17-423/723 Recitation: Design for Testability

Feb 14, 2025

Andrew IDs:

Consider a feature in an online shopping site that allows a customer to pay for an item in multiple installments over a period of time. For example, an item that costs $1000 can be paid in 4 installments of $250, with payments being made 1 month apart (e.g., Feb 14, March 15, April 15, May 15). The following class, **InstallmentGenerate**, is used to generate a series of installments and store them into a database.

```
public class InstallmentGenerator {

  // Handle to installment database
  private InstallmentDB db;

  public InstallmentGenerator(String dbAddress) {
     // Connect to the installment database at “dbAddress”
     this.db = InstallmentDB.connect(dbAddress);
  }

   public void generateInstallments(ShoppingCart cart, int numberOfInstallments) {
      // First installment begins today
      LocalDate nextInstallmentDueDate = LocalDate.now();

      // Calculate the amount per each installment
      double amountPerInstallment = cart.getValue() / numberOfInstallments;

      for(int i = 1; i <= numberOfInstallments; i++) {
         // Every installment is 30 days apart
         nextInstallmentDueDate = nextInstallmentDueDate.plusMonths(1);
         Installment newInstallment =
            new Installment(nextInstallmentDueDate, amountPerInstallment);
         // Store each installment in the database
         db.persist(newInstallment);
      }
   }
}
```

Q1. What are different test cases that should be considered to test the correctness of the installment generation feature?

Q2. List one controllability challenge and one observability challenge in testing this component.

Q3. What modifications would you make to the design of the component to improve its testability?

Q4. What are pre-conditions and post-conditions for the method **generateInstallments**?