two-way data binding*. Explain the term and give an example of a use-case for two-way data binding. (1.5P)

Q1.9: Vue.js knows the concept of *computed properties*. Describe what a computed property is and how it is different to a function/method. (1.5P)

Continuous Integration

Q1.10: When using Continuous Integration (CI), each commit/push to a repository is automatically built and tested. Describe (up to 3) potential benefits of this practice and explain how/why CI leads to these benefits. **(3P)**

Android Development

Q1.11: Figure 1 depicts part of the lifecycle of an Android Activity. Give one example each for cases in which *onPause()*, *onStop()* and *onResume()* are executed. **(1.5P)**

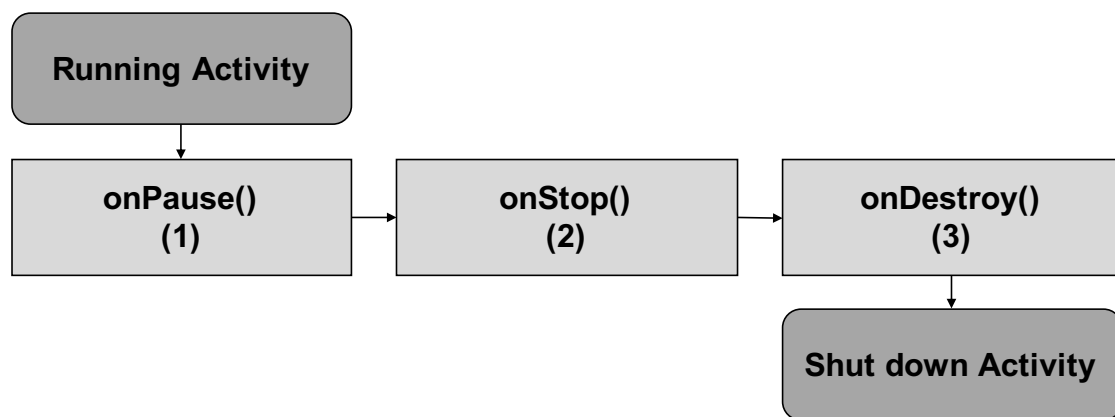


Figure 1: Excerpt of the Android Activity Lifecycle

Part 2 – Working with code (19P)

HTML and CSS

Q2.1: Figure 2 depicts an HTML page. Explain the purpose of each of the attributes in lines 6 and 7. **(2P)**

```
1  <!doctype html>
2  <html>
3      <head></head>
4      <body>
5          <h1>Hello World!</h1>
6          
7          <a href="next.html" target="_blank">Next</a>
8      </body>
9  </html>
```

Figure 2: A Simple HTML Page

Q2.2: Figure 3 depicts an HTML web page with a CSS style definition. Describe in which color the text in lines 17-20 is rendered and why. **(3P)**

```
1 <!doctype html>
2 <html>
3   <head>
4     <style>
5       p {
6         color: red;
7       }
8       .someClass {
9         color: blue;
10      }
11      #someID {
12        color: green;
13      }
14    </style>
15  </head>
16  <body>
17    <p style="color: black;">Text</p>
18    <p style="color: black;" class="someClass">Text</p>
19    <p id="someID" class="someClass">Text</p>
20    <p class="someClass">Text</p>
21  </body>
22 </html>
```

Figure 3: Several HTML Tags

JavaScript

Q2.3: Figure 4 depicts part of a JavaScript program. Describe what the program outputs in line 1, 3, and 6. **(1.5P)**

```
1 console.log(x);  
2 x = "text1";  
3 console.log(x);  
4 var x;  
5 x = "text2";  
6 console.log(x);
```

Figure 4: JavaScript Code

Q2.4: Which JavaScript behaviour is depicted in Figure 4? Name and describe the behaviour. **(1.5P)**

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

```
1 function func1(param) {  
2   setTimeout(function() {  
3     console.log("Foo");  
4   }, 0);  
5   console.log("Bar");  
6   console.log("Hello");  
7   param();  
8 }  
9  
10 function func2() {  
11   console.log("World");  
12 }  
13  
14 func1(func2);  
15 console.log("!");
```

Figure 5: Short JavaScript Program

Frontend Development

Q2.6: Figure 6 depicts a Vue.js program that makes use of components. Describe what a component is and what main benefit components offer. (2P)

Q2.7: Describe the functionality of the Vue.js Application depicted in Figure 6. (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            count: 12
11          }
12        },
13        template: `<button v-on:click="count++" v-bind:style="{
14                      fontSize: count + 'px' }">Click</button>`
15      });
16
17      var app = new Vue({
18        el: '#app'
19      });
20    </script>
21  </body>
```

Figure 6: Simple Vue.js Application With Components

Android Development

Q2.8: Figure 7 depicts an Android method that performs a potentially long-running operation¹ inside a new thread. Describe the reason why such an operation should be run in its own thread. (2P)

Q2.9: Describe the actual behaviour the method depicted in Figure 7 has and how this could be improved. (2P)

```
1 public void onClickButton (View view) {
2     new Thread (new Runnable() {
3         @Override
4         public void run() {
5             Button myBtn = findViewById(R.id.myBtn);
6             myBtn.setEnabled(false);
7             HttpURLConnection urlConnection = null;
8             try {
9                 URL url = new URL(getString(R.string.endpoint_url));
10                urlConnection = (HttpURLConnection)
11                    url.openConnection();
12                InputStream in = urlConnection.getInputStream();
13                //Process the input stream
14            } catch (Exception e) {
15                e.printStackTrace();
16            } finally {
17                if (urlConnection != null) {
18                    urlConnection.disconnect();
19                }
20            }
21        }).start();
22    }
```

Figure 7: Android Method Performing a Long-Running Operation

¹We have omitted handling the response in line 12 for the sake of brevity.

Part 3 – REST API Case (8P)

In Figure 8, you see a number of possible HTTP requests for a modified version of the current Gmail API. Gmail offers a RESTful API with access to different resources (here we use only *messages* and *labels*).

```
GET /users/:userId/labels
POST /users/:userId/labels
PUT /users/:userId/labels/:id

DELETE /users/:userId/:id/messages
GET /users/:userId/messages
```

Figure 8: Gmail API

Q3.1: Assuming that the API has been designed following the REST principles introduced in this course, describe what content and status code the following HTTP request returns (when successful). **(2P)**

1. GET /users/:userId/labels

Q3.2: Assuming that the API has been designed following the REST principles introduced in this course, describe what changes the following two HTTP requests cause on the server/in the database (when successful). **(2P)**

1. POST /users/:userId/labels
2. PUT /user/:userId/labels/:id

Q3.3: Assume that the following HTTP request deletes the message with id *:id* for user with id *userId*.

- DELETE /users/:userId/:id/messages

Describe what is wrong with the request format and how it should look like instead. **(1P)**

Q3.4: We would like to add an HTTP request that allows users to change any attribute of a specified *message*. What kind of request would we have to add following the REST principles? Name the required HTTP method and the URL. **(1P)**

Q3.5: Assume that two consecutive requests (without any other HTTP request in between) to the following endpoint produce different responses.

- GET /users/:userId/messages

Describe which behaviour this violates and why. **(2P)**

Part 4 – Reflection (6P)

Q4.1: In this course, we have covered web (frontend) development and mobile development. Apart from the programming languages, which differences do you see between developing an Android app and a web frontend with the same functionality? Describe three differences and give an example for each. (3P)

Q4.2: A typical software development life cycle includes the plan, develop, test, deploy and maintain steps² as depicted in Figure 9. Describe in which of the five steps security concerns can arise and, for each of those steps, give an example of a security concern. (3P)

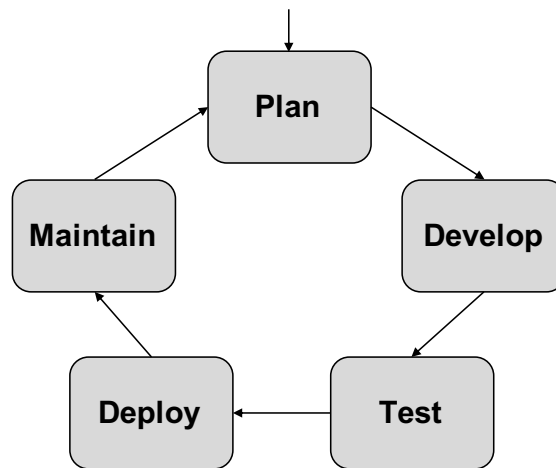


Figure 9: Software Development Life Cycle

²Note that, in practice, the order might change and steps might overlap.