AJHS 05 December 2018

Group Design Project - Feedback

Team XX= 70

Split credit.

- A.Surname @ 30% = **77**
- B.Person @ 30% = **77**
- C.Student @ 20% = **63**
- D.Worker @ 20% = **63**

[40] UML Design Accuracy = 26

- Use Case Diagram – good: UC11
- Information Model – poor: IM1, IM2, IM3, IM6, IM8, IM11, IM13
- Normalised Data Model – average: DM3, DM4, DM9, DM11
- GUI State Machine – excellent: SM1, SM12

[50] System Design Quality = 38

Functional Behaviour

Most system functions worked as expected. Screenshots showed strange level and credit information that should have been data-validated.

Secure Role-based Operation

Secure login with hashed and salted password storage. SQL prepared statements used, but note that this alone does not prevent SQL injection, unless you also do data sanitization. Used MVC to load different screen controllers for each role (great idea).

User Interface Quality

Attractive design for screens. One tester said that your system worked better than you reckoned yourselves!

[10] Team Working = 6

Seem to have delegated the work OK and collaborated on coding, but not so much on design. Your initial information model was actually a DB model (incomplete) that was unrelated to your final DB model (much better), so please check that you understand what an Information Model should look like. You needed to cross-check your design work more.

Feedback Comments

Please see the accompanying feedback comment-bank to decode the meaning of the feedback codes that applied to your UML designs. Free-text comments from your testers have been edited above.

UML Feedback Comment Bank

Check your feedback-keys against the remarks in this comment-bank.

Use Case Diagram

- UC1 - You missed some actors (Administrator, Registrar, Teacher, Student).
- UC2 - You missed some use cases (see list of tasks in the project brief).
- UC3 - You added extra use cases (too much implementation detail).
- UC4 - You missed some participations between actors and use cases; or added extra unnecessary participations (most use cases involve only one actor).
- UC5 - You missed or misplaced the system boundary (all use cases are part of the system; actors are external to the system).
- UC6 - You misused or overused the «include» dependency (it's not for arbitrary subroutine calls, but for showing shared flows that are part of several use cases).
- UC7 - You misused or overused the «extend» dependency (it's not for arbitrary branching choices, but for a significant optional insertion into the flow of a use case).
- UC8 - You misused a generalisation arrow (created too generic use cases; or related the wrong concepts; or used the arrow to indicate flow).
- UC9 – You used the wrong kind of line to link actors with use cases (should be a simple line, an association, not an arrow).
- UC10 – Concrete use cases were still somewhat abstract (e.g. "manage X" rather than "add X", "remove X", "update X").
- UC11 – (Some) use cases were of the wrong granularity (either: too large, standing for many tasks; or too small, standing for low-level actions with no business value outcome)
- UC12 – Notation error: actors (should be stick-figure); or use cases (should be ellipse) were drawn using the wrong notation; or some other bending of the notation rules.

Information Model

- IM1 – You missed some modelled domain entities (User, Student, Department, Degree, Module).
- IM2 – You missed some pure information entities (Level, Period), or encoded these wrongly as attributes in other entities.
- IM3 – You missed some needed association classes (Teaching, Approval, Enrolment) to record attributes of associations.
- IM4 – You added extra entities that had no structure at all (no attributes) in the information model (Administrator, Registrar, Teacher).
- IM5 – You included extra implementation classes for managers modelling the processing in your system (not part of the information model), or wrongly included such operations in the information concepts.
- IM6 – You missed one or more required associations (Degree-Department, Degree-Student, Degree-Module, Student-Period, Period-Module, Period-Level, Level-Approval).

- IM7 – Some multiplicities were wrong (e.g. one instead of many; or vice-versa) or missing.
- IM8 – You missed some required attributes (see the scenario again, e.g. to capture a degree requiring a year in industry; or the lead department teaching a degree; or user login and password; or pass and resit grades for one enrolment on a module).
- IM9 – You inserted extra attributes that were derived properties and should be operations (e.g. *passed* instead of *canProgress()* based on grades)
- IM10 – You included both an attribute and an association representing the same thing (you should have either one, or the other).
- IM11 – You missed the generalisation relationship (Student inherits from User), or shared attributes wrongly across this relationship.
- IM12 – You had the wrong kinds of operations (either: get-set pairs which don't appear in the IM; or did not restrict operations to computing derived values); or you misused the syntax in some way.
- IM13 – You did not include any/enough operations for mentioned derived properties, such as moduleResult(), levelResult(), degreeClass().
- IM14 - You included unnecessary aggregation relationships; or composition relationships for parts not wholly contained in their whole.

Database Model

- DM1 – You forgot to include the «table» or «view» labels identifying entities in your model as database tables (or did not use the UML database profile correctly).
- DM2 – You forgot to identify a PK for each of your tables.
- DM3 – You forgot to use compound PKs for some tables (Period, and PFKs for all former association classes that were promoted to linker tables: Teaching, Approval, Enrolment).
- DM4 – You invented spurious ID attributes as surrogate PKs when these were not needed (this bloats the data unnecessarily).
- DM5 – You forgot to identify a FK for each related table (the arrows indicate which table was related), or put in FKs for unrelated tables.
- DM6 – You failed to eliminate all many-to-many or one-to-one associations in your normalised data model.
- DM7 – You did not successfully promote (all) many-to-many associations (and association classes) to linker tables, or linked the wrong things with your linkers.
- DM8 – Your arrows did not always link the many-side to the one-side (or you did not use the UML database profile correctly).
- DM9 – You included an extra table where one was not strictly needed (e.g. Role, Tutor).
- DM10 – You had missing tables or attributes resulting from gaps in your earlier Information Model (e.g. StudyLevel, PeriodOfStudy, Teaching; e.g. yearInIndustry:Boolean, core:Boolean, leadDept:Boolean)
- DM11 – Some attributes were in the wrong table, e.g. should have Module(credits:Integer), Teaching(leadDept:Boolean)
- DM12 – You had missing, or misdirected associations (with FKs) to some related tables; or you included too many redundant associations.
- DM13 – You mixed up your PK and PFK keys.

- DM14 - When promoting association classes to linker tables, you got the many- and one-multiplicities the wrong way round (each linker relates to exactly one of each related table).

State Machine Diagram

- SM1 – You did not have an initial entry transition and/or a final exit transition when switching off.
- SM2 – You did not have a *LoggedIn* superstate, within which all the remaining system behaviour is nested, supporting a single *logout* operation (but it is OK to have separate role-based superstates with *logout* instead).
- SM3 – Your nested state machines did not all have an initial default entry transition.
- SM4 – You did not use superstates to separate the logged-in views and functions for each role; or you missed one or more of the roles.
- SM5 – Your transitions for logging in did not have guards which were mutually exclusive and exhaustive for the success and failure cases.
- SM6 – Your transitions for entering the state representing role-based views did not have guards that were mutually exclusive and exhaustive.
- SM7 – You missed transitions, or their event-labels, for some of the operations actually performed (check the scenario – some transitions loop back to the current view).
- SM8 – You included things that weren't states, or transitions that weren't triggered, or had states representing activity without showing the transition events.
- SM9 – Your state machine halted, or was blocked at some point (e.g. cannot logout); or it allowed an operation it should not (e.g. login when switched off)
- SM10 – Your transitions to superstates crossed the superstate boundary and/or targeted, or exited, substates directly (broke encapsulation of superstsate).
- SM11 – You used sequences of states, rather than nested states, to show the activities that were carried out when in a particular role (use a superstate per role).
- SM12 – You did not always clearly distinguish Boolean guards (text in square brackets) from triggering events (plain text labels).
- SM13 – You used the wrong notation (e.g. states and superstates are rounded rectangles, transitions are open-headed arrows)
- SM14 - You included extra transitions in substate machines that were already handled by your superstate machine.