INF 3176: Advanced development environment

Practice

Dr. Azanzi Jiomekong

Objective: The main goal of this project is to allow students, on the one hand, to capitalize the development techniques acquired during the first two years of their study. On the other hand, to use advanced development techniques that will be presented in this lecture in a real project. Thus, to avoid duplicates, the students will use the techniques presented in this lecture in the project of the lecture INF3196.

Methodology: The methodology that will be used is the Scrum methodology. To allow students to master the Scrum methodology, the students will work in groups of 5 and apply the group practices during the team work. The lecturer will be the product owner and the students designate the Scrum Master every week. The lecturer will assess the progress of the work and give all the necessary information that will help the group to improve their skills in mobile application development.

NB: The overall exercises are out of more than 20 for the continuous assessment and more than 30 for the practice. However, students' marks will be maintained out of 20 for the continuous assessment and 30 for the practice.

Continuous assessment /20

Exercise 1: requirements engineering (5pts)

Application of the Scrum practice that will be identified during the lecture

Exercise 2: requirements engineering (4pts)

1- Discuss with the supervisor of your project and identify functional requirements. Functional requirements describe what the system should do. **2pts**

2- Identify the non-functional requirements by justifying each of them. **1pt**

3- Give the Product-Backlog involving the backlog items that you have to proceed in order to build your product **1pt**

Exercise 3: Actors and use cases (3.5pts)

In this exercise, the UML language is used to determine the boundary of the system.

1- Given the exercise, what are the actors? **0.5pt**

2- Give the use cases classified by actors (for a good presentation, you should present in the table). **0.5pt**

3- Give the description of the use cases using the text formalism **0.5pt**

4- Give the description of the use cases using sequence diagrams **0.5pt**

5- Give the state machine diagrams of some important aspects of your system **0.5pt**

6- Build a table containing the main classes (followed by their description) necessary in this exercise **1pt**

Exercise 4: development environment (5.5pts)

1- Given your skills in different programming languages, determine the one you want to use during this exercise **0.5pt**

2- Choose (by justifying) and set-up your operating system **0.5pt**

3- Identify and set-up the Integrated Development Environment, the Frameworks and the APIs **1pt**

Given the complexity of your project, you decided to use DevOps.

4- Propose the DevOps architecture for the development of this game **0.5pt**

5- Set-up the DevOps tools choose given the architecture propose in question 4 **2pts**

6- Build a table (see the table below) containing the list of technologies you want to use in your project, their license their Open Source equivalence, their proprietary equivalence and the justification of your choice **1pt**

Technology	License	Open source equivalence (2)	Proprietary equivalence (2)	Short justification of the choice

Exercise 4: UI design (2pts)

1- Design the main UI **1pt**

2- Design the other UI **1pt**

Exercise 5: Software design (4pts)

1- Propose a class diagram **0.5pt**

2- Propose object diagrams (justify each object diagram you are proposing) **0.5pt**

3- Given to the class and the object diagrams, propose a package diagram **0.5pt**

4- By considering that we are using the monolithic architecture, propose this architecture **1pt**

5- Identify and describe the main algorithms that will be used during the development of this project **1.5pt**

Practice /30

Exercise 6: development (14pts)

1- Given the previous exercises, develop your product - development + algorithm **(5 + 5 = 10pts)**

2- Build the functional test **1pt**

3- Test your program **1pt**

4- Propose by justifying the software license you want to apply to your product (**NB:** a Software Composition Analysis may be used to justify your response) **1pt**

5- Evaluate your system using Technology Acceptance Model **1pt**

Exercise 8: Microservices architecture (11pts)

Given that the application can grow everyday, and become very big, to avoid the limitations of monolithic architecture, you decided to adopt the microservices architecture. This architecture helps us to structure the application as a collection of services that are highly maintainable, testable and loosely coupled. This is to enable the rapid, frequent and reliable delivery of the application.

1- Give five reasons to move the application to the microservices architecture (if there is no reason, explain) **1pt**

2- Build a table containing the different services and their description and the justification of the choice **1pt**

Microservice	Description	Justification

3- Use the question 2 to propose the microservices architecture of the application **1pt**

4- Use the question 3 to propose a deployment architecture **1pt**

5- Propose the new development environment by considering the new choices **2pts**

6- Propose the new version of the project, based on microservices architecture **5pt**

Exercise 9: Model-Driven Engineering (10pts)

Finally, you decided to give the possibility to players to design and automatically generate this application.

1- Propose the corresponding model (DSL)

2- Identify the use cases of the modeling and the generation environment

3- Propose the UI for the modeling your application

4- Propose the backend used for the generation of the new product

5- Test your application