

## eTeam Student Registration Program Test Cases

1. PROGRAM DATA INITIALIZATION .....	2
1.1 First run data initialization .....	2
1.2 Pristine Data State.....	3
1.3 Sample Data.....	4
2. THE GUEST INTERFACE .....	6
2.1 The Default Interface.....	6
2.2 The Guest Help Menu .....	6
2.3 Quitting the program from the Guest Interface.....	6
2.4 Course List - Guest Version .....	7
2.5 Create student account.....	7
3. THE ADMIN INTERFACE.....	9
3.1 Logging into the Admin Account.....	9
3.2 The Admin Help Menu .....	9
3.3 Quitting the program from the Admin Interface .....	9
3.4 Exiting from the Admin Interface to the Guest Interface.....	10
3.5 Account List.....	10
3.6 Course List - Admin Version.....	11
4. THE STUDENT INTERFACE .....	12
4.1 Logging into a Student Account.....	12
4.2 The Student Help Menu.....	12
4.3 Quitting the program from the Student Interface.....	12
4.4 Exiting from the Student Interface to the Guest Interface.....	13
4.5 List of registered courses .....	13
4.6 Course List - Student Version .....	14
4.7 Registering for a course.....	15
4.8 Unregistering from a course.....	18

## 1. Program data initialization

### 1.1 First run data initialization

**Test Objective:** When the program is run for the first time, the data files are created, the guest account is created, and an administrative account with username "admin" and password "password" is created.

**Preconditions:** No data files exist. Program is not running.

Step	Data	Expected Result
1. run the program	none	The following data files are created: ./accounts.dat ./courses.dat ./registrations.dat  The Guest Interface is presented .
2. Initiate login	"login"	You are prompted for a username
3. Login as Guest	"guest"	You are not prompted for a password, but are immediately presented with the Guest command prompt.
4. Initiate login	"login"	You are prompted for a username
5. Login as Admin	"admin"	you are prompted for a password
6. Provide password	"password"	The Admin interface is presented.

## 1.2 Pristine Data State

**Test Objective:** Ensure that the initial state of data for accounts, courses and registrations is correct

**Preconditions:** Data files exist, but no courses have been created, the sample data has not been loaded, and no accounts (other than Guest and Admin) have been created. Program is not running.

Step	Data	Expected Result
1. run the program	<i>none</i>	The Guest Interface is presented .
2. List courses	"courses"	An empty course list is displayed
3. Initiate login	"login"	You are prompted for a username
4. Login as Admin	"admin"	you are prompted for a password
5. Provide password	"password"	The Admin interface is presented.
6. List courses	"courses"	An empty course list is displayed
7. List accounts	"accounts"	Only the admin and guest account are displayed

### 1.3 Sample Data

**Test Objective:** Ensure that sample data loads correctly, and persists when the program exits and is re-run.

**Preconditions:** Data files exist, but no courses have been created, the sample data has not been loaded, and no accounts (other than Guest and Admin) have been created. Program is not running.

Step	Data	Expected Result
1. run the program	<i>none</i>	The Guest Interface is presented .
2. List courses	"courses"	An empty course list is displayed
3. Initiate login	"login"	You are prompted for a username
4. Login as Admin	"admin"	you are prompted for a password
5. Provide password	"password"	The Admin interface is presented.
6. Load sample data	"sample"	<i>*** finished loading sample data</i> is displayed
7. List accounts	"accounts"	In addition to the Admin and Guest accounts, dozens of student accounts are listed
8. List courses	"courses"	11 Courses are displayed. Most courses have 1 session. A few courses have 2 sessions, and 1 course has no sessions. A few of the courses are at capacity with student registrations, one of the courses is not quite full, and the rest have no enrollments.
9. Quit the program	"quit"	The program terminates
10. run the program		The Guest interface is presented.
11. List courses	"courses"	A non-empty course list is presented
12. Initiate login	"login"	You are prompted for a username
13. Login as Admin	"admin"	you are prompted for a

		password
<b>14. Provide password</b>	"password"	The Admin interface is presented.
<b>15. List accounts</b>	"accounts"	In addition to the Admin and Guest accounts, dozens of student accounts are listed
<b>16. List courses</b>	"courses"	A non-empty list of courses is presented.

## 2. The Guest Interface

### 2.1 The Default Interface

**Test Objective:** The guest interface is presented automatically when the program first starts up.

**Preconditions:** Program is not running

Step	Data	Expected Result
1. run the program	<i>none</i>	The guest help screen is displayed

### 2.2 The Guest Help Menu

**Test Objective:** Guest help screen is displayed when "help" command is issued or when an unrecognized command is issued

**Preconditions:** Guest Interface is active (2.1)

Step	Data	Expected Result
1. At the prompt, type the help command	"help"	The guest help screen is displayed
2. Type an unrecognized command	"garbage"	The guest help screen is displayed

### 2.3 Quitting the program from the Guest Interface

**Test Objective:** The program exits when you type the quit command.

**Preconditions:** Guest Interface is active (2.1)

Step	Data	Expected Result
1. At the prompt, type the quit command	"quit"	The program terminates

## 2.4 Course List - Guest Version

**Test Objective:** Test that the Guest Course List shows up and has appropriate fields

**Preconditions:** Guest Interface is active (2.1), sample data has been loaded (1.3).

Step	Data	Expected Result
1. At the prompt, type the list courses command	"courses"	A non-empty list of courses is presented. Each course displays: <ul style="list-style-type: none"><li>• course Id</li><li>• title</li><li>• description</li></ul> If a course has sessions, each session is displayed under the course description, with the following information: <ul style="list-style-type: none"><li>• Session Code</li><li>• Start Date</li><li>• Capacity</li></ul>

## 2.5 Create student account

**Test Objective:** Test that a valid student account can be created, and that invalid student accounts cannot be created.

**Preconditions:** Guest Interface is active (2.1), sample data has been loaded (1.3). No other accounts have been created.

Step	Data	Expected Result
1. Run the new account command	"new"	<ul style="list-style-type: none"><li>• prompt for new account username</li></ul>
2. submit guest username	"guest"	<ul style="list-style-type: none"><li>• prompt for new password</li></ul>
3. submit password	"password"	<ul style="list-style-type: none"><li>• prompt for first name</li></ul>
4. submit first name	"a"	<ul style="list-style-type: none"><li>• prompt for last name</li></ul>
5. submit last name	"b"	<ul style="list-style-type: none"><li>• an error message is presented saying that the account already exists</li></ul>
6. Run the new account command	"new"	<ul style="list-style-type: none"><li>• prompt for new account username</li></ul>

<b>7. submit admin username</b>	"admin"	<ul style="list-style-type: none"> <li>• prompt for new password</li> </ul>
<b>8. submit password</b>	"password"	<ul style="list-style-type: none"> <li>• prompt for first name</li> </ul>
<b>9. submit first name</b>	"a"	<ul style="list-style-type: none"> <li>• prompt for last name</li> </ul>
<b>10. submit last name</b>	"b"	<ul style="list-style-type: none"> <li>• an error message is presented saying that the account already exists</li> </ul>
<b>11. Run the new account command</b>	"new"	<ul style="list-style-type: none"> <li>• prompt for new account username</li> </ul>
<b>12. submit rrome username</b>	"rrome"	<ul style="list-style-type: none"> <li>• prompt for new password</li> </ul>
<b>13. submit password</b>	"password"	<ul style="list-style-type: none"> <li>• prompt for first name</li> </ul>
<b>14. submit first name</b>	"a"	<ul style="list-style-type: none"> <li>• prompt for last name</li> </ul>
<b>15. submit last name</b>	"b"	<ul style="list-style-type: none"> <li>• an error message is presented saying that the account already exists</li> </ul>
<b>16. Run the new account command</b>	"new"	<ul style="list-style-type: none"> <li>• prompt for new account username</li> </ul>
<b>17. submit mflood username</b>	"mflood"	<ul style="list-style-type: none"> <li>• prompt for new password</li> </ul>
<b>18. submit password</b>	"password"	<ul style="list-style-type: none"> <li>• prompt for first name</li> </ul>
<b>19. submit first name</b>	"Matt"	<ul style="list-style-type: none"> <li>• prompt for last name</li> </ul>
<b>20. submit last name</b>	"Flood"	<ul style="list-style-type: none"> <li>• A message is displayed saying the account has been created</li> </ul>
<b>21. Send login command</b>	"login"	<ul style="list-style-type: none"> <li>• prompt for username</li> </ul>
<b>22. Submit mflood</b>	"mflood"	<ul style="list-style-type: none"> <li>• prompt for password</li> </ul>
<b>23. Submit password</b>	"password"	<ul style="list-style-type: none"> <li>• Login is successful, Student Interface is presented.</li> </ul>



### 3. The Admin Interface

#### 3.1 Logging into the Admin Account

**Test Objective:** The admin account is accessible and the admin help menu and admin prompt are displayed when one logs into the admin account.

**Preconditions:** Guest Interface is active (2.1)

Step	Data	Expected Result
1. Send login command	"login"	Prompt for username
2. Send admin username	"admin"	Prompt for password
3. Send admin password	"password"	Admin help menu is displayed. Command prompt looks like: (ADMIN) Command>

#### 3.2 The Admin Help Menu

**Test Objective:** Admin help screen is displayed when "help" command is issued or when an unrecognized command is issued

**Preconditions:** Admin Interface is active (3.1)

Step	Data	Expected Result
1. At the prompt, type the help command	"help"	The admin help screen is displayed
2. Type an unrecognized command	"garbage"	The admin help screen is displayed

#### 3.3 Quitting the program from the Admin Interface

**Test Objective:** The program exits when you type the quit command.

**Preconditions:** Admin Interface is active (3.1)

Step	Data	Expected Result
1. At the prompt, type the quit command	"quit"	The program terminates

### 3.4 Exiting from the Admin Interface to the Guest Interface

**Test Objective:** You can get back to the Guest interface by typing the exit command

**Preconditions:** Admin Interface is active (3.1)

Step	Data	Expected Result
1. At the prompt, type the quit command	"exit"	The Guest help menu is displayed, along with the guest command prompt.

### 3.5 Account List

**Test Objective:** The account list works.

**Preconditions:** Admin Interface is active (3.1), sample data has been loaded (1.3)

Step	Data	Expected Result
1. At the prompt, type the accounts command	"accounts"	A list of accounts is displayed. For each account, the role, first name, last name, username and password are displayed. The guest account does not have a role, first name, last name or password.

### 3.6 Course List - Admin Version

**Test Objective:** The course list works and the right data is displayed

**Preconditions:** Admin Interface is active (3.1), sample data has been loaded (1.3)

Step	Data	Expected Result
1. At the prompt, type the courses command	"courses"	A list of courses is displayed, ordered by title. Each course displays the course ID, title and description. For courses with sessions, each session displays the Session Code, Start Date, Capacity, Number of Students Enrolled, and a list of all students enrolled in the course.

## 4. The Student Interface

### 4.1 Logging into a Student Account

**Test Objective:** The student account is accessible and the student help menu and admin prompt are displayed when one logs into the admin account.

**Preconditions:** Guest Interface is active (2.1), sample data has been loaded (3.1)

Step	Data	Expected Result
1. Send login command	"login"	Prompt for username
2. Send rrome username	"rrome"	Prompt for password
3. Send rrome password	"password"	Student help menu is displayed. Command prompt looks like: (rrome) Command>

### 4.2 The Student Help Menu

**Test Objective:** Student help screen is displayed when "help" command is issued or when an unrecognized command is issued

**Preconditions:** Student Interface is active (4.1)

Step	Data	Expected Result
1. At the prompt, type the help command	"help"	The student help screen is displayed
2. Type an unrecognized command	"garbage"	The student help screen is displayed

### 4.3 Quitting the program from the Student Interface

**Test Objective:** The program exits when you type the quit command.

**Preconditions:** Student Interface is active (3.1)

Step	Data	Expected Result
1. At the prompt, type the quit command	"quit"	The program terminates

#### 4.4 Exiting from the Student Interface to the Guest Interface

**Test Objective:** You can get back to the Guest interface by typing the exit command

**Preconditions:** Student Interface is active (3.1)

Step	Data	Expected Result
1. At the prompt, type the quit command	"exit"	The Guest help menu is displayed, along with the guest command prompt.

#### 4.5 List of registered courses

**Test Objective:** You can list the courses you are registered for

**Preconditions:** Sample data has been loaded (1.3). Student Interface is active for user "rrome" (3.1), No courses have been (un)registered for the **rrome** account.

Step	Data	Expected Result
1. Send the list registered courses command	"listreg"	2 courses are displayed, ordered by course title: <ul style="list-style-type: none"><li>• INTRODUCTION TO JAVA PART 1 - WIN14 Session</li><li>• INTRODUCTION TO JAVA PART 2 - SPR15 Session</li></ul>

#### 4.6 Course List - Student Version

**Test Objective:** The course list works and the right data is displayed in the right order.

**Preconditions:** Sample data has been loaded (1.3). Student Interface is active for user "rrome" (3.1), No courses have been (un)registered for the rrome account.

Step	Data	Expected Result
1. At the prompt, type the courses command	"courses"	<p>A list of courses is displayed, ordered by title. Each course displays the course ID, title and description. For courses with sessions, each session displays the Session Code, Start Date, Capacity, Number of Students Enrolled.</p> <p>There are *** asterisks next to the sessions that rrome is signed up for:</p> <ul style="list-style-type: none"><li>• INTRODUCTION TO JAVA PART 1 - WIN14 Session</li><li>• INTRODUCTION TO JAVA PART 2 - SPR15 Session</li></ul>

## 4.7 Registering for a course

### 4.7.1 - Happy Path - Single Session

**Test Objective:** A student can register for a course session. In this case, the Course has a single session, so the student just has to provide the course ID, and the session is selected automatically.

**Preconditions:** Sample data has been loaded (1.3). Student Interface is active for user "rrome" (3.1), the **rrome** account has not registered for the GL101/WIN14 Course Session.

Step	Data	Expected Result
1. Send the register for course command	"reg"	Prompt for course ID
2. Submit course ID	"GL101"	Message shows success. Student Command Prompt is displayed.
4. Send list registered courses command	"listreg"	The GL101 course, WIN14 session shows up in the list of registered courses.

### 4.7.2 - Happy Path - Multi-Session

**Test Objective:** A student can register for a course session. In this case, the Course has more than one session available, so the student has to choose the session.

**Preconditions:** Sample data has been loaded (1.3). Student Interface is active for user "rrome" (3.1), the **rrome** account has not registered for the WAT101/WIN14 Course Session.

Step	Data	Expected Result
1. Send the register for course command	"reg"	Prompt for course ID
2. Submit course ID	"WAT101"	The 2 session for WAT101 are displayed prompt for session code
3. Submit session Code	"WIN14"	Message shows success. Student Command Prompt is displayed.
4. Send list registered courses command	"listreg"	The WAT101 course, WIN14 session shows up in the list of registered courses.

#### 4.7.3 - Invalid Course ID

**Test Objective:** A student cannot register for an invalid course

**Preconditions:** Sample data has been loaded (1.3). Student Interface is active for user "rrome" (3.1). There is no course in the system with the course ID "junk".

Step	Data	Expected Result
1. Send the register for course command	"reg"	Prompt for course ID
2. Submit an invalid course ID	"junk"	Error message displays that an invalid course was provided. Student Command prompt is presented.

#### 4.7.4 - Can't re-register - Single Session

**Test Objective:** A student cannot register for a course session they are already registered for. In this case, the course only has one session.

**Preconditions:** Sample data has been loaded (1.3). Student Interface is active for user "rrome" (3.1), The **rrome** account is registered for the JAVA101 course.

Step	Data	Expected Result
1. Send the register for course command	"reg"	Prompt for course ID
2. Submit a valid course ID with a single session that the user is already enrolled in.	"JAVA101"	Error message displays that the user is already registered for that course Student Command prompt is presented.



#### 4.7.5 - Can't re-register - Multi Session

**Test Objective:** A student cannot register for a course session they are already registered for. In this case, the Course has more than one session, and the user is signed up for one of them.

**Preconditions:** Sample data has been loaded (1.3). Student Interface is active for user "rrome" (3.1), the **rrome** account has signed up for the WAT101/WIN14 Course Session (4.6.2).

Step	Data	Expected Result
1. Send the register for course command	"reg"	Prompt for course ID
2. Submit a valid course ID with a single session that the user is already enrolled in.	"WAT101"	List of course sessions is displayed. Prompt for Session ID.
3. Submit Session Code for session already enrolled in	"WIN14"	Error message displays that user is already registered for that session. Student command prompt is displayed.

## 4.8 Unregistering from a course

### 4.8.1 - Happy Path

**Test Objective:** A student can unregister from a course session

**Preconditions:** Sample data has been loaded (1.3). Student Interface is active for user "rrome" (3.1), the **rrome** account has not unregistered from the JAVA101/WIN14 Course Session.

Step	Data	Expected Result
1. Send list of registered courses command	"listreg"	The JAVA101 course shows up in the list of courses that user <b>rrome</b> is registered for.
2. Send the unregister from course command	"unreg"	Prompt for course ID
3. Submit course ID	"JAVA101"	prompt for Sesion Code
4. Submit Session Code	"WIN14"	Success message is displayed Student Command Prompt is displayed.
5. Send list of registered courses command	"listreg"	The JAVA101 course does not show up in the list of courses that user <b>rrome</b> is registered for.

#### 4.8.2 - Cannot unregister from a course you are not registered for

**Test Objective:** A student cannot unregister from a course session they are not registered for

**Preconditions:** Sample data has been loaded (1.3). Student Interface is active for user "rrome" (3.1), the **rrome** account has not unregistered from the JAVA101/WIN14 Course Session.

Step	Data	Expected Result
1. Send list of registered courses command	"listreg"	The CLOUD101 course does not show up in the list of courses rrome is registered for
2. Send the unregister from course command	"unreg"	Prompt for course ID
3. Submit course ID	"CLOUD101"	Prompt for Session Code
4. Submit Session Code	"WIN14"	Error message is displayed saying "Invalid course/session supplied" Student Command Prompt is displayed.