

Laborator 5

Cerinta: Pornind de la principalele elemente specifice diagramelor UML, descrieti pe scurt exemplele propuse. Cautati alte diagrame specifice fiecarui tip de diagrama. Analizati modul de generare considerand diferite limbaje si medii de programare.

Diagrame de clasa

Intr-o diagrama de clasa entitatile sunt reprezentate sub forma unui dreptunghi cu 3 compartimente ce contin, numele entitatii, atributele si metodele acesteia. In diagrama de clasa prezentata in lucrarea de laborator avem mai multe entitati: Student, Enrollment, Seminar, Professor, Address, Course, Person.

Din prima diagrama prezentata putem sa intelegem ca 1 student poate fi la 1 sau mai multe Enrollment. Un seminar poate sa apartina la unu sau mai multe Enrollment. Pe lista de asteptare sunt doua cazuri, Primul in care nu mai exista nici un student care sa doreasca sa participe la seminar, sau nu mai sunt seminar, dar mai sunt studenti pe lista de asteptare. Dupa diagrama de asemenea putem intelege ca pot sa existe seminar la care sa nu fie nici un profesor, iar de asemenea pot sa existe mai multi profesori decat seminar.

In a doua diagrama este prezentata relatia dintre student si adresa. Aceasta relatie este de 1 la 1, dar aici am fi putut pune o relatie de 1..n la 1 deoarece pot sa existe mai multi studenti care sa locuiasca la aceeasi adresa (de exemplu frati)

Urmatoarea diagrama prezinta relatia dintre seminar si curs. Din diagrama putem intelege ca fiecare seminar are un curs, dar se poate sa nu existe nici un seminar, ceea ce implica ca se poate sa nu existe nici un curs.

Ultima diagrama reprezinta clasa Person (pe care am putea sa o numim interfata deoarece din ea va mosteni si Student si Professor). se poate de asemenea observa ca se poate sa nu existe nici o persoana dar sa avem o adresa salvata.

Generarea de cod va fi diferita pentru fiecare limbaj de programare deoarece fiecare limbaj este diferit.

Diagrame de stare

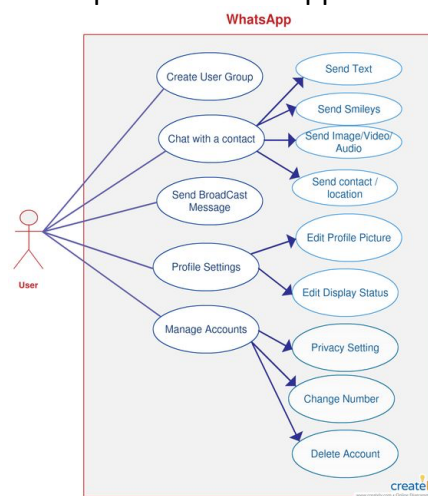
In diagrama de stare este prezentat mecanismul de inscriere al unui student la facultate. Un student va trece din starea de inceput (bila neagra din stunda) si poate ajunge in doua stari diferite. Poate sa fie aprobat sau anulat. In a doua diagrama se va prelunge prima diagrama iar pe langa intrarea la facultate se arata si diagrama de stare pentru participarea la cursuri si examen. Nu vor exista diferente majore in implementare deoarece diagrama de stare arata mai mult flow-ul de executie care trebuie sa fie respectat si arata cum trebuie sa se intample lucrurile. Diagrama nu depinde catusi de putin de limbajul de programare

Diagrame de activitati

Diagrama de activitati arata tot mecanismul de admitere la facultate. Se incepe prin completarea formularului de admitere, daca formularul este corect se trece la pasul urmator, daca nu ne intoarcem la pasul de completare de formular. Daca datele au fost corecte se introduc informarii despre candidatul. Data candidatului nu este admis, tot procesul se opreste. Daca candidatul este admis, atunci verificam daca este admis cu sau fara taxa. Daca este admis cu taxa trebuie sa plateasca taxa iar apoi sa se inscrie, daca nu are taxa atunci sarim direct la pasul de inscriere. De asemenea diagrama nu depinde de limbajul de programare.

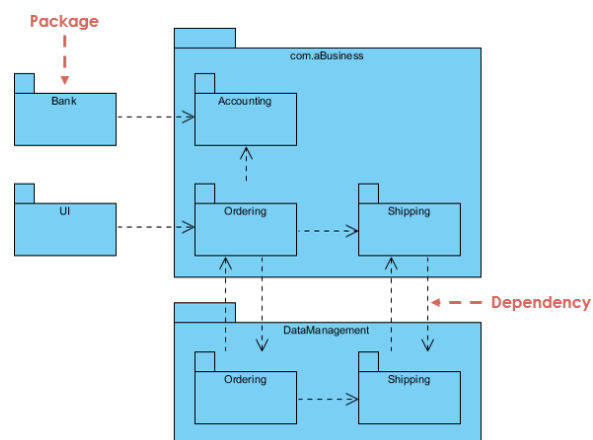
Diagrame de folosire

Exemplu de diagrama de folosire a aplicatiei WhatsApp.

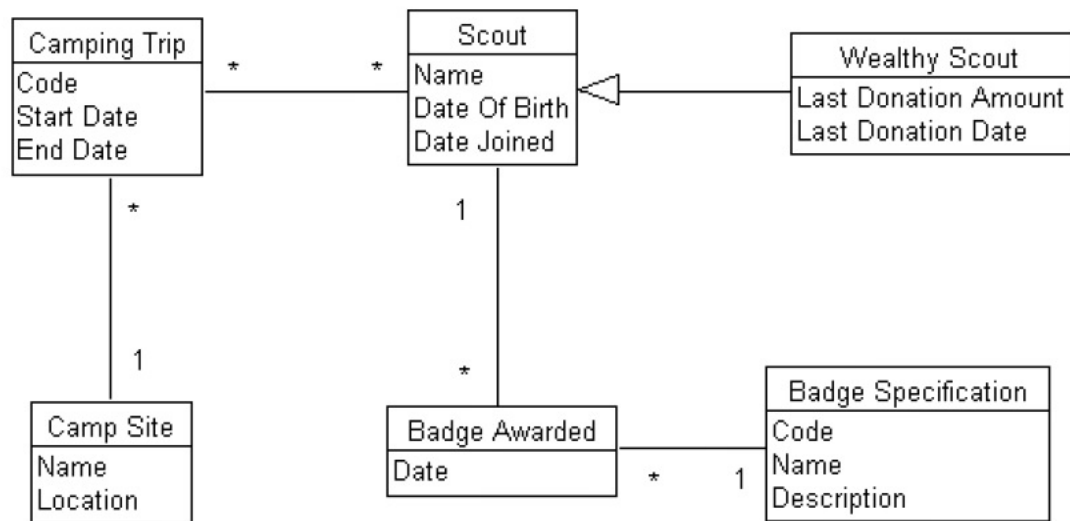


In use case diagram avem unul sau mai multi actori care fac actiuni. Aceste actiuni pot duce la alte actiuni si tot asa mai departe. De asemenea use case diagram nu depinde de limbajul de programare

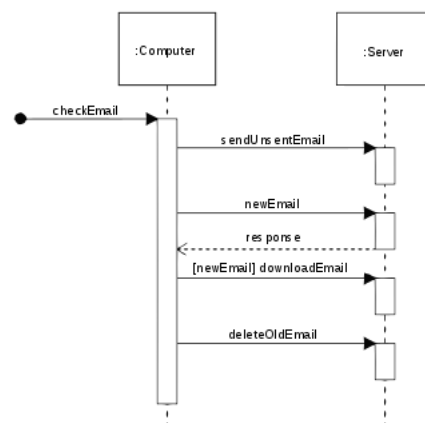
Package Diagram



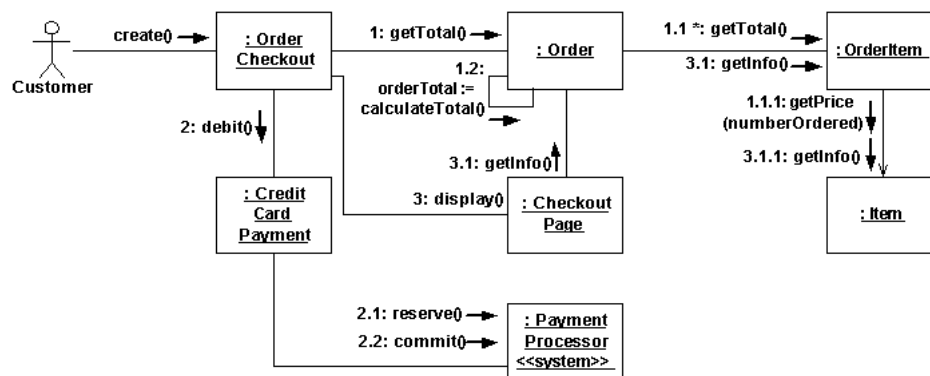
Object Diagram



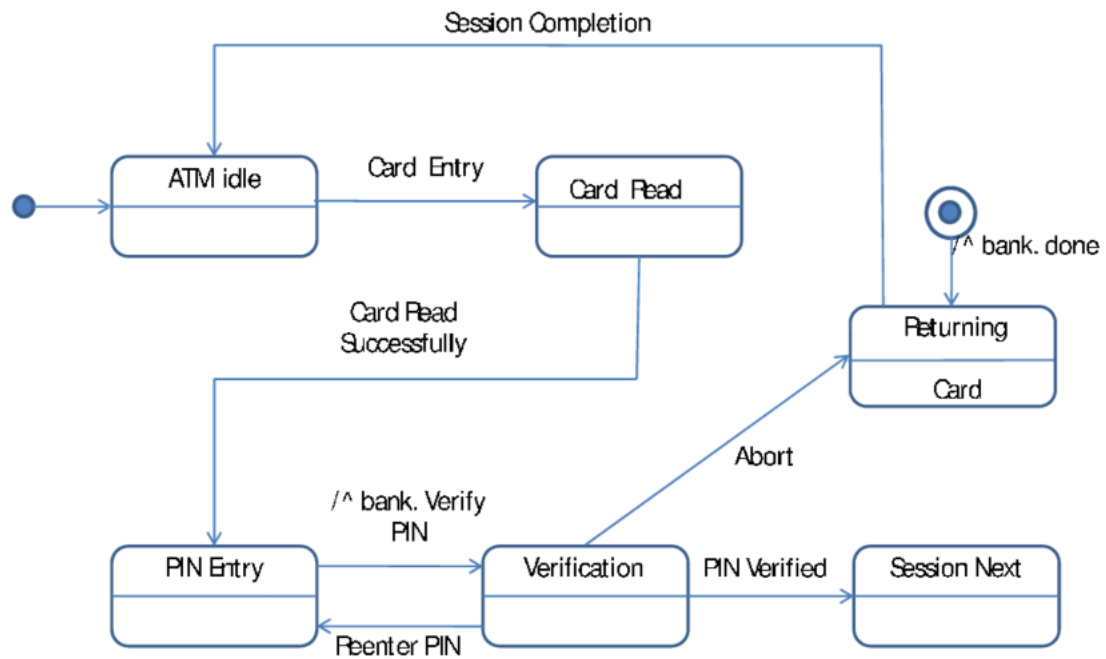
Sequence Diagram



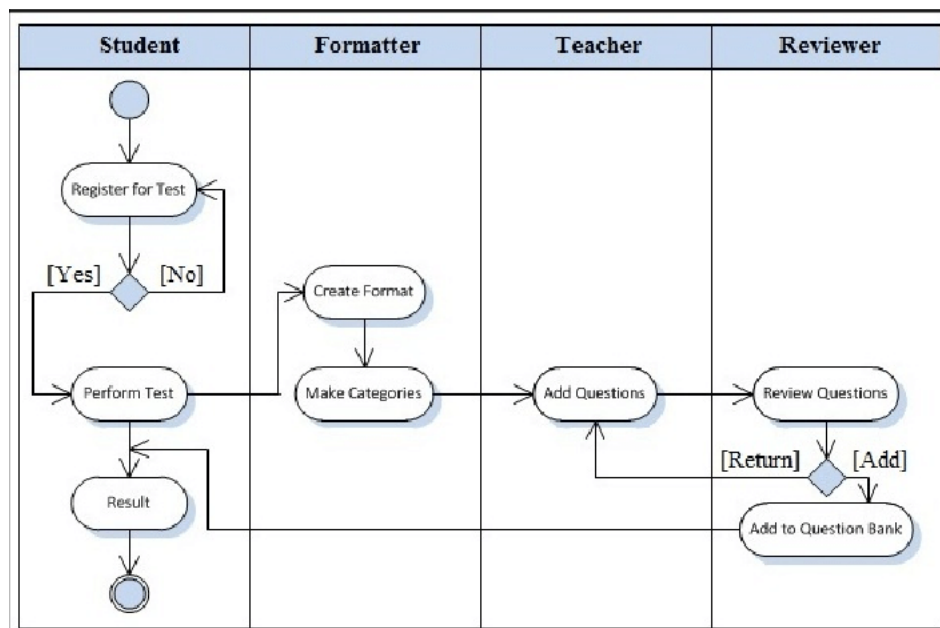
Collaboration Diagram



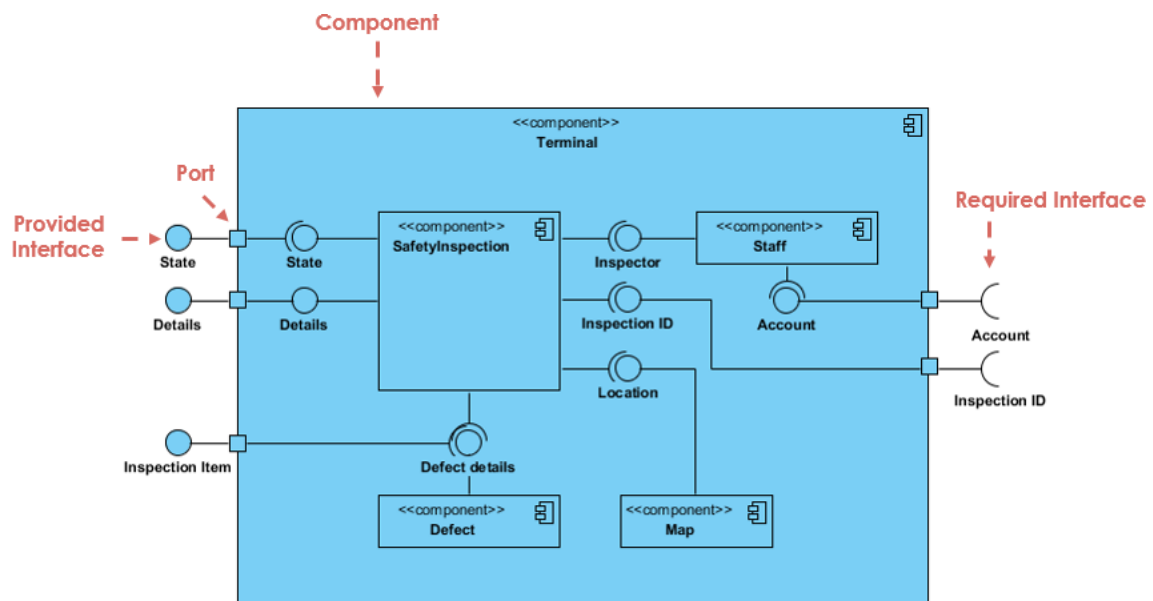
Statechart Diagram



Activity Diagram



Component Diagram



Deployment Diagram

