

FACULTY OF ENGINEERING, BUILT ENVIRONMENT AND INFORMATION
TECHNOLOGY



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

DEPARTMENT OF INFORMATICS

INFORMATICS 354

Semester test

April 2019

Lecturers

Mr J Brosens
Mr RM Kruger
Mr B Liu

Time: 90 min

Marks: 75

Instructions

1. This paper consists of 4 sections and 4 pages.
2. Answer all the questions from these sections in the answer book provided.
3. This is a closed book written test, with no internet access.
4. Please hand in the yellow card when asked.

Section 1: Multiple choice questions

(15)

- 1.1 MVC is a/an design pattern
- a) Behavioural
 - b) Architectural**
 - c) Creational
 - d) A combination of or none of the above
- 1.2 The relationship between JSON and Angular is that:
- a) JSON is a lightweight data-interchange format that is used to send data between all elements of an Angular application.**
 - b) JSON is a heavyweight data-interchange format that is used to send data between all elements of an Angular application.
 - c) JSON is a heavyweight data-interchange format that is used to send data between an API and the Angular application.
 - d) JSON is a lightweight data-interchange format that is used to send data between an API and the Angular application.
- 1.3 APIs can be written in:
- a) C#
 - b) Javascript
 - c) Java
 - d) All of the above**
- 1.4 In a collaborative team that works on the same code files, to put the code on a remote repository with git, you should use which of the following sequence of commands?
- a) add, commit, pull, push**
 - b) add, commit, push
 - c) add, commit, checkout, push
 - d) None of the above
- 1.5 A merge conflict happens when:
- a) Two parties change the contents of a repository.
 - b) Two parties change the same file in a repository.
 - c) Two parties change the same line in a repository.**
 - d) Two parties change the same code in a repository.
- 1.6 An information system is:
- a) A program or group of programs designed for end users.

- b) Formal, sociotechnical, organisational systems designed to collect, process, store, and distribute information to help an organisation to answer questions and solve problems relevant to the mission of the organisation.
 - c) Software applications designed to collect, process, store, and distribute information to help an organisation to answer questions and solve problems relevant to the mission of the organisation.
 - d) Software applications designed to collect, process, store, and distribute information to help end users to answer questions and solve problems relevant to the mission of the organisation.
- 1.7 Reports in organisations help managers with:
- a) Doing their jobs better.
 - b) Decision making
 - c) Supporting and organising employees.
 - d) All of the above
- 1.8 When software is robust, it
- a) is well documented.
 - b) never crashes.
 - c) never crashes and handles whatever data it receives.
 - d) thoroughly tested, never crashes and handles whatever data it receives.
- 1.9 To move to another folder in cmd, you use:
- a) cd
 - b) dir
 - c) mkdir
 - d) "cd .."
- 1.10 To generate a new service in angular, you use:
- a) ng g c
 - b) ng g s
 - c) ng generate service
 - d) b or c
- 1.11 With asynchronous calls, you should remember that
- a) the software application will wait for the response, and you should handle the case where data is not received.
 - b) the software application will not wait for the response, and you should handle the case where data is received.
 - c) the software application will wait for the response, and you should handle the case where data is received.

d) the software application will not wait for the response, and you should handle the case where data is not received.

1.12 A desktop computer is a

- a) large computer.
- b) a personal computer that is not a laptop.
- c) a computer suitable for use on an ordinary desk.
- d) only laptops or PCs.

1.13 To run an electron app, you use:

- a) The angular cli.
- b) npm
- c) Node
- d) Electron-packager

1.14 Native applications for iOS are developed in

- a) Objective Java
- b) Submissive Javascript
- c) Objective C
- d) Submissive C

1.15 Observer is a/an design pattern

- a) Behavioural
- b) Architectural
- c) Creational
- d) A combination of or none of the above

Section 2: Theory

(25)

- 2.1 Describe MVC and all its components. (3)

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components is built to handle specific development aspects of an application.

- 2.2 What is an API, give an example of its usage. (3)

Application Programming Interface – “A set of routines, protocols, and tools for building software applications” and good example.

- 2.3 Once you have made a change to a piece of code on your local computer, list the steps to adding the code to a branch titled "DatePickerBranch" on the repository. (3)

git checkout <branch>; git add; git commit; git pull; git push

- 2.4 List the four information processes that typically form part of an information system. (2)

Collect, Process, Store and Distribute

- 2.5 Mention and describe four different report types. (4)

Summary reports - Summary reports aggregate data by accounting periods, geographic regions, business units or product categories. The reports consolidate information in a format that makes it easy for managers to review and analyse.

Trend reports - Trend reports allow managers to compare the performance of business units or product categories over time.

Exception reports - An exception refers to data that are outside of normal ranges.

Exception reports aggregate these unusual conditions and present them separately.

A timely reporting of exception conditions makes it easier for a manager to isolate cases that require immediate attention.

Ad hoc reports - On-demand reports provide specific information as and when needed.

- 2.6 List three characteristics of a good component in terms of component-based development? (3)

fully documented

thoroughly tested

robust, never crashes handles whatever data it receives

passes back appropriate error messages

designed with an awareness that it will be put to unforeseen uses

- 2.7 Describe how you link a component to an API? (1)

Subscribing to an Angular service that interacts with the API via HTTP/HTTPS protocol

- 2.8 Describe how you would go about building a web, a mobile and a desktop application that link to the same API using angular. (5)

a good explanation involving electron and ionic, components, services and angular

- 2.9 What is “separation of concerns” in programming terms? (1)

Separation of concerns is a design principle for separating a computer program into distinct sections so that each section addresses a separate concern.

Section 3: Practical

(35)

- 3.1. Consider the following JSON object and answer the questions that follow:

```
[{"Name": "Jack", "Surname", "Jones", "Age", "35", "Profession": "Professor"},  
{"Name": "Jill", "Surname", "Jones", "Age", "30", "Profession": "Engineer"},  
{"Name": "Bill", "Surname", "Jones", "Age", "55", "Profession": "Farmer"},  
{"Name": "Mary", "Surname", "Jones", "Age", "60", "Profession": "Professor"}]
```

- a) This JSON object comes from an api with the url `http://testapi.com/getUsers`, which only allows for get requests , write an Angular service function with the function head “`getUsers`” that receives the object. (3)

```
getUsers(){  
    return this.http.get('testapi.com/getUsers')  
}
```

- 3.2. Write the import functions to include the service “`userService`” in your `userComponents` ts file. (1)

```
import {UserService} from '../data.service';
```

- 3.3. Your components constructor header is `constructor(private data: UserService)`. Write the `ngOnInit` function that subscribes to the service, the data should be stored in a local variable called `users`. (4)

```
ngOnInit() {  
    this.data.getUsers().subscribe(  
        data=> this.users$ = data)  
    }  
}
```

- 3.4. Write the html template that will render a table with the JSON data received. Each object should be on a new line in the table. (5)

```
<table>  
  <tr *ngFor="let user of users$">  
    <td>{{user.Name}}</td>  
    <td>{{user.Surname}}</td>
```

```
<td>{{user.Age}}</td>
```

```
<td>{{user.Profession}}</td>
```

```
</tr>
```

```
</table>
```

- 3.5. You want to create a repository called “Xhonixer”. Describe the steps that you would use to get your angular application on a Gitlab repository, **give all of the git commands that you would use.** (5)

Create repo, clone it, add, commit, pull, push.

- 3.6. Give the command that you would use to run your angular application. (1)

ng serve

- 3.7. The JSON object is compiled from two tables in your SQL server database, namely, the user and the professions table. Write the API controller function that will get the data from the database and send it in JSON format to the client. (5)

[HttpPost]

```
public List<dynamic> getUsers([FromBody] UpdateStatusVM obj)
```

```
{
```

```
List<dynamic> toReturn = new List<dynamic>();
```

```
db.Configuration.ProxyCreationEnabled = false;
```

```
var users= db.Users.Include(z => z.profession).ToList();
```

```
..... create JSON object.....
```

```
return toReturn;
```

```
}
```

- 3.8. Assume that there is a predesigned CrystalReport, with relative path `~/Reports/UsersReport.rpt`, that contains a single table to show all users in the database. Furthermore, assume that the API also contains a definition for a DataSet called UserReportModel, which contains a table called User with columns for Name, Surname, Age, and Profession. The CrystalReport design is bound to the UserReportModel DataSet.

Your API has an endpoint, namely getUsersReport(). Provide the following code snippets that would need to be included in this function so that a report is effectively generated and returned to the client.

- 3.8.1. Provide code to create a CrystalReport instance and load it from the specified path (1)

```
ReportDocument report = new ReportDocument();
```

```
report.Load(Path.Combine(HostingEnvironment.MapPath("~/Reports/UsersReport.rpt"))));
```

- 3.8.2. Assume you have retrieved all user records from the database and have them loaded in a list of User objects. The user object has the following properties: Name, Surname, Age, and Profession. Provide code to create a new instance of the UserReportModel and copy property values from each of the user objects in the list to the User table of the newly created UserReportModel object (2)

```
UserReportModel model = new UserReportModel ();  
foreach(User user in users)  
{  
    DataRow row = model.User.NewRow();  
    row["Name"] = user.Name;  
    row["Surname"] = user.Surname;  
    row["Age"] = user.Age;  
    row["Profession"] = user.Profession;  
    model.User.Rows.Add(row);  
}
```

- 3.8.3. Provide code to link the report object created in 3.8.1 with the model object created and populated in 3.8.2 (1)

```
report.SetDataSource(model);
```

- 3.8.4. Provide code to read the data bound CrystalReport object into a Stream object as PDF (1)

```
Stream stream = report.ExportToStream(CrystalDecisions.Shared.ExportFormatType.PortableDocFormat);
```

- 3.9. Describe the technical steps you would take to convert your angular application into a desktop application using electron. Give the code where it would be required. (4)

```
.....Install electron.....change main.js.....change  
package.json.....build the electron app.....
```

- 3.10. Imagine, your application was an ionic app, how would you go about testing it in a browser? (1)

```
ionic serve
```

- 3.11. Imagine, your application was an ionic app, how would you go about building a version for android? (1)

```
ionic cordova build android
```


