Shopper

<<Interface>>
Shopper

+ void createGroceryList()
+ boolean addItemToGroceryList(String itemName, int quantity, GroceryStore store)
+ void viewGroceryList()
+ boolean removeItemFromGroceryList(String itemName)
+ List<GroceryItemOrder> getGroceryListItems()
+ double getTotalGroceryListCost()

GroceryStore

- String name
- Map<String, Double> inventory

+ GroceryStore(String name)
+ void addItem(GroceryItem item)
+ void removeItem(String itemName)
+ void updatePrice(String itemName, double newPrice)
+ double getPrice(String itemName)
+ GroceryItemOrder getOrder(String itemName, int quantity)
+ String toString()

GroceryList

- static final int MAX_ITEMS = 10 // max items in a list
- GroceryItemOrder[] items = new GroceryItemOrder[MAX_ITEMS]
- int itemCount = 0

+ GroceryList()
+ void add(GroceryItemOrder item)
+ List<GroceryItemOrder> getItems()
+ void remove(GroceryItemOrder item)
+ boolean removeItem(String itemName)
+ double getTotalCost()
+ String toString()

Registry

- List<Student> students = ArrayList<>()
- List<Course> courses = ArrayList<>()
- List<Professor> professors = ArrayList<>()
- List<Semester> semesters = ArrayList<>()
- int nextStudentId = 10001 // Starting ID for Students
- int nextProfessorId = 20001 // Starting ID for Professors

+ Registry()
+ void initializeSemesters()
+ void initializeCourses()
+ List<Course> getAllCourses()
+ int addStudent(Student student)
+ void addCourse(Course course)
+ int addProfessor(Professor professor)
+ Professor findProfessorByName(String name)
+ Student findStudentByName(String name)
+ boolean enrollStudent(Student student, Course course, Semester semester)
+ void dropStudent(Student student, Course course)
+ List<Course> getCoursesBySemesterAndSemester(String level, Semester semester)
+ List<Course> getCoursesByProfessor(Professor professor)
+ Course findCourseByNumberAndSemester(String courseNumber, String semester)
+ List<Course> getCoursesBySemester(String semester)
+ String toString()
+ Iterator<Student> iterator()

Person

- String firstName
- String lastName
- GroceryList groceryList // everyone can have a list
- String role

+ Person(String firstName, String lastName, String role)
+ String getFirstName()
+ void setFirstName(String firstName)
+ String getLastName()
+ void setLastName(String lastName)
+ String getRole
+ void setRole(String role)
+ String getFullName()
+ GroceryList getGroceryList()
+ void createGroceryList()
+ boolean addItemToGroceryList(String itemName, int quantity, GroceryStore store)
+ boolean removeItemFromGroceryList(String itemName)
+ void viewGroceryList()
+ List<GroceryItemOrder> getGroceryListItems()
+ double getTotalGroceryListCost()
+ String toString()

GroceryItem

- String name
- double pricePerUnit

+ GroceryItem(String name, double pricePerUnit)
+ String getName()
+ double getPricePerUnit()
+ void setName(String name)
+ void setPricePerUnit(double pricePerUnit)
+ String toString()

GroceryItemOrder

- GroceryItem item
- int quantity

+ GroceryItemOrder(GroceryItem item, int quantity)
+ double getCost()
+ GroceryItem getItem()
+ int getQuantity()
+ void setQuantity(int quantity)
+ String getName()
+ String toString()

Professor

- String department
- int id
- List<Course> coursesTaught

+ Professor(String firstName, String lastName, String department)
+ String getDepartment()
+ void setDepartment(String department)
+ void setId(int id)
+ int getId()
+ void addCourse(Course course)
+ void removeCourse(Course course)
+ List<Course> getCoursesTaught()
+ List<Course> getCoursesForSemester(String Semester)
+ String toString()

Student

- int id
- String studentType
- List<Course> registeredCourses = new ArrayList()
- List<Course> waitListedCourses = new ArrayList()
- GroceryList groceryList

+ Student(String firstName, String lastName, String studentType)
+ int getId()
+ void setId(int id)
+ String getStudentType()
+ void setStudentType(String studentType)
+ void addRegisteredCourse(Course course)
+ boolean removeCourse(Course course)
+ void addWaitListedCourse(Course course)
+ void removeFromWaitList(Course course)
+ List<Course> getRegisteredCourses()
+ List<Course> getWaitListedCourses()
+ void createGroceryList()
+ booelean addItemToGroceryList(String itemName, int quantity, GroceryStore store)
+ void viewGroceryList()
+ String toString()

Course

- int creditHours
- String courseNumber
- String courseTitle
- int maxEnrollment
- Set<Student> enrolledStudents
- LinkedList<Student> waitList
- String semester
- Professor professor
- String level

+ Course(int creditHours, String courseNumber, String courseTitle, int maxEnrollment, String semester, Professor, professor, String level)
+ int getMaxEnrollment()
+ String getSemester()
+ void setSemester(String semester)
+ Professor getProfessor()
+ void setProfessor(Professor professor)
+ String getLevel()
+ String getCourseNumber()
+ void setLevel(String level)
+ String getCourseTitle()
+ boolean enrollStudent(Student student)
+ void removeStudent(Student student)
+ void checkWaitlist()
+ Set<Student> getEnrolledStudents()
+ LinkedList<Student> getWaitList()
+ int compareTo(Course o)
+ boolean equals(Object obj)
+ int hashCode()
+ String toString()

Graduate

- int year

+ Graduate(String firstName, String lastName, String studentType, int year, int id)
+ int getYear()
+ void setYear(int year)
+ double getDiscountRate()
+ String toString()

Undergrad

- int year

+ Undergrad(String firstName, String lastName, String studentType int id, int year)
+ int getYear()
+ void setYear(int year)
+ double getDiscountRate()
+ String toString()

Relationships:
- owned by
- selects store
- stocks (1..n)
- contains
- includes
- manages (1, 0..n)
- is-a
- contains
- teaches
- waitlisted for
- enrolls

Notes

Registry: Center point of our system, holds Students, Professors, and Courses.
Person: General class for anyone, adopting the role of a shopper to go grocery shopping
Professor: Teaches a Course and be assigned different Courses for each semester
Student: Can enroll in a Course and be placed on a Waitlist, if the capacity of that Course has been reached (cap set @5 for now).
Graduate: Receive special discount
Undergrad: No discount
Course: Taught by Professors, and taken by a Student, maximum capacity of 5 (classes can only have a certain amount of Students, 5 in our case)
Shopper: Implemented by Person allowing them to go grocery shopping
GroceryStore: A store where Shoppers can buy items, stocks multiple grocery items that someone can buy at a quantity.
GroceryItem: Item that can be bought at a store, each item has a price per unit and is included in orders.
GroceryList: Owned by a Shopper, contains all item orders a shopper wishes to buy.
GroceryItemOrder: Specific order of a item, detailing the quantity of an item.