**Functional Requirements**

**Whole System:**

FR1. The system shall receive data from an arbitrary number of external sources.

FR2. The system shall label the classification level (i.e. unclassified, secret, and top secret) for any data received.

FR3. They shall store this data in a database.

FR4. The system shall be interfaced via an API endpoint for web applications.

FR5. The system shall authenticate the user every time they access an individual API.

FR6. The system shall execute or deny requests based on the user’s security context.

FR7. Users shall be able to read data, write data, and view reports of data which they have access to.

**Data Storage:**

FR8. Data elements shall be stored in the database alongside their SELinux labels

FR9. The user shall only be allowed to access data they have the privilege to access and shall be restricted from accessing any other data.

FR10. Users shall be able to submit data for storage they have access to based on their security context.

**Container Runtime:**

FR11. The container shall provide compute resources for RESTful services behind the API Gateway.

FR12. This component shall start up a container with the enforced security context of the requested service.

FR13. The service running inside the container must gracefully handle ‘Access Denied’ errors when attempting to retrieve data from the Data Storage component.

FR14. This container shall be able to be activated and deactivated at arbitrary times to serve data to users with the proper security context.

FR15. This component shall ensure multiple instances of RESTful services running inside each container cannot share information with each other.

**Logging:**

FR16. All logs shall be forwarded to an external log aggregator.

FR17. If a failure of log forwarding occurs, an alert shall be sent where an operator will see it.

**Encryption:**

FR18. The system shall encrypt the data based on its security context.

**Non-functional Requirements**

NFR1. The system shall prioritize security over performance

NFR2. It shall ensure container startup time is less than 5 seconds. The Docker images shall include all executables, libraries, and configuration data so that application startup is the only action required when the container starts.

**Constraints**

C1. The final configuration must run on CentOS 7 with SELinux enabled.

C2. AWS shall be used for the API gateway and serving endpoints.

C3. The PostgreSQL database should use scram-sha-256 for password authentication.

C4. This component shall allow connections from the API Gateway to the Docker Daemon on TCP Port 2376 with configured encrypted communication.

C5. Each RESTful service shall be given login credentials to connect to the SE-PostgreSQL database within the Data Storage component.

**UC1: Logging In**

1.1 Preconditions

The account with the pertinent user type (student, instructor, coordinator) has been set up in the system.

1.2 Main flow

The user inputs their username [S1] and is greeted with a list of possible actions [S2][S3][S4].

1.3 Subflows

* [S1] The system is able to display the proper menus for the user type associated with the given username [E1]. We will not enforce passwords in this system.
* [S2] A coordinator has the ability to view their account details, modify courses, and modify other users.
* [S3] An instructor has the ability to view their account details and modify grades.
* [S4] A student has the ability to view their account details, view their course schedule, and view their grades.

1.4 Alternate flows

* [E1] If no account has been set up in the system, the user is not allowed to log in.

1.5 Logging

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Transaction code | Description | Logged in username | Secondary user | Transaction type |
| 100 | Failed login | IP address | N/A | Other |
| 101 | Successful login | user | N/A | Other |
| 102 | Logged out | user | N/A | Other |

1.6 Data Format

|  |  |
| --- | --- |
| Field | Format |
| User name | Between 6 and 20 alpha characters and symbols - or \_ |

**UC2: Viewing Account Details**

2.1 Preconditions

The active user has an account set up with the system and has logged in (UC1)

2.2 Main flow

No matter what user type, users can see their username, id number, and full name [S1]. Instructors can see their course schedule and course rosters [S2]. Students can see their course schedule and GPA [S3]. Coordinator can see information about all students and instructors.

2.3 Subflows

* [S1] No user can modify account details.
* [S2] Instructors cannot add their own courses.
* [S3] Course grades are set for students by instructors and can be empty.

2.5 Logging

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Transaction code | Description | Logged in username | Secondary user | Transaction type |
| 200 | View options | user | N/A | View |

**UC3: Course Modification**

3.1 Preconditions

A coordinator has an account in the system and has logged in (UC1)

3.2 Main flow

Coordinators can add courses [S1], and remove courses [S2]. Coordinators can enroll students in courses [S3].

3.3 Subflows

* [S1] Coordinators add courses by specifying a course name, day, time range, and instructor [E1].
* [S2] Coordinators can select a course to remove from a list of existing courses.
* [S3] Coordinators can add a student to a course after it has been created by entering the student’s username [E2] [E3].

3.4 Alternate flows

* [E1] A course addition is denied for a nonexistent instructor.
* [E2] A student addition is denied for a nonexistent student.
* [E3] A student cannot be enrolled more than once in the same course.

3.5 Logging

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Transaction code | Description | Logged in username | Secondary user | Transaction type |
| 300 | Successful add course | coordinator | N/A | Add |
| 301 | Successful delete course | coordinator | N/A | Delete |
| 302 | Successful add student | coordinator | student | Add |
| 303 | Failed add student | coordinator | N/A | Other |

3.6 Data Format

|  |  |
| --- | --- |
| Field | Format |
| courseName | Between 6 and 20 alpha characters |
| day | Between 1 and 6 day initials (M,T,W,Th,F,S) |
| time | HH:MM |
| instructor | Between 6 and 20 alpha characters and symbols - or \_ |
| student | Between 6 and 20 alpha characters and symbols - or \_ |

**UC4: User Modification**

4.1 Preconditions

A coordinator has an account in the system and has logged in (UC1)

4.2 Main flow

Coordinators can add users [S1], and remove users [S2].

4.3 Subflows

* [S1] Coordinators add users by specifying a user type, username [E1], and full name. Id numbers will be assigned by the system.
* [S2] Coordinators can specify a username to remove from a list of existing users.

4.4 Alternate flows

* [E1] Coordinators cannot add users with duplicate usernames.

4.5 Logging

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Transaction code | Description | Logged in username | Secondary user | Transaction type |
| 400 | Successful add user | coordinator | user | Add |
| 401 | Successful delete user | coordinator | user | Delete |
| 402 | Failed add user | coordinator | user | Add |

4.6 Data Format

|  |  |
| --- | --- |
| Field | Format |
| userType | Student/ Instructor |
| username | Between 6 and 20 alpha characters and symbols - or \_ |
| name | Between 6 and 40 alpha characters and symbols - or \_ |

**UC5: Grade/ Roster Modification**

5.1 Preconditions

An instructor has an account set up with the system, has been added to a course with an enrolled student, and has logged in (UC1) (UC3)

5.2 Main flow

Instructors can select a student from any course and modify their grade [S1]. Instructors can select a course and remove a certain student from their roster [S2].

5.3 Subflows

* [S1] Grades are stored for each student-course pair as a floating-point number between 0 and 4 [E1].
* [S2] Instructor can see a roster for each course he is assigned. He can select the option to delete/ remove a student by their username [E2].

5.4 Alternate flows

* [E1] It could be empty if none of the courses have been graded yet.
* [E2] The specified username is not on the roster.

5.5 Logging

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Transaction code | Description | Logged in username | Secondary user | Transaction type |
| 500 | Successful add/ change grade | coordinator | student | Add |
| 501 | Successful delete student | coordinator | student | Delete |

5.6 Data Format

|  |  |
| --- | --- |
| Field | Format |
| grade | Float value between 0 and 4 |
| username | Between 6 and 20 alpha characters and symbols - or \_ |

**UC6: Viewing Course Schedule**

6.1 Preconditions

A student or instructor has an account with the system, has logged in, and has been added to a course (UC1) (UC3)

6.2 Main flow

The user can see their schedule for the week [S1]. Students see the grades associated with their class [S2] and instructors have the option to navigate to the roster for each class [S3].

6.3 Subflows

* [S1] The user can see their schedule for the upcoming week.
* [S2] If the user is a student, their grade is displayed next to the course [E1].
* [S3] If the user is an instructor, they can just select their course and be shown the course roster.

6.4 Alternate flows

* [E1] It could be empty if none of the courses have been graded yet.

6.5 Logging

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Transaction code | Description | Logged in username | Secondary user | Transaction type |
| 400 | Successful see schedule | coordinator/ student | N/A | View |
| 401 | Successful see roster | coordinator | N/A | View |
| 402 | Successful see grade | student | N/A | View |

###### **UC7: View access logs**

7.1 Preconditions

A user is a registered user of the Course Manager system. The user has authenticated himself or herself into the system.

7.2 Main Flow

A user can add data or request data. Every transaction will be logged so the coordinator can track any illegal access.

7.5 Logging

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Transaction code | Description | Logged in username | Secondary user | Transaction type |
| 700 | Access logs viewed | coordinator | N/A | View |