**Table of Contents**

1. Application Overview
2. System Processes (use case diagram/description) - Done
3. System Flow (activity diagram) – Done
4. Object model (class diagrams)
5. User interface mockups (menus)
6. Data structures (text file format)
7. Test cases

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Changes** | **Version** |
| Kyle Hinsz | 05/13/2014 | Initial Draft | 1.0 |
| Shamima Huq | 05/17/2014 | Overview, Test Cases, Data Structures, TOC, Revision history | 2.0 |
| David Mattox | 05/18/2014 | Added menu mockups | 2.1 |
| Ajay Chankramath | 05/18/2014 | Added object model | 3.0 |
| David Mattox | 05/18/2014 | Added file model | 4.0 |

**Technical Design**

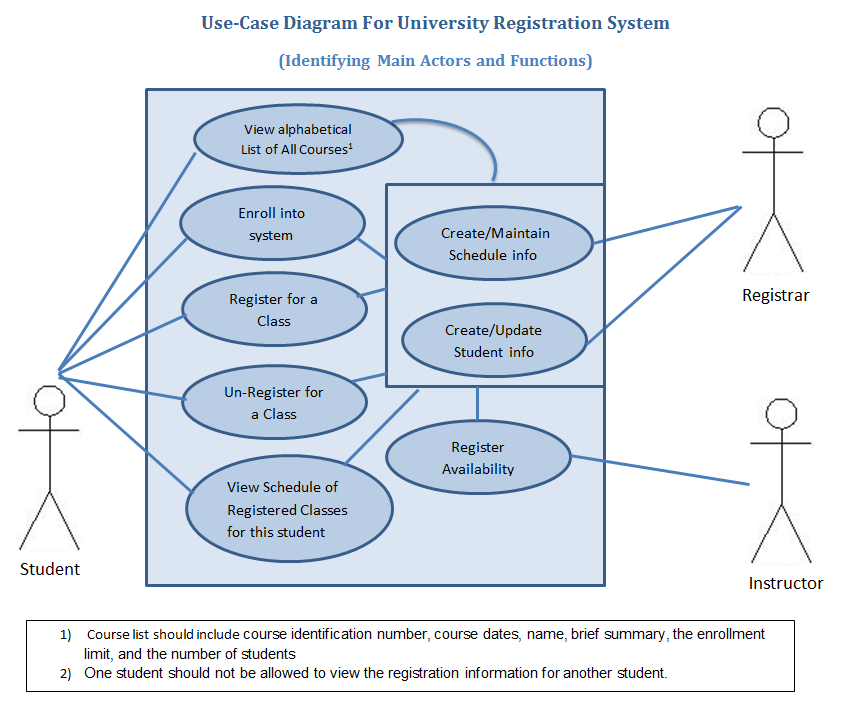
**1. Application Overview**

The application is a student registration system that displays an alphabetically ordered list of courses available for registration. The initial course list is a text file that includes the course identification number, course dates, name, brief summary, the enrollment limit, and the number of students already enrolled.

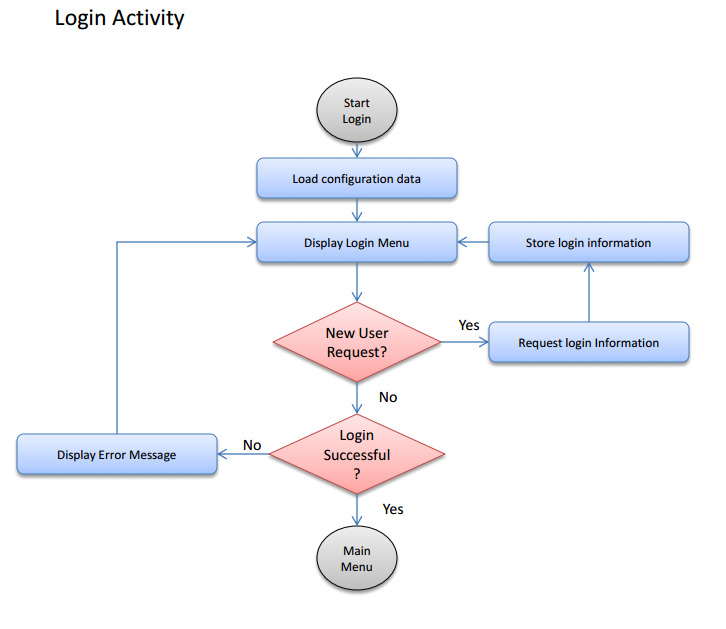
The program shows current registration numbers and available seats for each course. As students register/unregister the course’s counter of currently registered students will be adjusted. After a student successfully registers, the registration information is stored in a file and the system displays a list of classes for which he/she is currently registered.

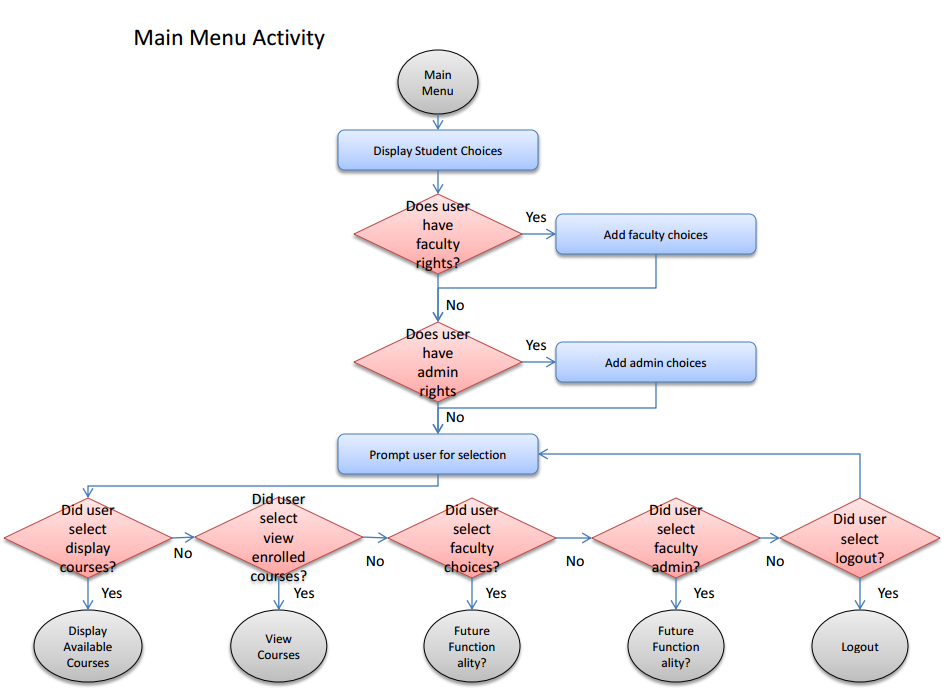
The system does not allow a student to register for a course beyond its maximum student capacity. This number is assigned in the initial text file as input. Also, before a student is able to un-register from a course, the system checks to make sure he/she was actually registered. Additionally, the system does not allow one student to view the registration information for another student.

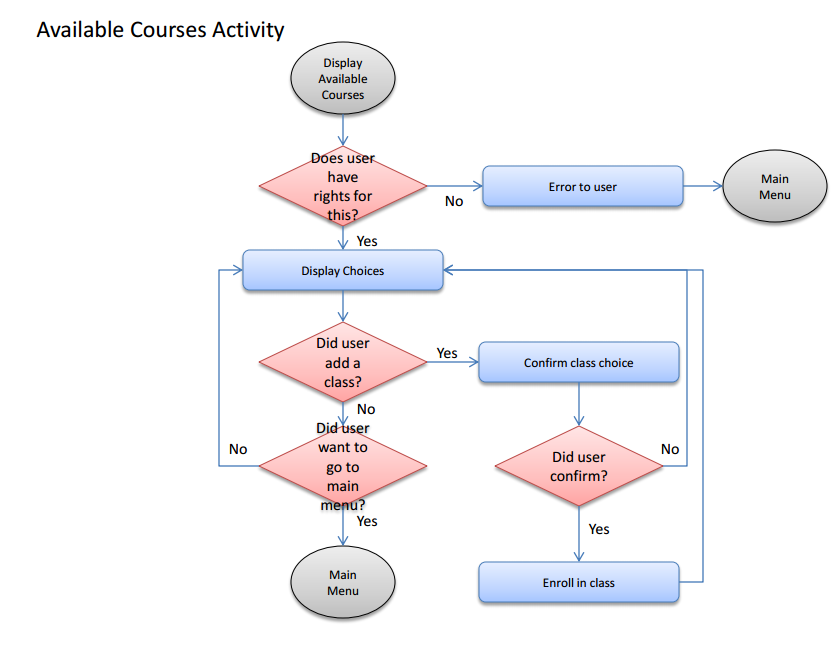
**2. System Process**

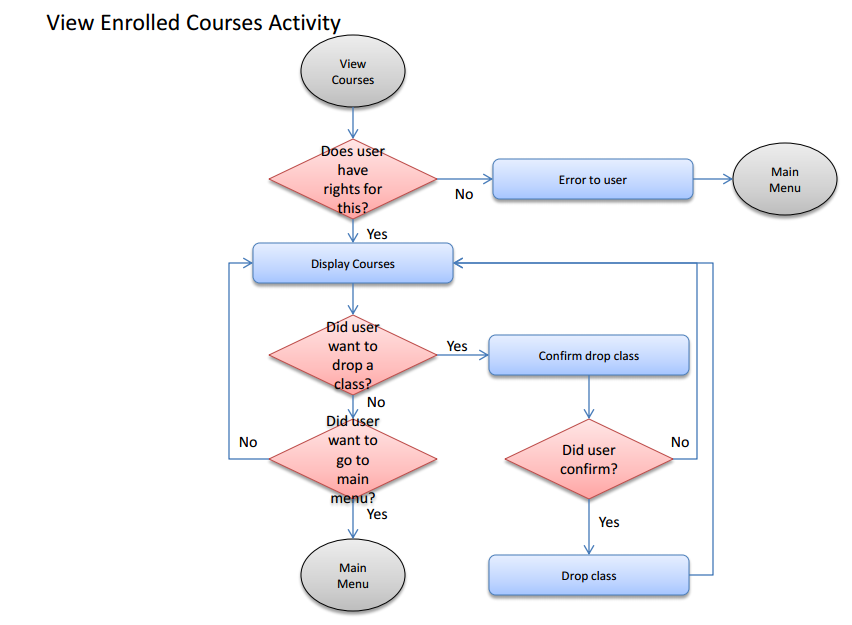


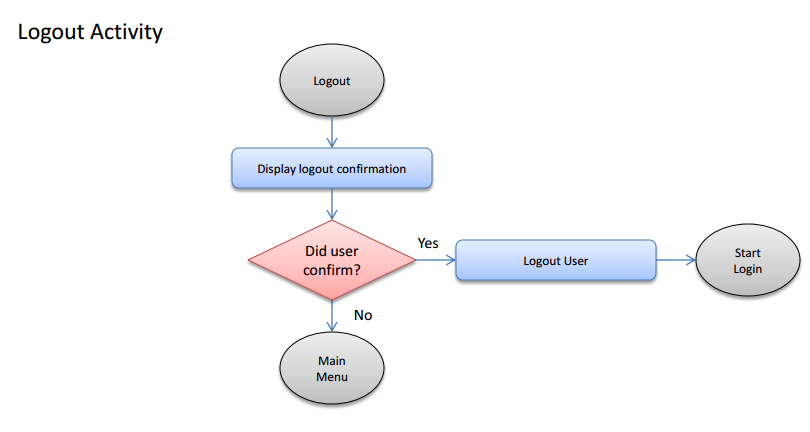
**3. System Flow**





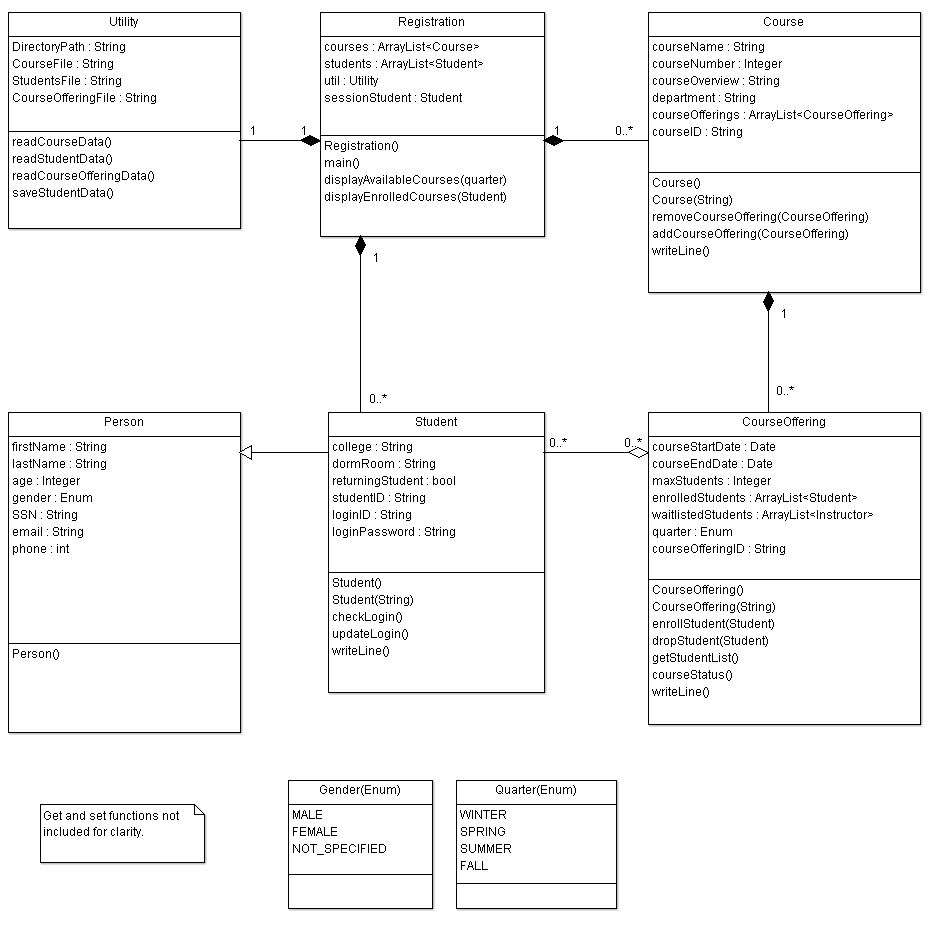






**4. Object Model**

**4.1 Class Diagram**

****



**4.2 Member Functions**

**\*All member variables will have get and set functions made available.**

**Registration Class**

Registration() : Default constructor

main() : Main function for the application

loginMenu() : displays the login menu

loginPrompt() : displays the prompt for the user ID and password

newUserPrompt() : displays the prompt for new user enrollment

mainMenuPrompt() : displays the main menu

logoutPrompt() : displays the logout prompt

displayAvailableCourses() : displays the list of available courses for the current quarter

displayEnrolledCourses() : displays the list of enrolled courses for the current student

**Course Class**

Course() : Default constructor

Course(String) : creates a instance from a string in the same format as writeLine()

removeCourseOffering(CourseOffering) : Removes a an offered course (not used for this assignment)

addCourseOffering(CourseOffering) : Adds an offered course

String writeLine() : returns a comma delimited string with the member variables of the class

**CourseOffering Class**

CourseOffering() : Default constructor

CourseOffering(String) : creates a instance from a string in the same format as writeLine()

enrollStudent(Student) : enrolls a student in the course

dropStudent(Student) : drops a student from the course

ArrayList<Student> getStudentList() : returns the list of students in the course

courseStatus() : returns the status of the course

String writeLine() : returns a comma delimited string with the member variables of the class

**Student Class**

Student() : Default constructor

Student(String) : creates a student from a string in the same format as writeLine()

Boolean : checkLogin(String userID, String password) : returns true/false if the userID and password match

updateLogin(String userID, String password) : updates the student’s user ID and password

String writeLine() : returns a comma delimited string with the member variables of the class

**5. User interface mockups**

**Login Menu:**

==================================

Welcome to the Registration System

Please select from the following:

1. Login

2. Request Access

3. Exit

Enter the number of you request: \_

**Login Prompt:**

==================================

Login

Please enter your login ID.

\_

Please enter your password.

\*

**Login Prompt Error Message:**

Unrecognized login id/password combination.

**New User Prompt:**

==================================

New user request

Please enter your student ID.

\_

Please enter your requested login ID.

\_

Please enter your password.

\*

Please confirm your password.

\*

**New User Prompt Error Messages:**

Unrecognized student ID.

Student already enrolled.

Login ID not available, please select another.

Password does not meet security requirements.

Passwords do not match.

**Main Menu:**

==================================

Main Menu

Please select from the following:

1. Display all available courses

2. Display all enrolled courses

3. Logout

Enter the number of you request: \_

**Logout Prompt:**

Please confirm you would like to logout (Y/N): \_

**Display Available Courses:**

==================================

Displaying all available courses

Please select from the following:

1. Add -> CS150 - Intro to computer programming - 50 seats available

2. Waitlist -> CS160 - Intro to Java - 25 on Waitlist

3. Add -> Advanced Java - 7 seats available

4. Main Menu

Enter the number of you request: \_

**Display Available Courses Prompts:**

Please confirm you would like to enroll in CS150 - Intro to computer programming (Y/N): \_

Please confirm you would like to be added to the waitlist for CS160 - Intro to Java (Y/N): \_

**Display Enrolled Courses:**

==================================

Displaying enrolled courses

Please select from the following:

1. Drop -> CS150 - Intro to computer programming

2. Drop -> CS160 - Intro to Java

3. Main Menu

Enter the number of you request: \_

**Display Enrolled Courses Prompt:**

Please confirm you would like to drop CS150 - Intro to computer programming (Y/N): \_

**6. Data structures**

The data file structures will be comma separated text files list of fields in which one courses data is captured on each line.

* Each field type matches the type specified in the Class Diagram
* There are no header and trailer records.
* No control information (number of records, size of files, etc.).

**6.1 Course.csv structure**

**Description:** contains the list of courses offered by the school

**Fields:** courseID, courseName, courseNumber, courseOverview, department

**6.2 CourseOffering.csv structure**

**Description:** contains the instances of courses offered by the school.

**Fields:** courseOfferingID, courseStartDate, courseEndDate, maxStudents, quarter

**6.3 StudenttoCourseMapping.csv structure**

**Description:** contains the list of enrolled/waitlisted students mapping to course offerings

**Fields:** studentID, status (“ENROLLED”/”WAITLIST”), courseOfferingID

**6.4 Student.csv structure**

**Description:** contains the list of enrolled/waitlisted students mapping to course offerings

**Fields:** studentID, firstName, lastName, age, gender, SSN, email, phone, college, dormRoom, returningStudent, loginID, loginPassword

**7. Test Cases**

Test cases will be prepared to evaluate functionality of the system as the application is developed. Initial few records have been added as students already in the system.

* Enroll into System
  + New user … create account
  + Existing user – (login pass/fail)
* Register for a Course
  + Register for a course – First time– (pass/fail)
  + Attempt to register for a course (already registered) – (pass/fail)
* Unregister for a Course
  + Unregister for a course (already registered) – (pass/fail)
  + Unregister for a course (not already registered) – (pass/fail)
* View Course List
  + View all registered courses – (pass/fail)
  + Check Alphabetic List of all Available Courses – (pass/fail)

The list below is the student text file input for the system

<Username>, <Password>, <Registered Course List>

Cecile2014, nm$rh&, MS201, SC203

SarahMAY13, Sr34%f^, CS101, MS104, SC305

Bill4456, 3654SRK@, MS403

Ly%3342, dsw@4567, MS405