THE UNIVERSITY OF WESTERN AUSTRALIA

DESK No. | | | |

FAMILY NAME: _____

GIVEN NAMES: _____

SIGNATURE: _____

STUDENT NUMBER: | | | | | | | |

**SEMESTER 1, 2020 EXAMINATIONS**

# CITS3403
# Agile Web Development

Physics, Mathematics & Computing
EMS

This paper contains: Pages **(including title page)** Time Allowed: **2:00** hours

**INSTRUCTIONS**:
This exam consists of eight questions worth 50 marks all together.
Attempt all questions, and write your responses in the boxes provided.
If you do not understand any material, explain your confusion in the notes field provided, and try to make a reasonable assumption so that you can continue with the exam.
No notes, written materials, or electronic devices are allowed.
Reference materials are provided for the exam, as a pdf you can refer to.

**THIS IS A CLOSED BOOK EXAMINATION**

| **SUPPLIED STATIONERY** | **ALLOWABLE ITEMS** |
|---|---|

**PLEASE NOTE**

*Examination candidates may only bring authorised materials into the examination room. If a supervisor finds, during the examination, that you have unauthorised material, in whatever form, in the vicinity of your desk or on your person, whether in the examination room or the toilets or en route to/from the toilets, the matter will be reported to the head of school and disciplinary action will normally be taken against you. This action may result in your being deprived of any credit for this examination or even, in some cases, for the whole unit. This will apply regardless of whether the material has been used at the time it is found.*
*Therefore, any candidate who has brought any unauthorised material whatsoever into the examination room should declare it to the supervisor immediately. Candidates who are uncertain whether any material is authorised should ask the supervisor for clarification.*

*Candidates must comply with the Examination Rules of the University and with the directions of supervisors.*
*No electronic devices are permitted during the examination.*
*All question papers and answer booklets are the property of the University and remain so at all times.*

This page has been left intentionally blank

1. (5 marks)

- (3 Marks) Explain the key differences between agile software processes and any traditional software process.

- (2 Marks) Explain the difference between HTTP and HTML.

2. (10 marks) Describe, as clearly as you can, the appearence and style of the page generated by the attached code, when it is rendered by a modern web browser.

```
<html>
  <head>
    <meta charset='utf-8'/>
    <title>10 marks!</title>
    <style>
      body {color:black; background-color:yellow;}

      .a {margin:1em; border:solid; padding:2px; background-color:white;}

      ol.b {list-style-type: lower-roman;}

      ul ul li + li{color: blue;}

      ul > li.h:hover {color: green}
    </style>
  </head>
  <body>
    <h1>Agile Wed Dev Exam, Question 2</a>
    <div class='a'>
    <ul>
      <li> An ordered list:
        <ol class="b">
          <li class="h">Zoom</li><li>Teams</li><li>Slack</li>
        </ol>
      </li>
      <li> An unordered list:
        <ul class='b'>
          <li>Examsoft/Examplify</li><li>LMS</li><li class="h">cssubmit</li>
        </ul>
      </li>
    </ul>
  </body>
</html>
```

3. (10 marks) Write an HTML page, including javascript, to have the following functionality:

(a) The page should have a heading, "My To Do List".

(b) The page has an unordered list of items, the to-do list.

(c) There is a text box, where new items can be written, with a button labelled "Add" beside it.

(d) When the Add button is pressed, if the text box is not empty, the text is added to a new item at the bottom of the to-do list.

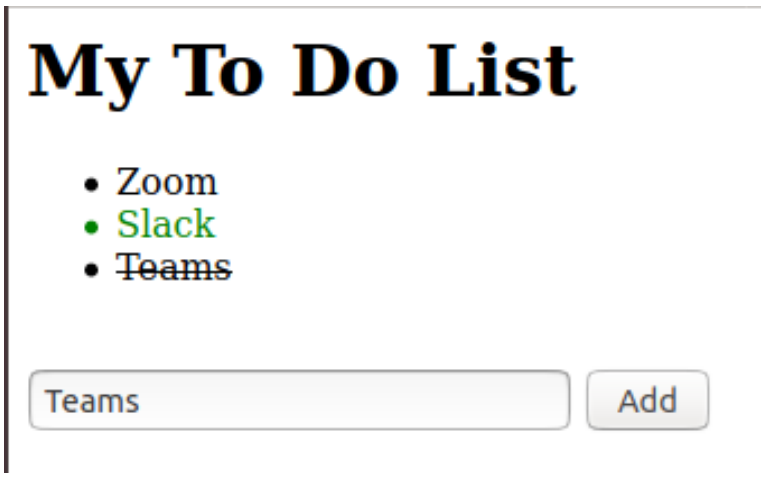
(e) When the mouse hovers over an item, it becomes green.

(f) When a list item is clicked, a line is placed through the item (hint: you can set text-decoration: line-through to achieve this effect).

(g) When a list item with a line through is clicked, the line through is removed.

(h) When any list item is double-clicked, it is removed.

An example of what the webpage may look like is below.



4. (5 marks)

*"The web was designed to be scalable and robust, but it was not designed to be secure."*

Discuss this statement with respect to the six characteristics of the Representative State Architecture.

5. (10 marks) Using your Flask project as an example, describe the process of registering a new user account using a Flask application, noting:

- What HTTP requests and responses will be sent.
- What database operations will be performed.
- How the forms are generated and served to the user.
- How the user-submitted data is validated.

6. (10 marks)

a. [5 marks] Consider the flask code below to set up some models linked to a database via SQLAlchemy. Describe the schema of the database as clearly as possible, as well as the models produced.

```
1  from app import db
2  from sqlalchemy.orm import relationship
3
4  class Student(db.Model):
5      __tablename__='students'
6      id = db.Column(db.String(8), primary_key = True)
7      first_name = db.Column(db.String(64))
8      surname = db.Column(db.String(64))
9      project_id = db.Column(db.Integer, db.ForeignKey('projects.project_id'),nullable=True)
10
11     def get_partners(self):
12         if not self.project:
13             return []
14         team = self.project.team
15         team.remove(self)
16         return team
17
18     def __repr__(self):
19         return '[Number:{}, Name:{}]'.format(self.id, \
20         self.__str__())
21
22 class Project(db.Model):
23     __tablename__='projects'
24     project_id = db.Column(db.Integer, primary_key = True)
25     description = db.Column(db.String(64))
26     team = relationship('Student', backref='project')
27
28     def __repr__(self):
29         return '[PID:{}, Desc:{}]'.format(\
30             self.project_id,\
31             self.description)
```

b. [5 marks] Write some unit tests to test the `get_partners` method in the code from part a. The tests should have with full coverage of the method. Some basic boiler plate code is available below, which you can use if you wish, but it is not required.

```
import unittest, os
from app import app, db
from app.models import Student, Project

class StudentModelCase(unittest.TestCase):

  def setUp(self):
    app.config['SQLALCHEMY_DATABASE_URI']= 'sqlite:///test.db'
    self.app = app.test_client()
    db.create_all()
    #initialize test fixtures here...

  def tearDown(self):
    #reset test fixtures here...
    db.session.remove()
    db.drop_all()

  def test_get_partners(self):
    #tests and assertions go here...

if __name__=='__main__':
  unittest.main(verbosity=2)
```

End of Paper