***SOFTWARE ANALYSIS SPECIFICATION***

**1.0 Introduction**

**This section provides an overview of the entire requirement document. This document describes all data, functional and behavioral requirements for software.**

**1.1 Goals and objectives**

**Overall goals and software objectives are described.**

This software’s objective is to allow the company a better method of self auditing in terms of increasing security, decreasing paperwork, and reduce the amount of time it takes to audit departments of the company.

**1.2 Statement of scope**

**A description of the software is presented. Major inputs, processing functionality and outputs are described without regard to implementation detail.**

Inputs: ~

1. Answers: Departments will be able to answer documents through text boxes. The departments can then submit the STENERs.
2. Questions: The Oversight Team will be able to add, remove, and modify question on the STENERs.
3. Login: Both Oversight and Department will be inputting their login credentials to access interfaces appropriate to their positions.
4. STENERS: Oversight will be able to create, modify, reject, and approve STENERs. They will also be able to submit logs on violations with a date and description.

Processing Functionality: ~

1. Modifications to STENERs and questions by oversight and submitted STENER answers are updated to the Database.
2. STENERs with questions that have not been completed will not be accepted by the system.

**1.3 Software context**

**The software is placed in a business or product line context. Strategic issues relevant to context are discussed. The intent is for the reader to understand the 'big picture'.**

* This is a low-risk software that will be developed by 5 developers.
* It is meant to be used in-house in a formal environment.
* This software will be used to conduct legal work for the entity that it is inclosed in. The legal work has connections to government interactions.
* The software will support a single platform for the time being.

**1.4 Major constraints**

**Any business or product line constraints that will impact the manner in which the software is to be specified, designed, implemented or tested are noted here.**

There are no major constraints to mention.

**2.0 Usage scenario**

**This section provides a usage scenario for the software. It organized information collected during requirements elicitation into use-cases.**

**2.1 User profiles**

Oversight user- They will be able to control the software, by managing the users, STENERS, by modifying the questions on them, as well as the delivery, accepting, and rejection of them. They will also keep a log of all the violations committed by the departments.

Department user- They will be able to fill out the appropriate STENER for their department, and type in the proper information to the questions on the STENERs.

**2.2 Major software functionality**

* LOGIN
* Oversight
  + Receive a submitted STENER
  + Create a STENER (one STENER per Department)
  + Adding/Editing/removing questions on STENER
  + Approving/rejecting STENER
  + Can make Violation logs when rejecting STENER
  + Add/Remove/Modify Users
* Department
  + Answer current STENER
  + Submit STENER to oversight team.
  + Able to answer N/A as an answer
  + Able to submit where evidence is rather than input a file as evidence
  + Able to view violation logs associated to their department
* STENER
  + Save current document when anything is submitted to it.
  + Violation Logs

**2.3 Special usage considerations**

There are no special requirements for the software to be used.

**3.0 Data Model and Description**

**This section describes information domain for the software**

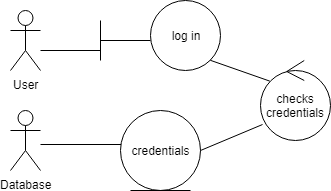
**3.1 Data Description**

**Robustness Diagrams - data objects that will be managed/manipulated by the software are described in this section.**

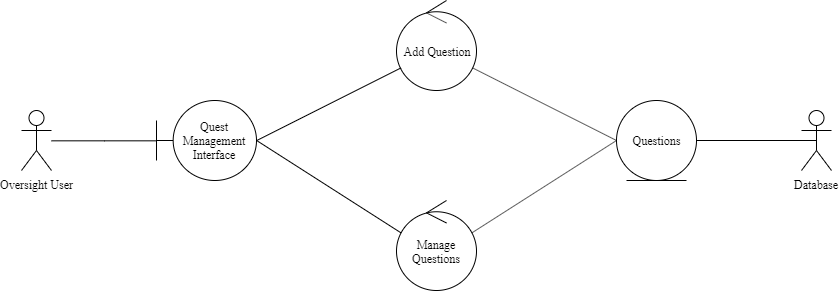
Use Case 1

****

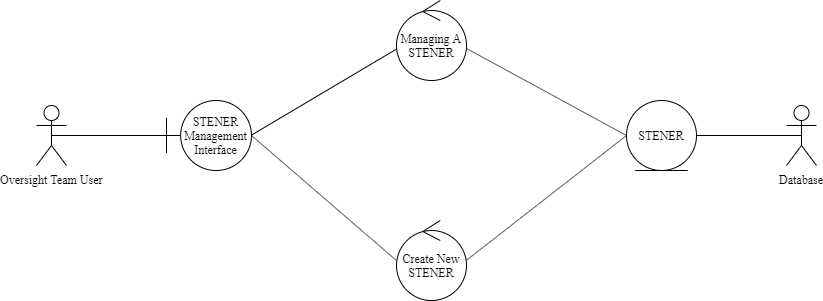
Use case 2

****

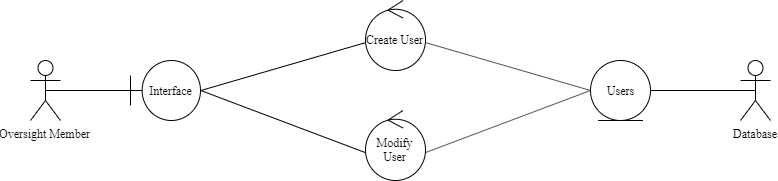
Use case 3

****

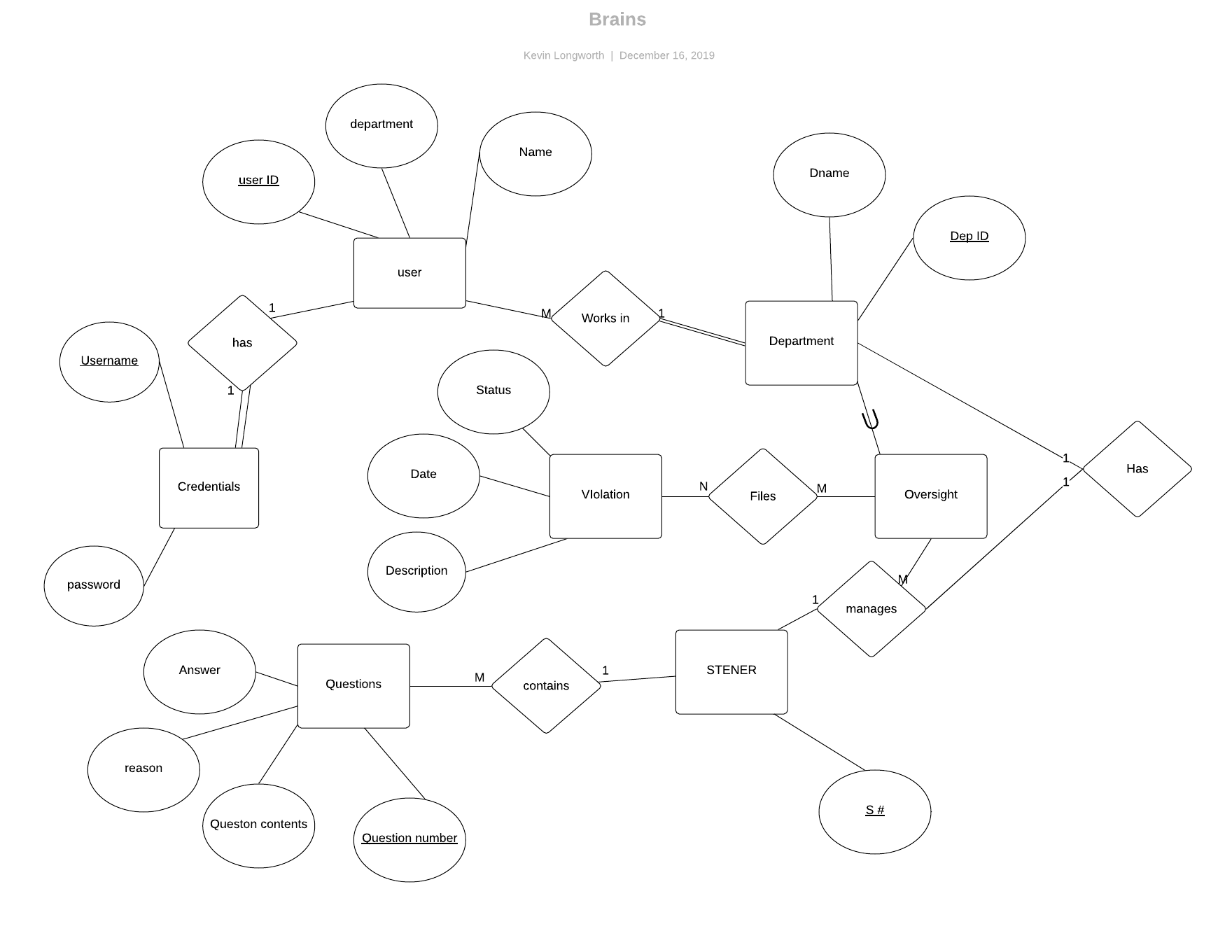
Use case 4



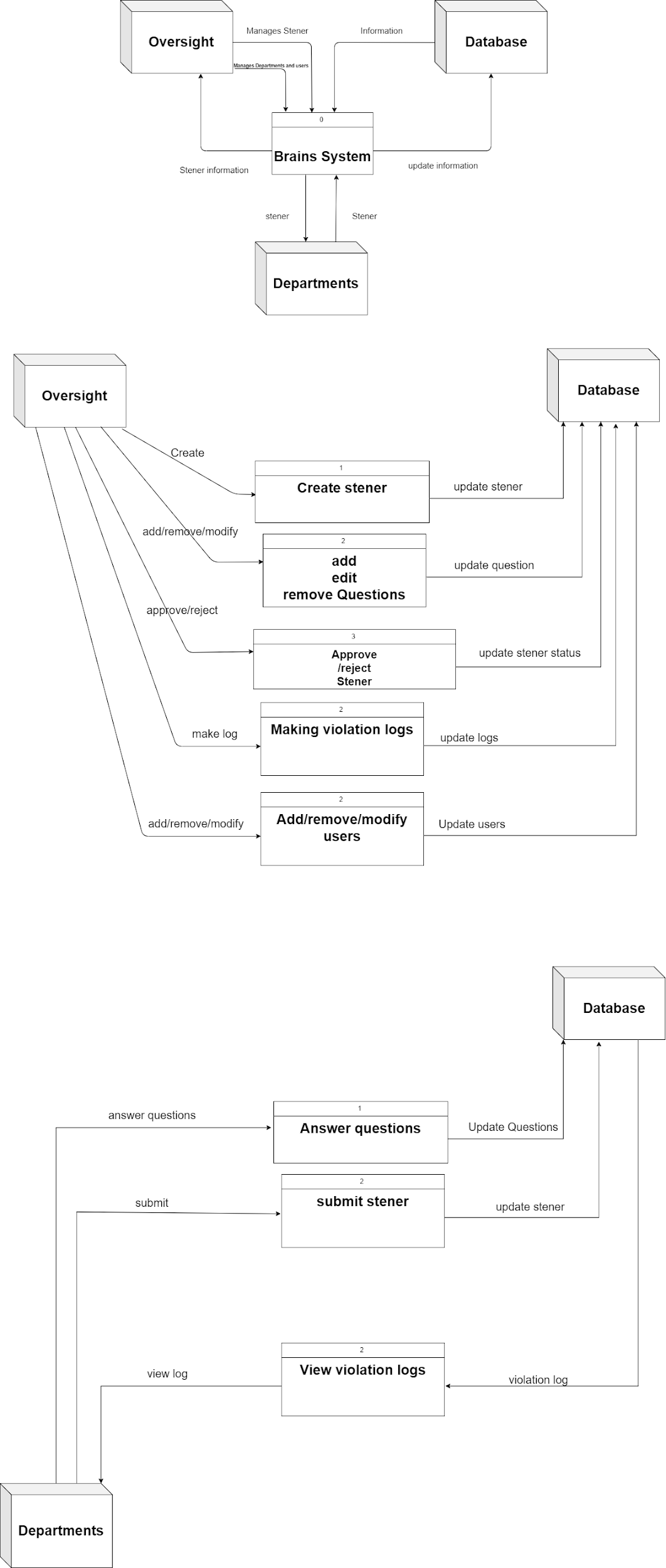
Use case 5

****

**3.1.1 Entity-Relationship Diagram**

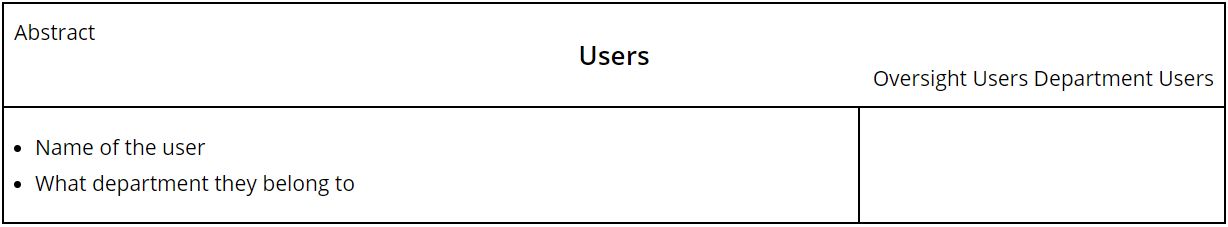
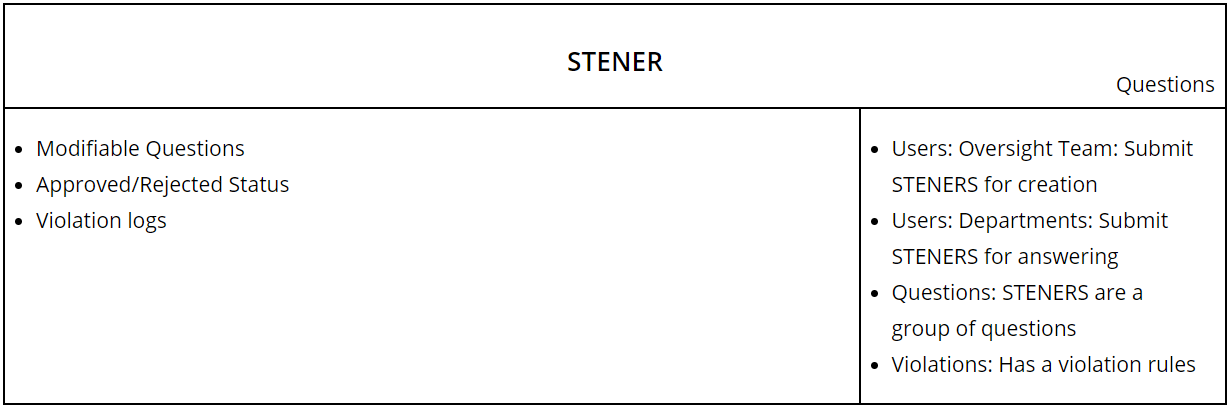
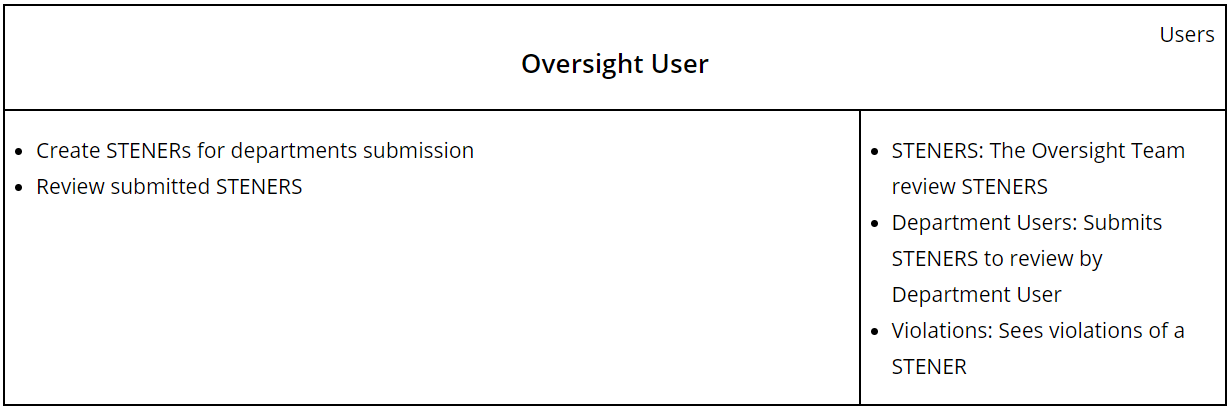
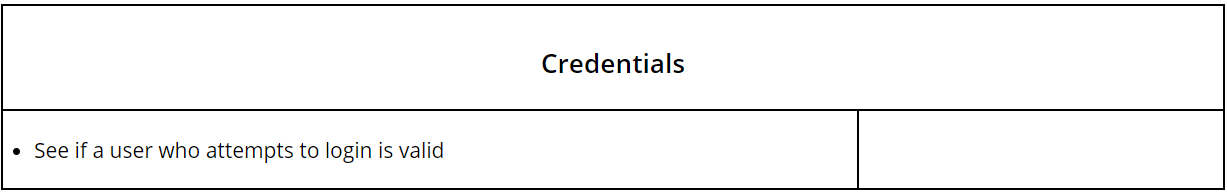


**3.1.2 Data Flow Diagram**

****

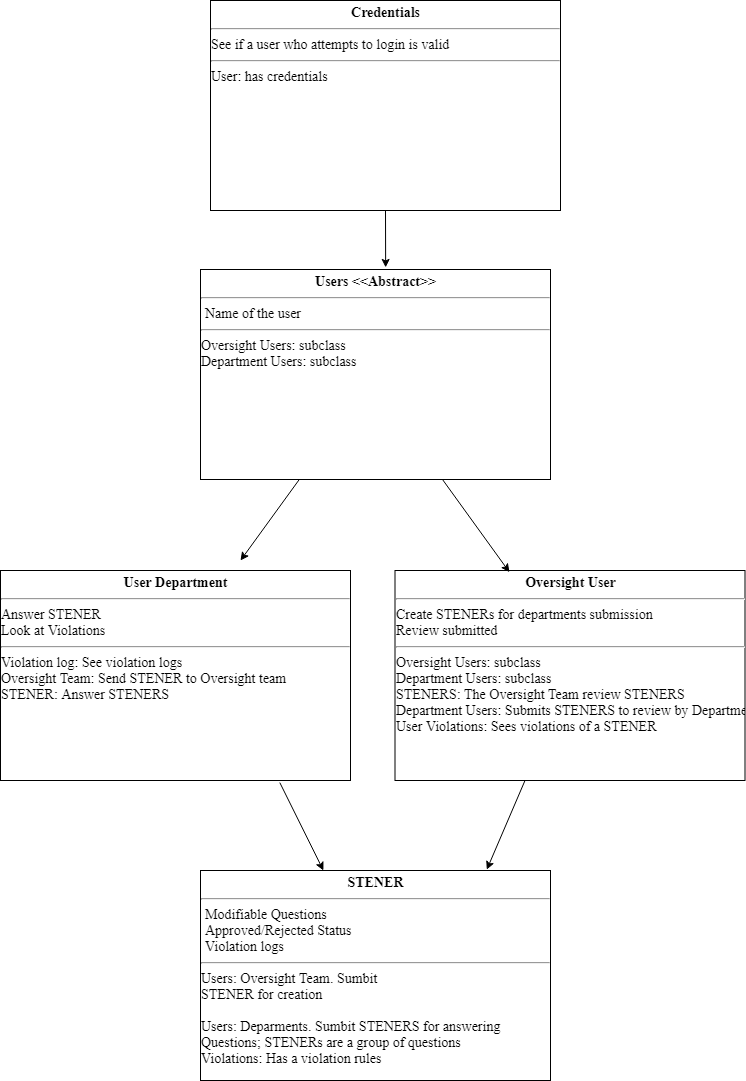
**3.1.3 Object Relationships**

**Relationships among data objects are described using CRC cards. No attempt is made to provide detail at this stage.**

****

**3.1.4 Complete data model**

**An UML Class model (class diagram) for the software is developed – through attributes and actions (not data typing, method signatures, access**

****

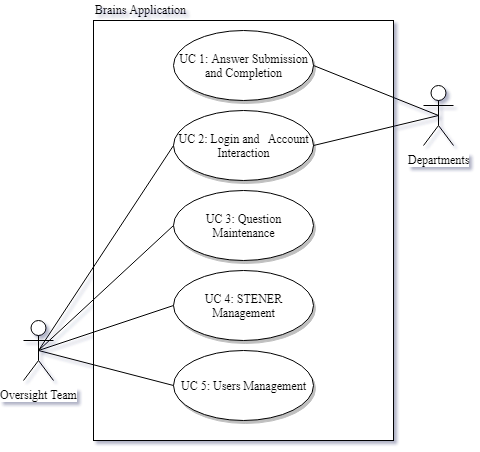
**3.1.5 Data dictionary**

<https://drive.google.com/drive/u/0/folders/1jqjH4FglkTj92gf2I6sMMYNnz3ShDr2C>

**4.0 Functional Model and Description**

**Description of major software functions along with UML Use Case, sequence, and communication diagrams.**

**4.1 Use cases**



**4.2 Software Interface Description**

**The software interface(s)to the outside world is(are) described.**

**4.2.1 External machine interfaces**

There are no other interfaces to external machines

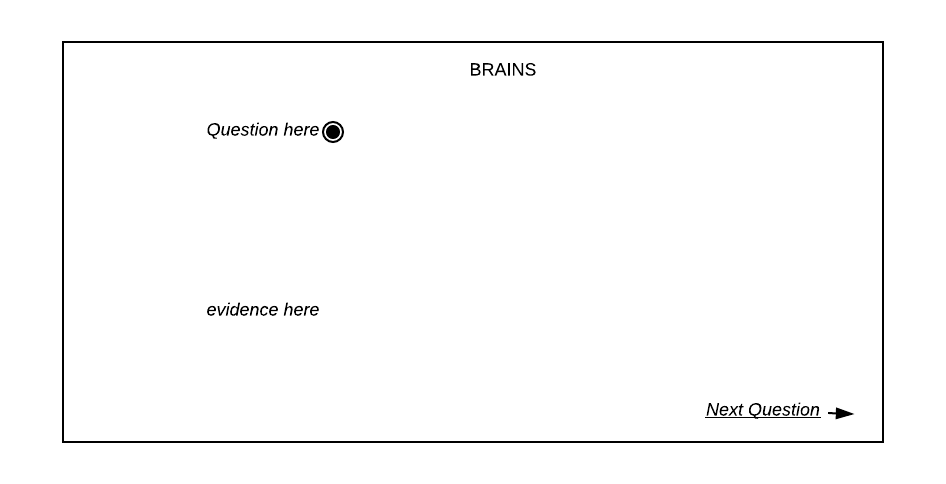
**4.2.2 External system interfaces**

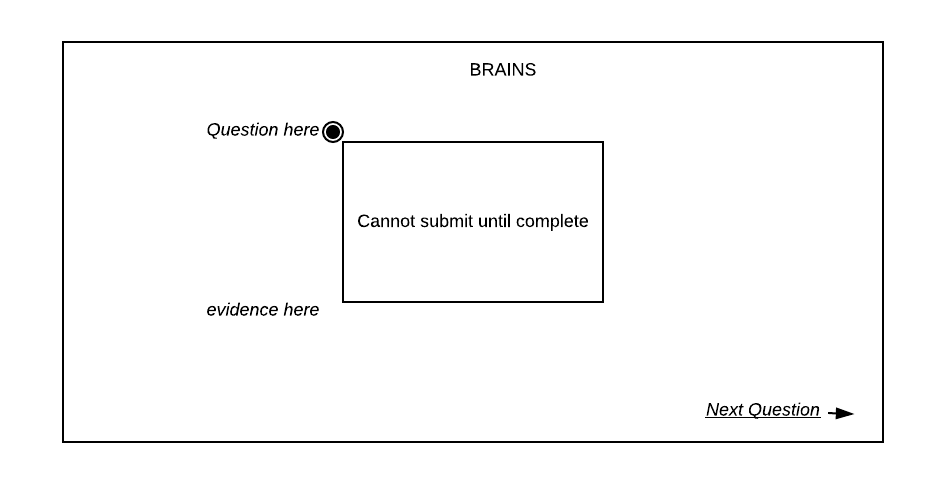
The external interface for the database is through microsoft access, which allows data be manipulated from the program as well as managing it.

**4.2.3 Human interface**

There is no human interfaces involved

**4.2.3.1 User screen interface layouts**

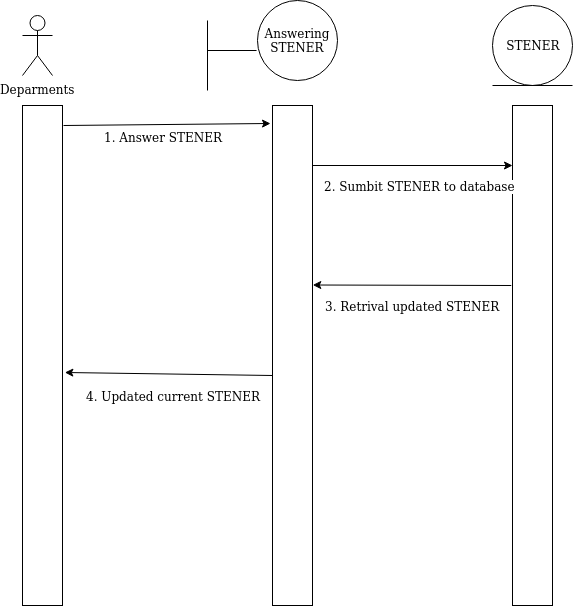
****

****

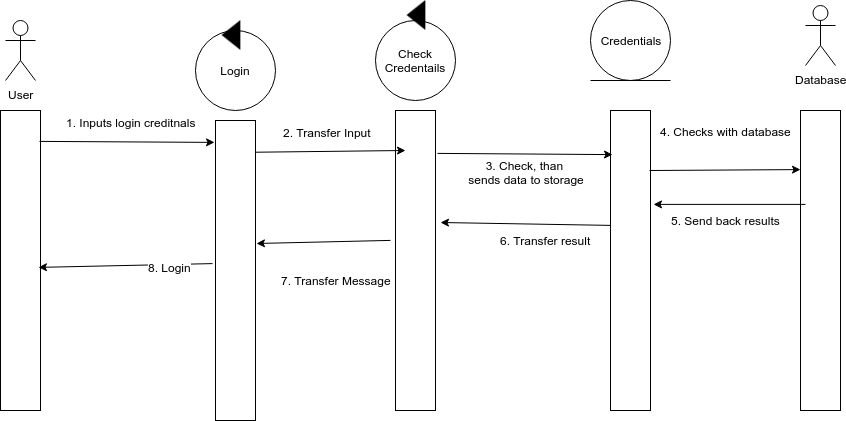
**4.3 Sequence Diagrams**

**Used to model the class interactions needed for the use cases.**

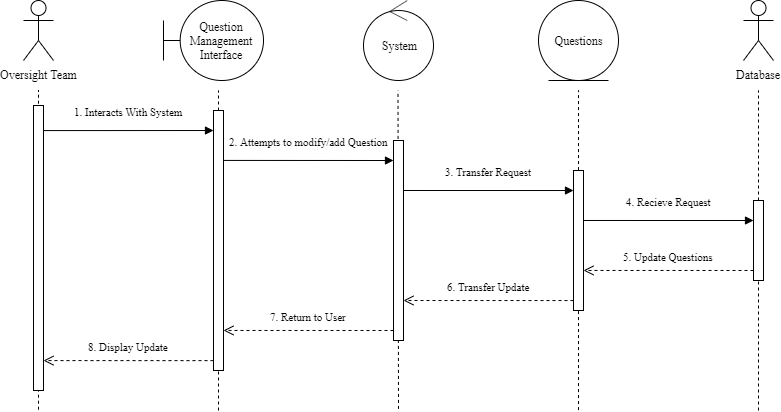
Use case 1



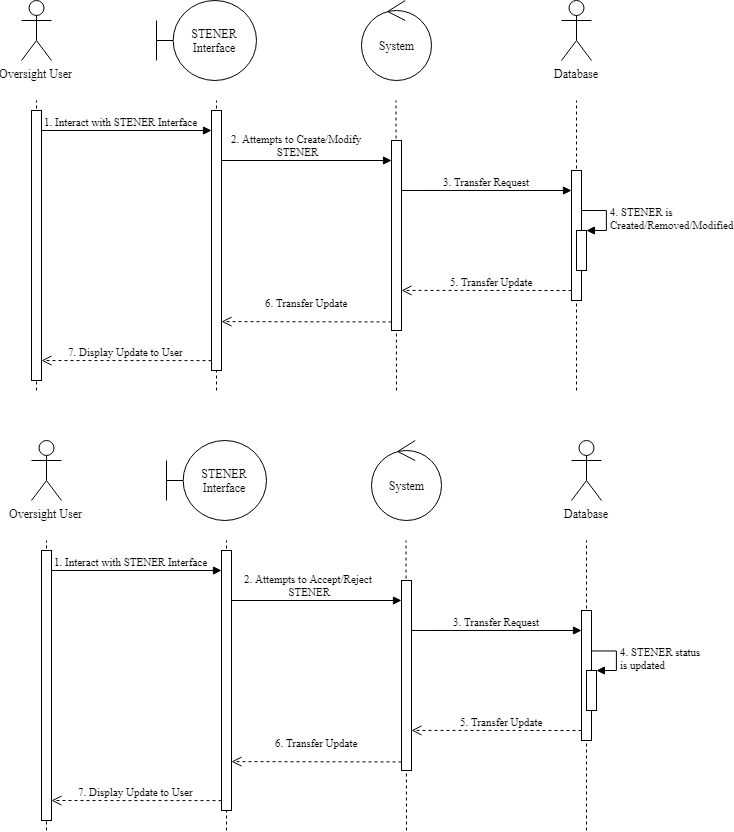
Use case 2

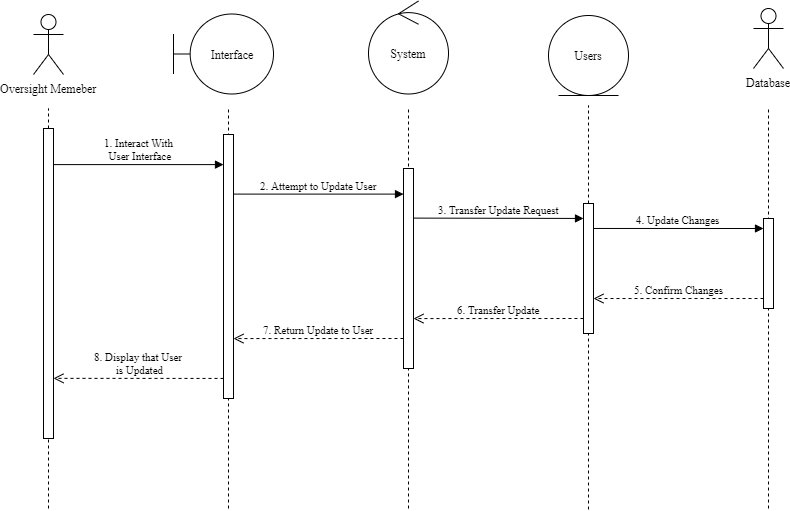
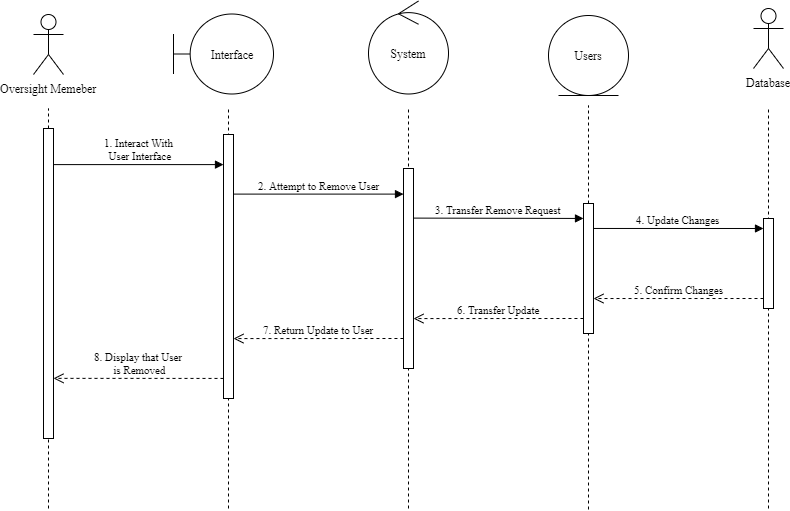
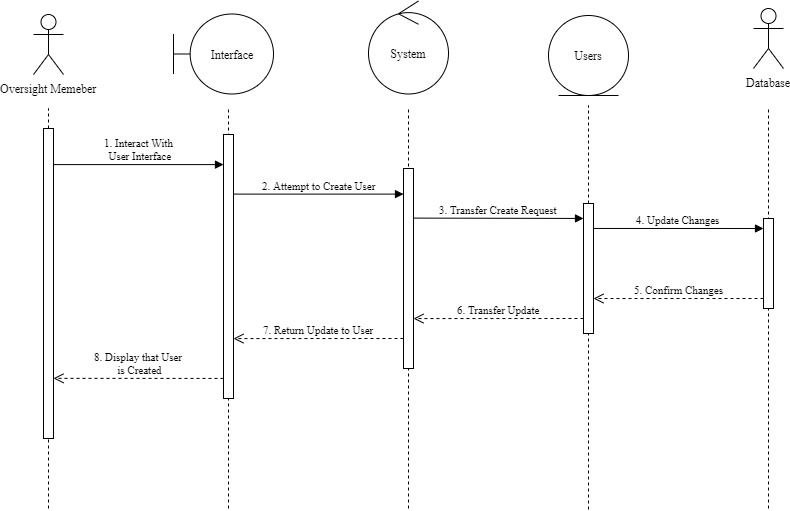


Use case 3



Use case 4

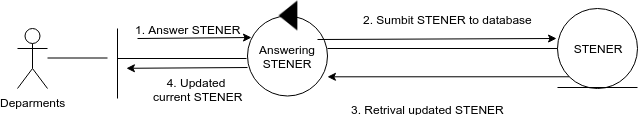


Use case 5

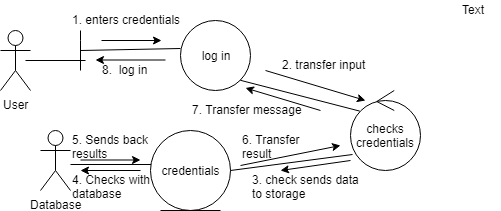
**4.4 Communication Diagrams**

**Used to model the message passing structure of the system functions.**

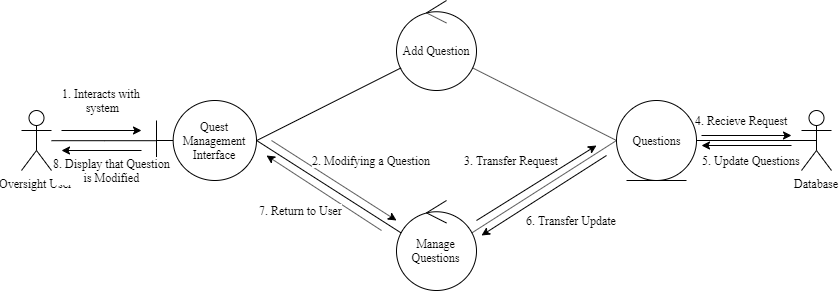
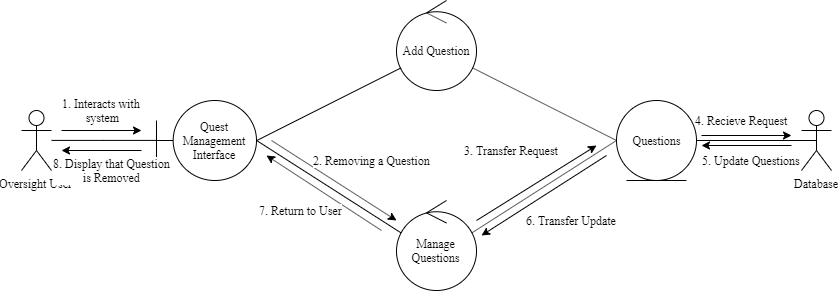
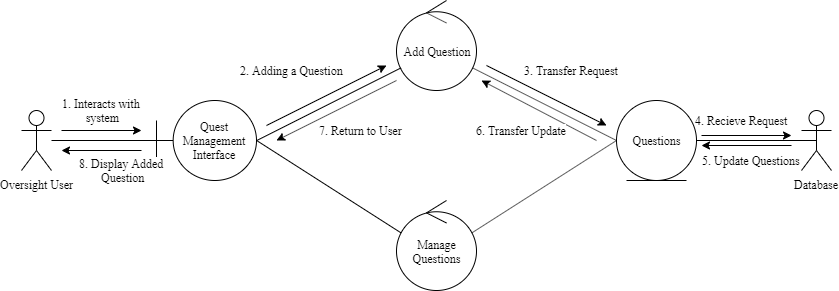
Use case 1



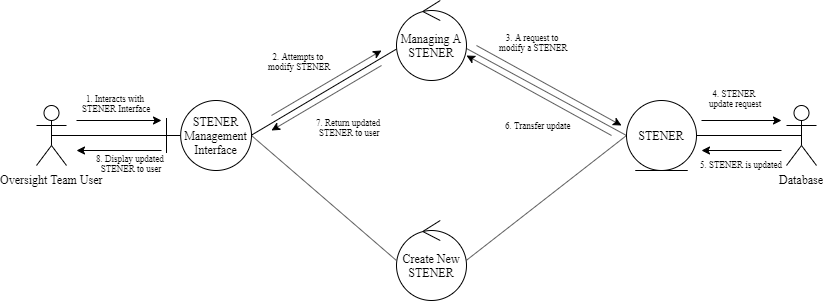
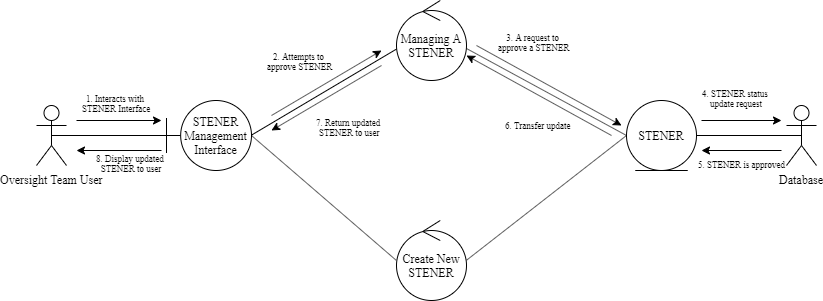
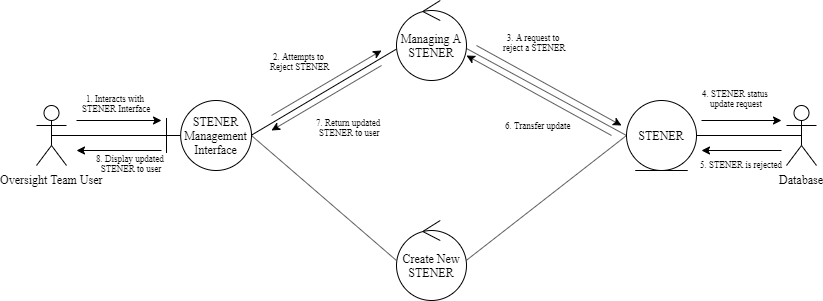
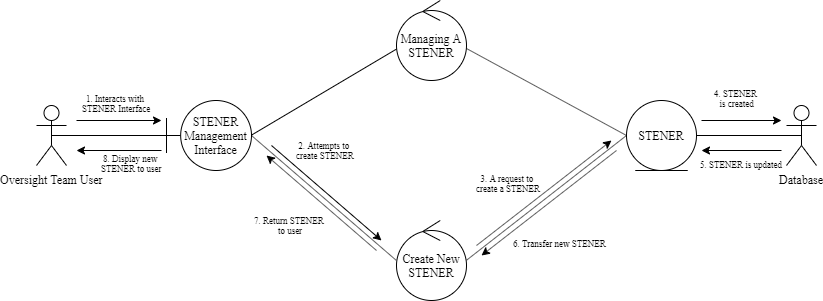
Use case 2

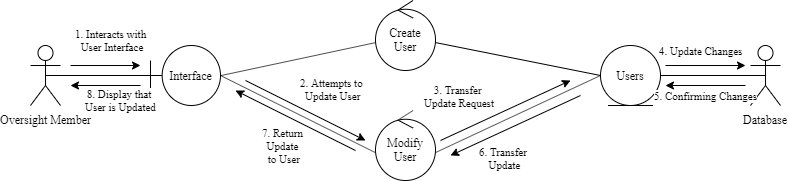
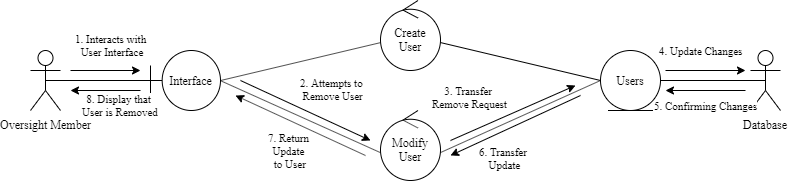
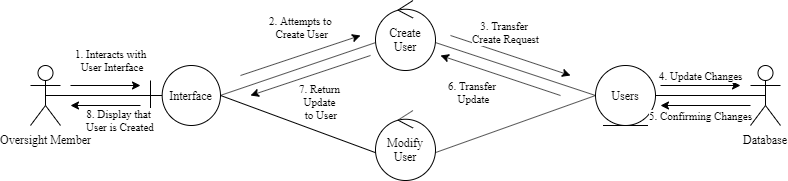


Use case 3



Use case 4



Use case 5

**5.0 Behavioral Model and Description**

**A description of the behavior of the software is presented.**

**5.1 Description for software behavior**

**A detailed description of major events and states is presented in this section.**

**5.1.1 Events**

**A listing of events (control, items) that will cause behavioral change within the system is presented.**

1. Entering credentials to login screen and attempting to login
2. Login Credentials are Invalid
3. User is Department User
4. User is Oversight User
5. Click view to Violation Logs
6. Pressing the submit button after answering questions
7. Missing Answers in STENER
8. Add, Remove, or Modify Users
9. Interacting with STENER Interface
10. Leaving STENER Interface
11. Delete, add, modify, reject, or approve a STENER
12. Interacting with Question Interface
13. Delete, add, or modify a question
14. Leaving Question Interface
15. Press yes in create violations menu
16. Create violation logs
17. Exit the Application

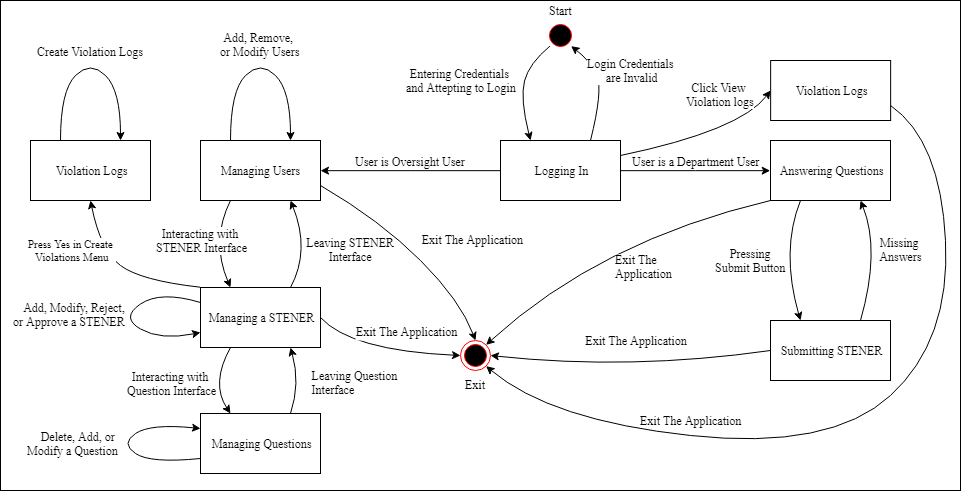
**5.1.2 States**

**A listing of states (modes of behavior) that will result as a consequence of events is presented.**

1. Start
2. Logging In
3. Submitting STENER
4. Answering Questions
5. Managing Questions
6. Managing a STENER
7. Managing Users
8. Violation Logs
9. Exit

**5.2 State Transition Diagrams**

**Depict the manner in which the software reacts to external events.**

****

**6.0 Restrictions, Limitations, and Constraints**

Some special limitations is the amount of time there is to complete the software, the physical resources we have and the experiences we have in creating software.

**7.0 Validation Criteria**

**The approach to software validation is described.**

**7.1 Classes of tests/Test Strategy**

Testing for valid input- testing to see if a valid input works in the input field.

testing for invalid input- see if inputs that are not supposed to be valid come up as invalid in the program.

test for no input- see if the program allows for the software to take in no input depending on the field.

test for too much input- if the size of the input is too big.

**7.2 Expected software response**

Valid input- the input correctly goes through and is stored.

Invalid input- an error message occurs and the user re-enters.

No input- The system asks for the user to enter an input.

Too big a value- an error is shown that the input is too big.

**7.3 Performance bounds**

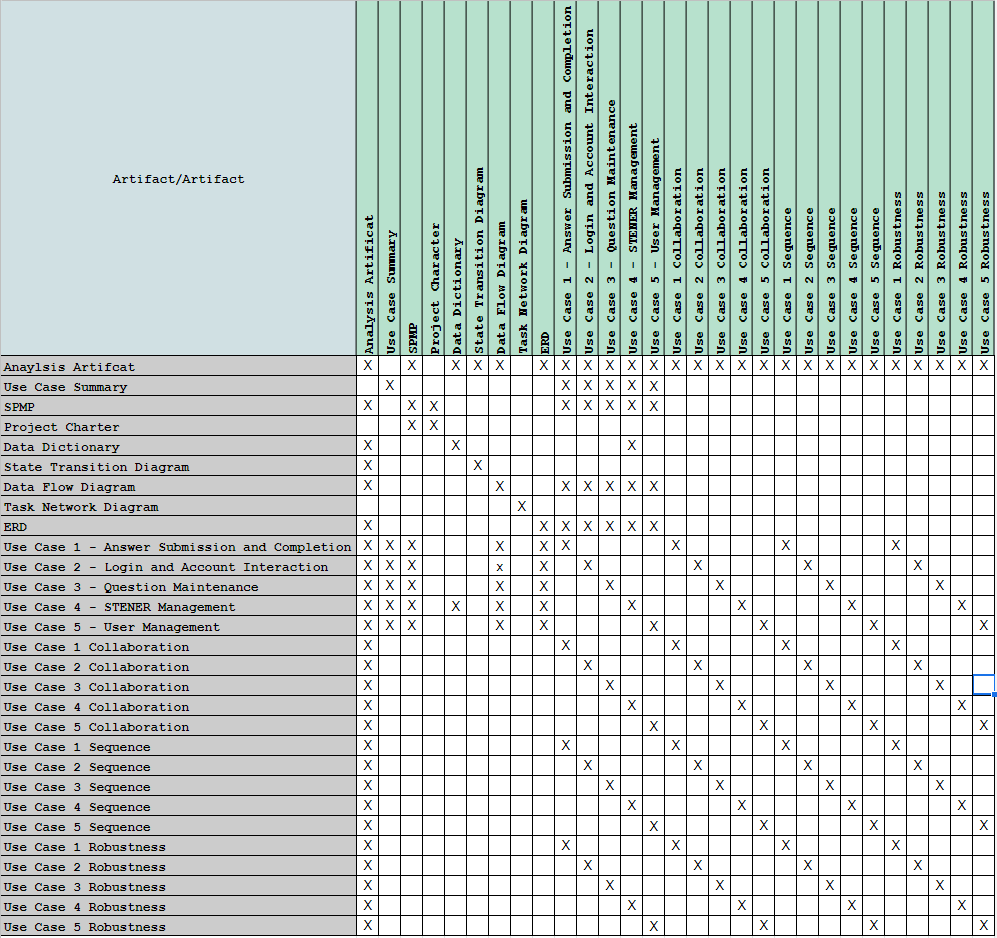
none

**8.0 Appendices**

**Presents information that supplements the Requirements Specification**

**8.1 System traceability matrix**

**A matrix that traces stated software requirements back to the system specification.**

****

**8.2 Product Strategies**

**If the specification is developed for a product, a description of relevant product strategy is presented here.**

The goal of this project is to both speed up and secure the auditing process, as to better comply with regulations.

**8.3 Analysis metrics to be used**

**A description of all analysis metrics to be used during the analysis activity is noted here.**

* User certification: the time it takes for a user to be authenticated.
* STENER transfer: The time it takes the database to send and receive STENERS.

**8.4 Supplementary information (as required)**