HOGESCHOOL ROTTERDAM / CMI

Project 5-6

Developing a web application

INFPRJ0156

ECTS: 8
Module responsibility: F. Di Giacomo

Description of the course

Modulenaam:	Project 5-6 – Developing a web application
Modulecode:	**INFPRJ0156**
Aantal studiepunten en studiebelastinguren:	This course provides you with eight (8) study points, which corresponds to a workload of 224 hours. The recommended distribution of these 224 hours during the study weeks is as follows: Supervised lectures: Kick-off: 3 * 50 minutes 2,5 hours Project lesson (during 6 weeks): 6 * 7 hours 42 hours Presentations of the product: 2 * 3 * 50 minutes 5 hours Unsupervised hours: Time to work on the project incl. literature study 174,5 hours **Total** **224 hours**
Vereiste voorkennis:	Previous knowledge gathered from Dev-Anl-Skl courses is required.
Werkvorm:	Project-based education (group-work)
Toetsing:	Examination is based on the delivered product and the process of the project.
Leermiddelen:	Development tools, Dev and Anl courses literature
Draagt bij aan competentie:	▪ Design (**O**ntwerpen) ▪ Implementation (**R**ealiseren)
Leerdoelen:	▪ [**O**] the student can design the application at all levels (use cases, modular design, ERD, etc.); ▪ [**R1**] the student can use testing techniques to validate the correctness of the implementation and to grant the quality of the final product; ▪ [**R2**] the student can implement a layered data-driven web application by using modern technologies for web development; ▪ [**R3**] the student can refactor the application to include feedback
Inhoud:	You learn to work in a group context (*process*) and to realize a realistic project assignment (*product*) for a client. A company is involved in the kickoff (and *may* give feedback on the best products).
Opmerkingen:	Attendance is obligatory. Groups will be made by the tutor. Groups will be 4 or 5 students each and are made randomly. *Note*: Groups that, during the course of the project, lose team members and remain with 3 (or less) students, will discuss with their project teachers about adjusted criteria for the evaluation.
Modulebeheerder:	F. Di Giacomo
Datum:	19 September 2016

1. General information

1.1 Introduction

This project will last 6 weeks in total: three at the end of the first quarter (weeks 8, 9, 10 of OP1) plus three weeks at the end of the second quarter (weeks 8, 9, 10 of OP2). You will work in a project team and you will be working together on an assignment that has to be completed by the end of week 10 of OP2. You will also receive an intermediate grade at the end of week 10 of OP1, which will determine if you are allowed to continue the project or not. You will be followed regularly by your Product Owner and Tutor; they will monitor your progress and will give you feedback on the process and the product.

1.2 Relationship with other courses

This project is related to previous courses in the following way:

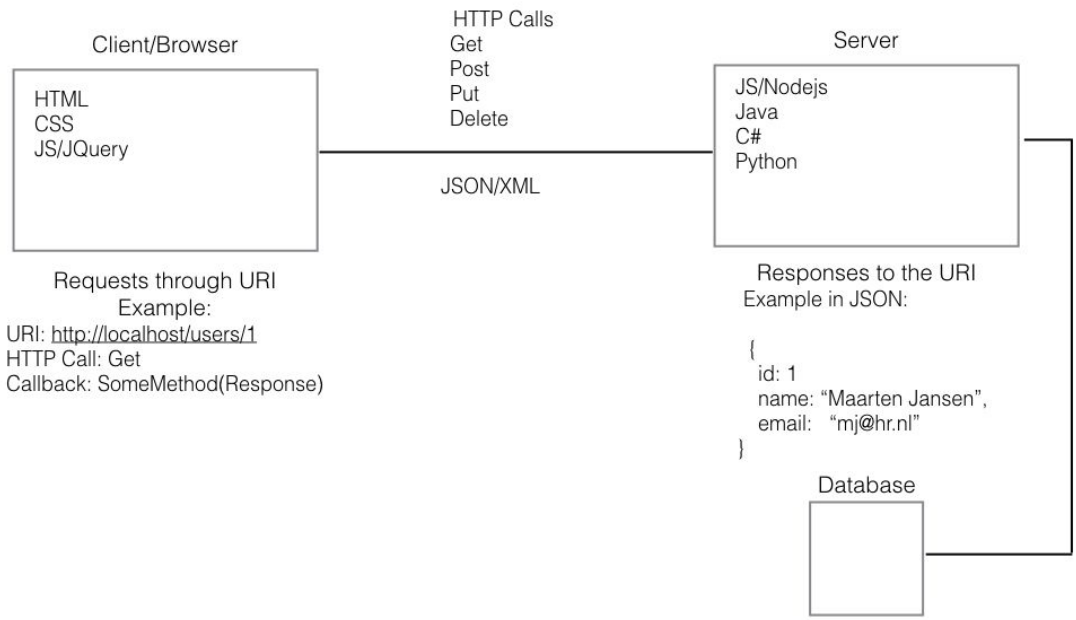
- INFANL02-2: conceptual database design with Entity-Relationship Diagrams and use of SQL queries.
- INFDEV03-5: normalization of relational databases, use of NoSQL database systems and relative query language.
- INFDEV01-4: code maintainability and flexibility by using appropriate design patterns for the client and server application.
- INFDEV02-6A: data structures usage and optimization.
- INFANL01-6: setting up a testing environment for the server-side of the application.

1.3 Learning materials

- N@tschool (for deliveries), Classroom (to communicate during the project)
- Github, https://github.com/

1.4 Technical requirements

The architecture you must build is a *restful application*[1] with an MVC architecture. In the following picture you can see the general structure of this architecture:



[1] https://en.wikipedia.org/wiki/Representational_state_transfer

The technologies you are allowed to use in this project are the following:

- *Client side*: choice of language between C#, Javascript, Java, Python, (optional JQuery). No AngularJS. No use of frameworks, but only one of the following *micro-frameworks* for marshalling HTTP protocol requests:
    - WCF (for C#) https://msdn.microsoft.com/en-us/library/ms731758.aspx
    - Restify with nodejs (for Javascript) http://restify.com/
    - Spark (for Java) http://sparkjava.com/
    - Flask (for Python) http://flask.pocoo.org/
- *Server side* (implementing the listener to handle the requests from HTTP): choice of language between C#, Javascript, Java, Python.
- *Database*: layer with an interface to Document database or any RDBMS.

2. Program and contents

2.1 Assignment

Build a webshop for a category of products chosen by you, with the following functionalities.

1. The webshop application must distinguish between **admin** and normal **users** (**customers** or **unregistered users**).
2. Users can visualize lists of items filtered by certain properties (i.e. type, price, ...).
3. It must be possible to register to the webshop by providing a unique username and password, plus some personal information (age, address, email, etc…).
4. Only customers can buy products but unregistered users can add products to the cart and they will be asked later to login or register.
5. The admin must be able to visualize at least three plots with statistics on the data available about the store (customers, sales, products, ...). Make sure that your P.O. is satisfied by the three plots you plan to create and adjust your proposals to his/her feedback/wishes.
6. The admin must be able to modify information about the user accounts, like resetting the password, deleting it, blocking for legal issues, …
7. The customer must be able to see the history of his/her orders.
8. The customer must be able to save bought products in a favourite list.
9. The customer must be able to save a wish list.
10. The wish list of a customer is by default private, but the customer might decide to make it public, and doing so enables all other users to see it.

The description above contains the basic and minimal functionalities that your product must include. You are of course free to develop and include additional features (coordinate this beforehand with your P.O.).

Note: read the attachments containing the evaluation forms carefully to know exactly what you will be graded on.

2.2 Week scheme

The project covers the last three weeks of OP1 and the last three weeks of OP2: week 8-OP1 (sprint 1), week 9-OP1 (sprint 2), week 10-OP1 (sprint 3), week 8-OP2 (sprint 4), week 9-OP2 (sprint 5), week 10-OP2 (sprint 6). In this project you are free to choose a Project methodology (Scrum, Waterfall, Kanban, …); however, we will still use Scrum terminology in this course description, because you are familiar with the terms (P.O., sprint, etc.).
Every week you will meet both your Product Owner and your Tutor (in separate timeframes). For the exact timeframe of these lessons, look at your class schedule. In addition to this, there will be a final presentation scheduled at the end of each block (sprint 3 and sprint 6).

The goal of the first three weeks is to complete the *design of the architecture* and to have a *working prototype* (to show that you have started programming with the languages/technologies chosen by your group). During those three weeks you are expected to experiment as much as possible, following a trial-and-error approach to improve your knowledge of the technologies and your product. Always keep the P.O. informed of what you are experimenting with. You also have to coordinate with the P.O. to establish which three features must be done by the end of the first three weeks (see Attachment 3).
At the end of the project, you must have fully completed the *product*, including *testing* and *errors handling*. Apart from the constraints for the end of sprint 3 and 6 (better specified in the evaluation attachments), students are required to plan their weeks *independently*, coordinating with their P.O.. This means that each week each group has the responsibility of giving their P.O. a clear and written document about what they expect to do during the following week (for example, through a Sprint Backlog). If you do not do this or if you

do not respect the promises made[2], you will receive a penalty in the form of a *yellow card* (see the effect of yellow cards in the evaluation attachments).

In the following table you can see the lessons/meetings of each week and the corresponding deliveries. The deliveries will be done through N@tschool[3]. **Unless specified otherwise**, deliveries must be done before the start of the corresponding meeting.

Week	Sprint	Meetings[4]	Deliveries for each meeting
8-OP1	1	- P.O. - Tutor	- P.O.: by end of meeting, plan for following week - Tutor: none
9-OP1	2	- P.O. - Tutor	- P.O.: by end of meeting, plan for following week - Tutor: none
10-OP1	3	- P.O. - Tutor - Presentations	- Tutor: [optional] STARR (draft to receive feedback) - *Presentation*: working prototype + "final version" of the architectural diagrams (inclusive feedback from previous weeks)
8-OP2	4	- P.O. - Tutor	- P.O.: by end of meeting, plan for following week - Tutor: none
9-OP2	5	- P.O. - Tutor	- P.O.: by end of meeting, plan for following week - Tutor: none
10-OP2	6	- P.O. - Tutor - Presentations	- Tutor: STARR (definitive version) - *Presentation*: final product (code+screenshots), diagram of the testing pipeline, updated version of the architectural diagrams, presentation slides, Github plot

Important note: for all deliveries, read the evaluation attachments for further details.

[2] Remember that *partially completed* tasks will bring you to a yellow card. For example, if you promise to complete all the sequence diagrams for the following week, but then you create only a couple of them, you will get a yellow card. Be careful in what you promise during the planning.
[3] Choose one group member who will always have the duty of uploading the documents to deliver for his/her group in a compressed file called "INF2X – Group Y – Week Z".
[4] If compatible with personal work schedules, Tutors are invited to attend the P.O.s lessons, and vice versa.

3. Evaluation

	Evaluated by	Evaluated through	Partial result
Individual contribution and reflection	Tutor	*Attachment 1* (Evaluation form Tutor)	Individual result between +1 and -1 or No Go (for individual commitment)
Weekly progress (sprint 2,3,5,6)	P.O.	*Attachment 2* (Evaluation form Progress)	Possible yellow cards
Intermediate milestone (sprint 3)	P.O.	*Attachment 3* (Evaluation form Milestone)	GO/REPAIR/NO GO (determines if the group is allowed or not to continue with the project during OP2)
Final product (sprint 6)	P.O.	*Attachment 4* (Evaluation form Final Product)	Grade between 0 and 10 points

The final grade of the project for each student is the *sum of the grades coming from Attachment 4* (group grade) *and Attachment 1* (individual grade), with a maximum of **10 points** and a minimum of **1**.
The forms associated to each evaluation part are given in the attachments.

3.1 – Examples

The following partial grade:
- 5 points from Attachment 3

means that the group cannot continue with the project in OP2 → onvoldoende

The following partial grades:
- -1 point from Attachment 1
- 8 points from Attachment 3
- 6,5 points from Attachment 4

bring to a final grade of -1 + 6.5 = **5.5** (voldoende)

The following partial grades:
- **Individual No Go** from Attachment 1
- 9 points from Attachment 3
- 7 points from Attachment 4

bring to a final grade of onvoldoende

3.2 – Herkansing (resit)

In case of an insufficient grade (*onvoldoende*) for INFPRJ0156, the following scheme applies:
- If you got an individual No Go, then you **cannot** repeat the project this year;
- If your final grade (sum of Attachments 1 and 4) is smaller than (<) 4.5, then you **cannot** repeat the project this year;
- In all other cases, you have to implement individually at least one *complete feature* given by the P.O., starting from the client-side and ending into the database. Each individually implemented feature has a value of 1 point. The final grade is 4.5 plus the amount of points obtained through your individually programmed additional features. The time available to implement these features is *2 weeks*.

If you do not succeed in the resit (or if you cannot repeat the project this year), you will need to follow this course again during next school year.

Attachment 1 – Evaluation form [Tutor]

CLASS: INF2…., GROUP …

Attendance

Attendance will be checked during every lesson (both P.O. meetings and Tutor meetings). Attendance is obligatory to get a grade for the project. Only when a valid reason is given to the Tutor or Product Owner prior to the lesson, an exception might be granted.
The Tutor will write down the attendance to the Tutor meetings and individual points in the following table:

	OP1 week 8 attendance	OP1 week 9 attendance	OP1 week 10 attendance	OP2 week 8 attendance	OP2 week 9 attendance	OP2 week 10 attendance	Individual points (+1 / 0 / -1 / No Go)
Name 1							
Name 2							
Name 3							
Name 4							
Name 5							

The Tutor has the opportunity to give you a higher or lower grader for your individual commitment. The individual commitment will be graded with +1 / 0 / -1 or a No Go. As opposed to last year, there are no specific tutor parts that you have to deliver. However, there are things you can do to 'earn' a point but you can also 'lose' a point if you do not do what is expected of you. Specifics are written down below.

Reasons when a Tutor, eventually in accordance with the group, might decide to give a different individual grade (see also the table below for a summary):

+1: the student has proven that he/she has passed one of the learning goals using the STARR-method[5] adequately AND a project method (such as Scrum, or for example KANBAN or the Waterfall method) has been adopted by the group consistently and completely.

0: the student has done his/her part during the project but the STARR-method was not used adequately or a project method has not been adopted consistently and completely by the group.

-1: no STARR is present or the STARR is of very low quality making it unclear if the learning goal was passed.

No Go: the student has delivered considerably less work during the entire project or almost all weeks. He or she has been a burden to the group's development and cannot really show the amount of work he or she has done individually. The student was absent for a substantial amount of time and therefore has not done enough work. Being absent for several days without compensatory behaviour afterwards might be a reason to get a no go.

[5] The STARR-method is explained at the end of this attachment. The STARR report will be checked by both Tutor and P.O..

STARR method → **Project method ↓**	Used adequately	Not used adequately enough	Very low quality or missing
Applied consistently and completely	+1	0	-1
Not applied consistently and completely	0	0	-1

STARR-method

In the upcoming projects (7/8, ICT-lab, etc.) you will have to work with the STARR-method to explain your progress with regard to the learning goals. In Project 5/6 there is already a chance to practice with this method. As mentioned above, if you do it really well, you could get a +1 on your final grade.
The STARR report is **individual**.

Pick one of the learning goals that are written down in this course manual. Use concrete examples to explain how you passed that goal. Use the following method to describe your actions:

Situation:
- Can you give an example of a situation that proves that…?
- What was the context of the situation?

Task:
- What was your task?
- What tasks did you perform?
- Can you specify in further detail what your role has been?

Activity:
- What actions did you perform?
- What steps did you take?
- How did you approach the task?
- What did you take into consideration to perform the task like you did?

Result:
- What was the effect, the result of your action?
- What are your thoughts about the result?
- What did the other involved parties (group members, P.O., etc.) think of the result?

Reflection:
- In hindsight, what could you have done differently?
- What can you improve next time?
- If, instead of …. you would have done….., how could you have achieved that?

If you want to get feedback on your use of this method you can deliver a concept version of your STARR in week 10 of OP1. The final version will have to be delivered in week 10 of OP2.

Attachment 2 - Evaluation form progress [P.O.]

CLASS: INF2…., GROUP …

Note: during the first P.O. meeting (in week 8-OP1) you will hand the P.O. the list of activities that will be completed before the following meeting (week 9-OP1). This means that the first "progress check" happens during the meeting of week 9-OP1. The second "progress check" is at the beginning of week 10-OP1. At the end of week 10-OP1 the groups will present their work (you will be evaluated for this through Attachment 3).

The same happens during OP2: in the first P.O. meeting (week 8-OP2) you prepare the list of features to do for the following week, which means that the first "progress check" is in week 9-OP2. The second "progress check" is at the beginning of week 10-OP2. At the end of week 10-OP2 final presentations will take place (you will be evaluated for this through Attachment 4).

P.O. meeting	**Progress check**		**Feedback on the progress**
Week 9-OP1 (sprint 2)	The group has delivered what was promised at the meeting of week 8-OP1?	▢ Yes ▢ No → Yellow card	
Beginning of week 10-OP1 (sprint 3)	The group has delivered what was promised at the meeting of week 9-OP1?	▢ Yes ▢ No → Yellow card	
Week 9-OP2 (sprint 5)	The group has delivered what was promised at the meeting of week 8-OP2?	▢ Yes ▢ No → Yellow card	
Beginning of week 10-OP2 (sprint 6)	The group has delivered what was promised at the meeting of week 9-OP2?	▢ Yes ▢ No → Yellow card	

Attachment 3 – Evaluation form intermediate milestone [P.O.]

CLASS: INF2…., GROUP …

Note: a **prerequisite** to get a grade in this attachment is that the feedback from the P.O. of the previous weeks has been included.

		Value	Check
Prototype, including *at least* three features	Feature 1: decided in coordination with the P.O.	1 point	▢ Yes ▢ No
	Feature 2: decided in coordination with the P.O.	1 point	▢ Yes ▢ No
	Feature 3: decided in coordination with the P.O.	1 point	▢ Yes ▢ No
Database	Complete (and normalized if relational)	1 point	▢ Yes ▢ No
	ERD	1 point	▢ Yes ▢ No
Architectural diagrams	Use cases representing all interactions planned to implement plus their descriptions using the template from Astah	1 point	▢ Yes ▢ No
	Deployment diagram of the application including components	1 point	▢ Yes ▢ No
	(Conceptual) class diagram of the application	1 point	▢ Yes ▢ No
	Sequence diagrams for the following scenarios: registering a new user, login, adding a product to the cart, completing a purchase, deleting a user, visualizing the most sold items	1 point	▢ Yes ▢ No
	Sum of the points:	_____	

Result
If the sum of the points is…

- 8 or 9 ⇒ GO (you can continue with the project in OP2)
- 5, 6 or 7 ⇒ REPAIR
- Else (< 5) ⇒ NO GO (you cannot continue with the project in OP2)

The REPAIR means that the group must work some more on their delivery, which will be re-evaluated (using this attachment) by the start of week 4-OP2. To get a GO, the result must be 8 or 9.

Attachment 4 – Evaluation form final product [P.O.]

CLASS: INF2…., GROUP …

The final grade for this part is obtained by summing up the points for each criteria.

CRITERIA	POINTS
• The delivery is complete (code of the final product, diagram of the testing pipeline, updated version of the architectural diagrams, contributors plot from Github, screenshots of the application, presentation slides) • Code is commented • Each code file contains a maximum of 350 lines of non-generated code	**Prerequisite** to get the points
A) The product is complete, that is it includes all the features stated in your product backlog, which contains at least the features mentioned in the description of the assignment (see paragraph 2.1).	YES => 3 points NO => 0 points
B) There is a delivery pipeline made by a testing server and a production server. Moreover, *all* features actually implemented in the application (also the optional ones) are accompanied by a suite of tests (unit+integration) supported by running examples, both successful and failed.	YES => 2,5 points NO => 0 points
C) There is a document containing all the architectural diagrams made in the first three weeks, improved and updated with respect to the current state of the product.	YES => 1 point NO => 0 points
D) Other optional additional features have been implemented and tested: ❑ [up to 1p] Encryption ❑ [up to 1p] Single page application ❑ [up to 1p] Design patterns for interfacing with the DB (no SQL but method calls) ❑ [up to 1p] Containerized deployment ❑ [up to 1p] Caching of frequently used queries ❑ [up to 1p] Multiplatform ❑ [up to 1p] Concurrency in the backend ❑ Other features approved by the P.O.	**NOTE**: If you do not get the points for A (product completeness) and B (testing), no points for *additional features* will be given. Score of additional features: _____
The group received yellow cards...	0 yellow cards ⇒ 0 points 1 yellow cards ⇒ -0.5 points 2 yellow cards ⇒ -1.5 points 3 yellow cards ⇒ -3 points 4 yellow cards ⇒ NO GO
Sum of the points:	_____