Software Engineering and Project Management - Final Exam

2019-06-19. **Duration:** 9:00 - 12:30

Start by reading all the questions, to see if anything is unclear. The teacher will be available to answer any doubts at davide.vega@it.uu.se until the end of the exam.

The exam has a total of 40 points and you need 21 points to pass the exam (extra points will be added afterwards, once the exam is already graded). The grade is on the scale U/G.

Any preexisting material (including the course book, slides and your coursework) may be used, but:

- You must not communicate with anyone else regarding the exam or the course.
- When an exam question requires an example, you must produce your original example. In other words, examples from the course book, slides or solutions provided with the questions bought are not allowed.

Answers must be written in English and submitted in digital format. Submit your answers preferably in PDF (.pdf) format, alternatively in text (.txt) or markdown (.md).

The exam file must named "1DL251_<ANONYMOUS_EXAM_CODE>.pdf" where the <ANONYMOUS_EXAM_CODE> must by substituted by the code provided when you registered to the exam in Ladok.

Next to each question, you will find the maximum possible points. Consider 1 page ≈ 500 words.

The format and extension of your answer should adapt to the type of questions:

- **Short questions:** explanations, reviews, comparisons… the *optimal* answer is usually a few (2 or 3) paragraphs (This does not include drawing).

- **Questions for reasoning:** your arguments are presented in a concise, well-reasoned way,… and written with your own words. The *optimal* answer is usually less than 2 pages. (This does not include drawing).

CONTINUES NEXT PAGE

A checklist of common mistakes that cost points:

- Answer all **6** questions. A bad answer never gives less points than no answer.

- Read the question again after you have written the answer. Verify that you have actually answered the question. Verify that you answered *all* parts. Verify that you have *not* hidden the answer between many other irrelevant comments about the topic. *Cursives are added to highlight the key elements of the questions.*

- In particular, don't forget to give an example if that is requested, and make it a concrete one.

- Make sure all examples provided are originals.

- When a question asks you to compare two things A and B, make sure to highlight the contrasts: their differences. I do *not* want a full description of A and a full description of B, leaving it to me to find the differences. It is better to use tables than paragraphs of text to compare.

- Make sure you have named your file as ***1DL251_<ANONYMOUS_EXAM_CODE>.pdf***

CONTINUES NEXT PAGE

Short questions.

Question 1. [3 points].
Describe in detail one (1) agile process (Kanban or Scrum) using a software project as an *example*. *Justify* your decision.

Question 2. [6 points].
The diagram 1 (page 5) is a partial *class diagram* of a banking system.

2.a (2/6 points) *Indicate* three (3) design patterns from the diagram.

2.b (4/6 points) For two of them, *identify* the class/es involved and *explain* how the pattern is used in the context of the example.

Question 3. [5 points].
3.a (2/5 points) *What* are code reviews? or *Why* are they useful? *List* four (4) different processes for code review.

3.b (1/5 points) *Divide* the four (4) processes into two groups, A and B, with two (2) processes each. Indicate at least one (1) *common disadvantage* of the processes in group A compared with the processes in group B. All processes in group A must have the same disadvantage when compared with all processes in group B.

3.c (2/5 points) *Explain* the two (2) processes in group A.

Questions for reasoning.

Question 4. [9 points].
Your team is entrusted with the quality assessment of a streaming software system from the 80's that recommends new songs based on the users' preferences and their previous purchases on the platform.

4.a (3/9 points) *Describe* with a *diagram* what would be the process to decide if it is better to improve the current software or to start from scratch the application requested with new tools.

4.b (3/9 points) Assume that the team has decided to improve the current software. *Propose* and *describe* two (2) inspection methods to use. Justify your answer.

4.c (3/9 points) *Write* a complete test suite including one (1) stress test, one (1) integration test and one (1) load test for the service described at the beginning of this question with special emphasis on the elements of a test case. Three (3) tests in total.

CONTINUES NEXT PAGE

Question 5. [9 points].
You are part of a software engineering team, entrusted with the task of eliciting an online social media service to rent musical instruments for short periods of time (~2-3 days).

5.a (3/9 points) *Explain step by step* the process of requirements engineering, with special emphasis on *which* actors and/or stakeholders are responsible for each step of the process and *what* are their responsibilities.

5.b (2/9 points) *Compare* functional and non-functional requirements. *Indicate* at least two (2) differences between them.

5.c (2/9 points) Describe two (2) non-functional requirements, each one of different type, for the software described at the beginning of this question. Indicate the type of each requirement.

5.d (2/9 points) *Write* one (1) complete user story, with a corresponding *diagram*, related with the application described at the beginning of this question.

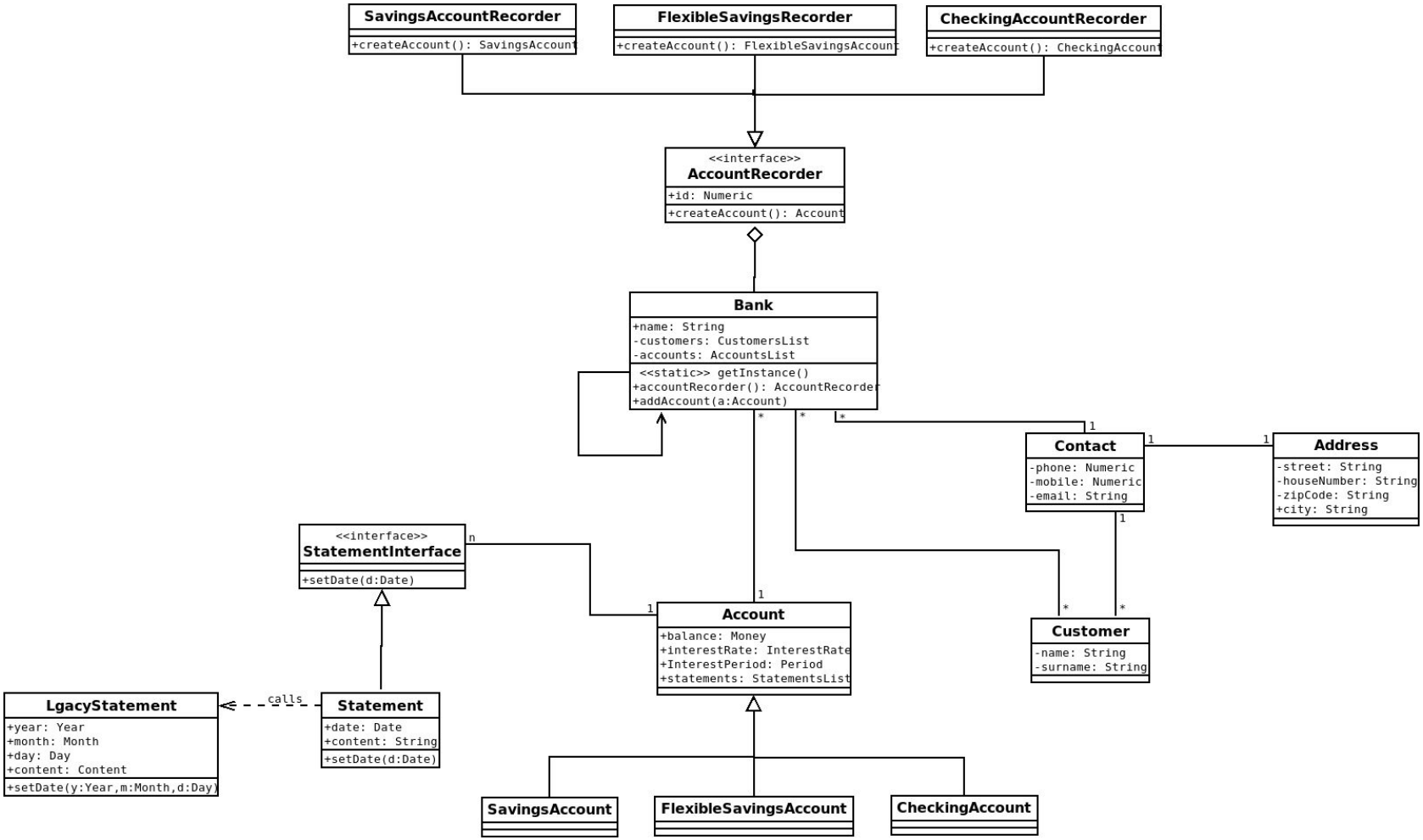
Question 6. [8 points].
VegaSoft AB, a new game development company, wants to hire your company to create a new VR (virtual reality) experience for their customers. In short, the experience consists of a new virtual puzzle game for kids between 3-5 years old. The game is played by the kids using some new VR glasses that show them a virtual world with puzzles. These glasses can be connected to any mobile device that has the new application installed, so kids can enjoy the game everywhere, and does not require any Internet connectivity.

6.a (3/8 points) *Describe* and *sketch a working diagram* of one (1) type of logical architecture for the application. Justify your answer.

6.b (3/8 points) Assume that the team has decided to use a 2-tier architecture. *Would it be reasonable* to use a 2-tier architecture in combination with the logical architecture you have chosen? *Justify* your answer with an example or a counterexample.

6.b (2/8 points) *Would it be better* to use a thin or a fat client? *Justify* your answer.

CONTINUES [UML DIAGRAM] NEXT PAGE


SavingsAccountRecorder
+createAccount(): SavingsAccount
FlexibleSavingsRecorder
+createAccount(): FlexibleSavingsAccount
CheckingAccountRecorder
+createAccount(): CheckingAccount
<<interface>>
AccountRecorder
+id: Numeric
+createAccount(): Account
Bank
+name: String
-customers: CustomersList
-accounts: AccountsList
<<static>> getInstance()
+accountRecorder(): AccountRecorder
+addAccount(a:Account)
Contact
-phone: Numeric
-mobile: Numeric
-email: String
Address
-street: String
-houseNumber: String
-zipCode: String
+city: String
<<interface>>
StatementInterface
+setDate(d:Date)
Account
+balance: Money
+interestRate: InterestRate
+InterestPeriod: Period
+statements: StatementsList
Customer
-name: String
-surname: String
LgacyStatement
+year: Year
+month: Month
+day: Day
+content: Content
+setDate(y:Year,m:Month,d:Day)
calls
Statement
+date: Date
+content: String
+setDate(d:Date)
SavingsAccount
FlexibleSavingsAccount
CheckingAccount