Assignment 2
Example Business Blueprint

Assume the "Domain" and "Salient Stakeholder Qualities" sections below represent an excerpt from an Assignment 1 submission.

Domain: Course Registration

Roles

- *Consumer*
  - *Advisor*
    - *Faculty*
    - *Advising Staff*
  - *Student*
- *Producer*
  - *Developer*
  - *Architect*
  - *Tester*
  - *Requirements Engineer*

Functions

- *Approve student schedule:* Review proposed schedule to ensure student has prerequisites and courses apply to program of work.
  - Inputs:
    - *Registration authorization (transient data from "Verify student registration status")*
    - *Proposed schedule (from external)*
    - *Course catalog*
    - *Student master record*
  - Outputs:
    - *Advisor approval form (transient data to "Process add requests" or "Process delete requests")*
  - Performer: *"Advisor" stakeholder role (any faculty member or designated department staff that are authorized to advise students.)*
  - Location/Environment: *Student records cannot be taken off campus; otherwise, no restrictions.*
- *Process add requests:* Add a list of classes to the student's schedule, if not full. If a class is full, place the student on the waiting list.
  - Inputs:
    - *Advisor approval form (transient data from "Approve student schedule")*
    - *Proposed schedule(from "Approve student schedule")*
    - *Student master record*
    - *Course roll*

        - *Course waitlist*
    - Outputs:
        - *Student master record (updated)*
        - *Course roll (updated)*
        - *Course waitlist (updated)*
    - Performer: *Anyone with permission to access student records (i.e., anyone in "Consumer" role, as long as he/she has the approval form)*
    - Location/Environment: *Student records cannot be taken off campus; otherwise, no restrictions.*

- *Process drop requests:* Drop one or more courses from the student's schedule.
    - Inputs:
        - *Advisor approval form (transient data from "Approve student schedule")*
        - *Proposed schedule(from "Approve student schedule")*
        - *Student master record*
        - *Course roll*
    - Outputs:
        - *Student master record (updated)*
        - *Course roll (updated)*
    - Performer: *Anyone with permission to access student records (i.e., anyone in "Consumer" role, as long as he/she has the approval form)*
    - Location/Environment: *Student records cannot be taken off campus; otherwise, no restrictions.*

- *Update student profile:* Update contact info and related information.
    - Inputs:
        - *Verified student id (transient data from "Verify enrollment")*
        - *Student master record*
        - *Profile updates (from external)*
    - Outputs:
        - *Student master record (updated)*
    - Performer: *Anyone with permission to access student records (i.e., anyone in "Consumer" role, as long as "Verified student id" received from "Verify enrollment")*
    - Location/Environment: *Student records cannot be taken off campus; otherwise, no restrictions.*

- *Verify enrollment:* Ensure student is enrolled at the university.
    - Inputs:
        - *Student id (from external)*
        - *Matriculation file*
    - Outputs:
        - *Verified student id (transient data to "Update student profile" or "Verify student registration status")*
    - Performer: *Anyone in "Consumer" role*
    - Location/Environment: *Student records cannot be taken off campus; otherwise, no restrictions.*

- *Verify student registration status:* Ensure student is currently permitted to register given major and classification.
    - Inputs:
        - *Verified student id (transient data from "Verify enrollment")*
        - *Student master record*
    - Outputs:
        - *Registration authorization (transient data to "Approve student schedule")*
    - Performer: *Anyone with permission to access student records (i.e., anyone in "Consumer" role, as long as "Verified student id" received from "Verify enrollment")*
    - Location/Environment: *Student records cannot be taken off campus; otherwise, no restrictions.*

Scenarios

- *Register for semester courses*
    1. Verify enrollment
    2. Update student profile
    3. Verify student registration status
    4. Approve student schedule
    5. Process add requests

- *Reduce semester load*
    1. Verify enrollment
    2. Update student profile
    3. Verify student registration status
    4. Approve student schedule
    5. Process drop requests

Salient Stakeholder Qualities

Quality	**Description**
Cost	Budget for course registration implementation and maintenance is being obtained by reducing budget allocated to other proposed projects.
Extensibility	While the essential registration process remains relatively constant, catalog rules and major programs of work evolve
Performance	10K students must register within a two-week period
Consistent student master record	Students, staff, and faculty need to see consistent student info

Notes

- All are considered important, but clearly there are tradeoffs between cost and extensibility and performance.

Business Blueprint (BB)

Allocation of functions and data to components

Component	**Functions and Data**
Authenticator	Functions: • Verify enrollment Data: • Matriculation file
Advisor	Functions: • Approve student schedule Data: • Course catalog
Registrar	Functions: • Process add requests • Process drop requests • Update student profile • Verify student registration status Data: • Student master record • Course roll • Course waitlist

Component-to-Component I/O Dependencies

Components	**Functions and Data**
FROM: Authenticator TO: Registrar	• *Update student profile* requires *Verified student id* from *Verify enrollment* • *Verify student registration status* requires *Verified student id* from *Verify enrollment*
FROM: Advisor TO: Registrar	• *Process add requests* requires *Advisor approval form* from *Approve student schedule* • *Process drop requests* requires *Advisor approval form* from *Approve student schedule* • *Process add requests* requires *Proposed schedule* from *Approve student schedule* • *Process drop requests* requires *Proposed schedule* from *Approve student schedule*
FROM: Registrar TO: Advisor	• *Approve student schedule* requires *Registration authorization* from *Verify student registration status* • *Approve student schedule* requires data *Student master record*, which was allocated to component *Registrar*

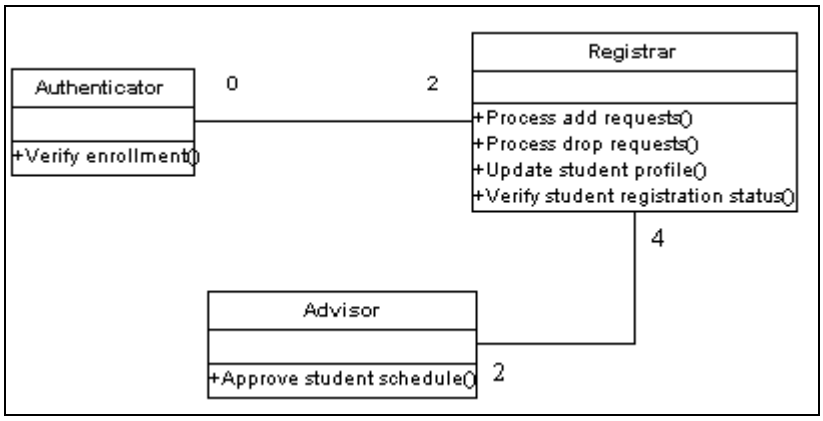

Figure 1: UML-based depiction of BB

Notes

- Some inputs are transient, such as *Verified student ID, Registration authorization,* and *Advisor approval form*. These are not "owned" by any component.
- The counts depicted on the BB diagram are *dependencies between components* (e.g., *Update student profile* and *Verify student registration status* both require *Verified student id* from *Verify enrollment*).
    - Each count represents the number of inputs needed by the adjacent component.
    - The counts do not includes inputs from external sources.
    - The counts can be determined directly from the "Component-to-Component I/O Dependencies" table.
- The diagram uses the UML class diagram syntax. However, for our purposes, we're interpreting the classes as BB components and the class-to-class associations as component-to-component input dependencies.