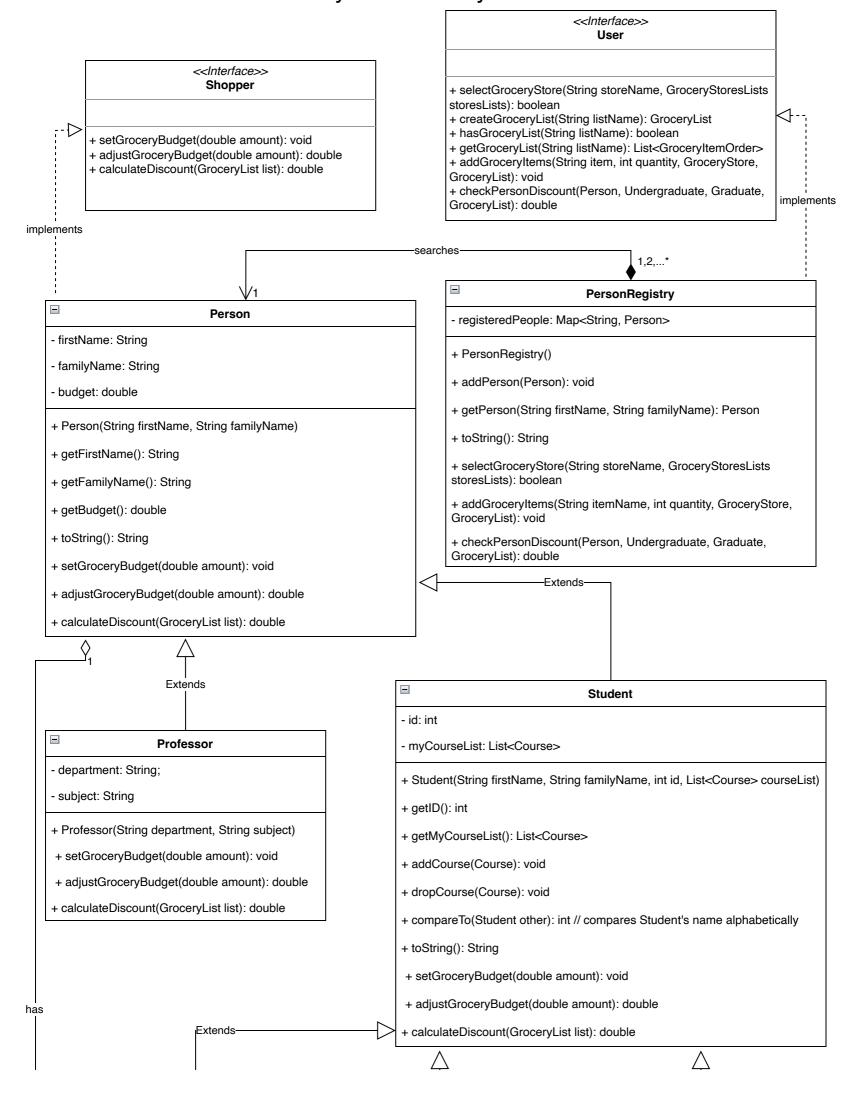
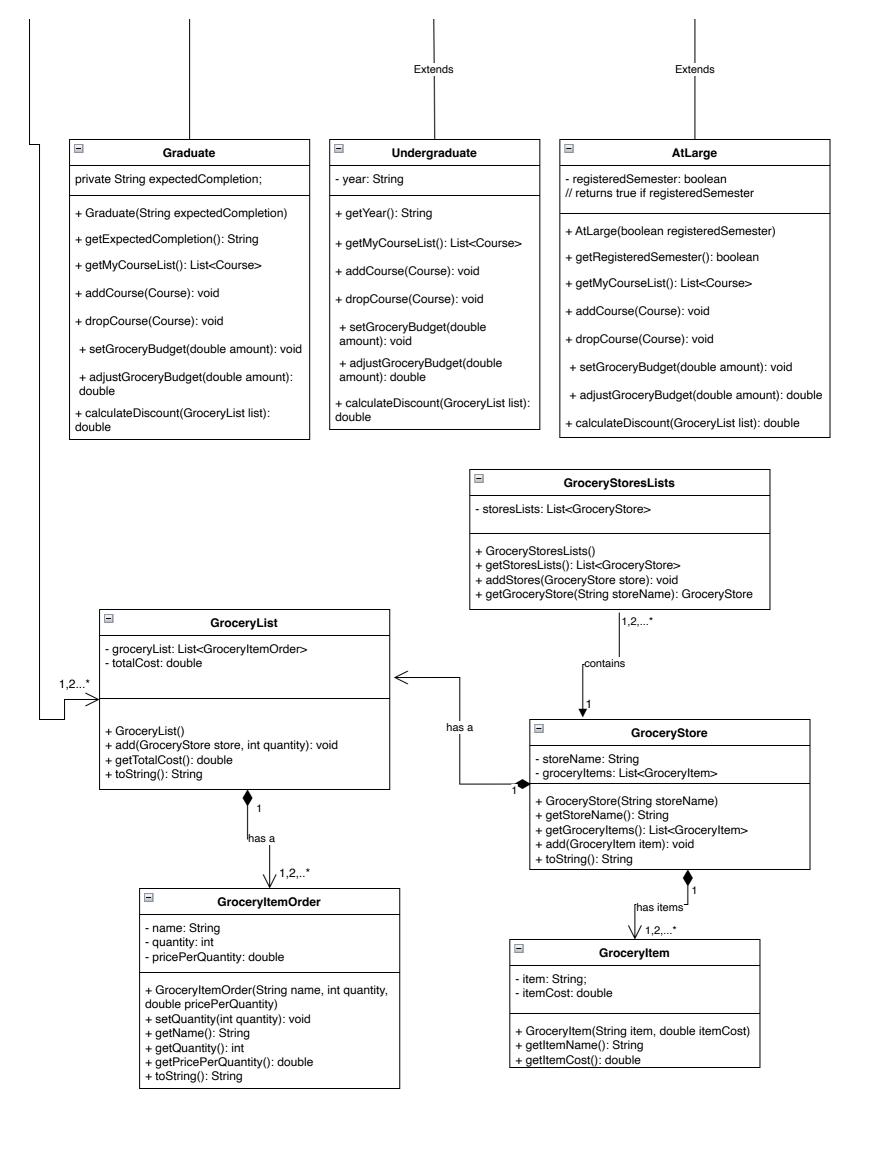


Grocery Functionality to Person





Student Course WaitList -CourseRegistry - nextStudent: Iterator<Student> // iterate over a student in a roster - nextWaitList: Iterator<Student> Course + CourseRegistry(Course) // sets the roster and waitList to a particular - department: String Iterator + findRegisteredStudents(Course): List<Student> - courseNum: int + findWaitListStudents(Course): List<Student> - courseTitle: String - courseCreditHour: int + contains(Student, Course): boolean // checks if the Student contains in a particular course - courseRoster: List<Student> + equals(Student, Student, Course): boolean // checks if two Students - courseWaitList: List<Student> are in the same section of the course - MAX_ENROLLMENT: final int // total spots left in a particular course - MAX_CAPACITY: final int // total number of courses a student can take teaches 1,2,...* + Course(department, courseNum, courseTitle, courseCreditHour) -Person + enroll(Student): boolean - firstName: String + unenroll(Person): boolean - familyName: String + toString(): String - budget: double 0,1,2,...MAX_CAPACITY + Person(String firstName, String familyName) + getFirstName(): String + getFamilyName(): String teaches + getBudget(): double takes Extends + toString(): String 1,2,3...MAX_ENROLLMENT Extends -Student _ **Professor** - id: int department: String; - myCourseList: List<Course> - subject: String + Student(int id, List<Course> courseList) + Professor(String department, String subject) + getID(): int + percentDiscount(double amount): double + getMyCourseList(): List<Course> + addCourse(Course): void + dropCourse(Course): void Extends + compareTo(Student other): int // compares Student's name alphabetically + toString(): String Extends Extends

☐ Graduate

private String expectedCompletion;

- + Graduate(String expectedCompletion)
- + getExpectedCompletion(): String
- + getMyCourseList(): List<Course>
- + addCourse(Course): void
- + dropCourse(Course): void

Undergraduate

- year: String
- + getYear(): String
- + getMyCourseList(): List<Course>
- + addCourse(Course): void
- + dropCourse(Course): void

AtLarge

- registeredSemester: boolean // returns true if registeredSemester
- + AtLarge(boolean registeredSemester)
- + getRegisteredSemester(): boolean
- + getMyCourseList(): List<Course>
- + addCourse(Course): void
- + dropCourse(Course): void