|  |  |
| --- | --- |
|  | Identify Design Mechanisms |
|  |  |
|  | Group 5  **OOAD INT3310 20**  11/28/19 |

***SalesMax***

**A POS system**

1. **Introduction:**
   1. **Purpose:**

This is a report about the Design Mechanisms Specification of the SalesMax system, a POS system. This describe architectural mechanisms used in the project.

* 1. **Intended Audiences:**
* Target:
  + Project Manager ( PM): Person who take responsibility for software quality. PM should carefully and thoroughly read this document
  + Programmer: The one to setup software from imagebuilding and documents to software that is able to executed. This document help installing software more accurately
  + Tester: Have to ensure requests are valid before confirmation. Tester should also read this document
  + User: User who have access to entire development proccess and during system transfering
* Document should also describe non-functional requirements of the system such as bounded requirements, usability, reliability, performance, etc..
  1. **Product’s goal:**

This system is for anyone whose demands is for buying or storing items, searching and managing items.

* 1. **Product’s scope:**

**The software’s main users are sellers and buyers/ consumers. In an enviroment created by the software, user can look up, buy, sell, interact with other users as well as storing their item in personal storage.**

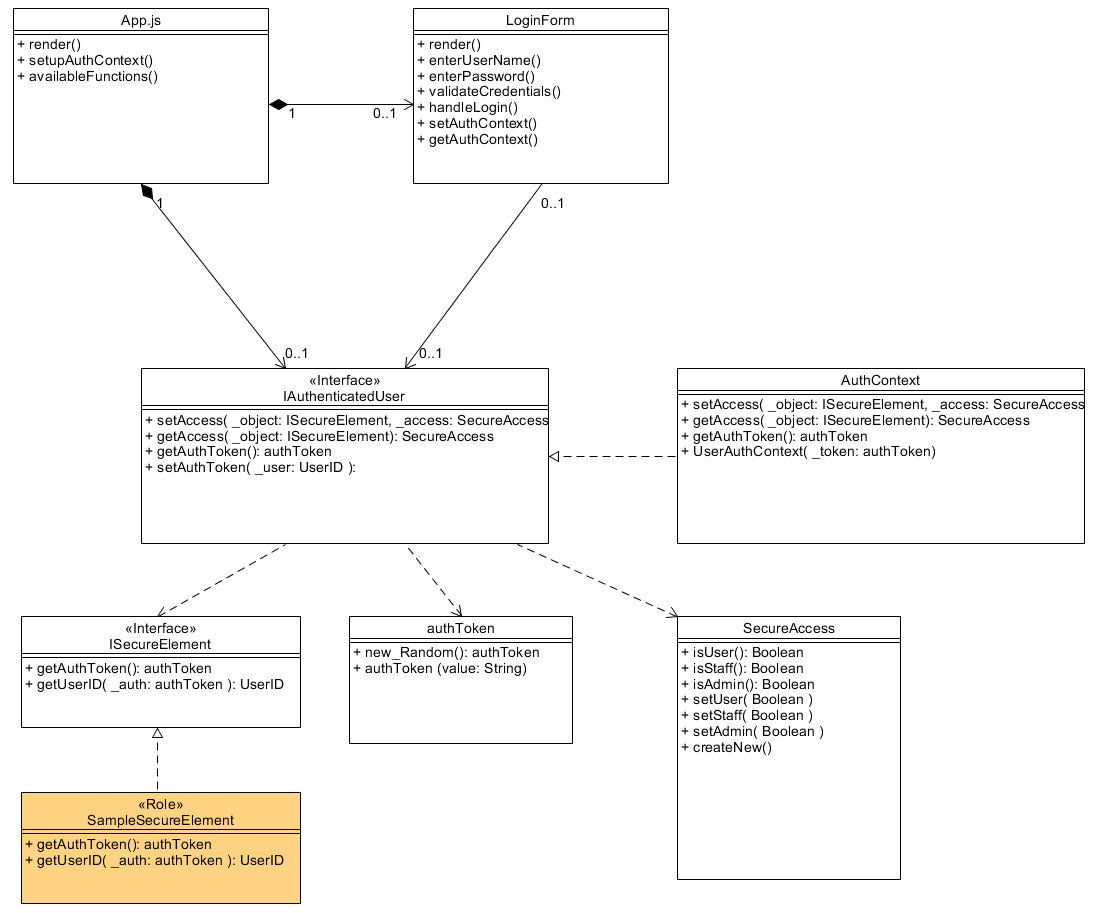
**Software also open to people who want to earn income and or for seller who looking for orders which are suitable for them.**

1. **Architectural Mechanism**
   1. **Analysis Mechanisms:**

* **Persistency: A means to make an element persistent (i.e., exist after the application that created it ceases to exist).**
* Security: A means to control access to an element.
  1. **Analysis-to-Design-to-Implementation Mechanisms Map:**

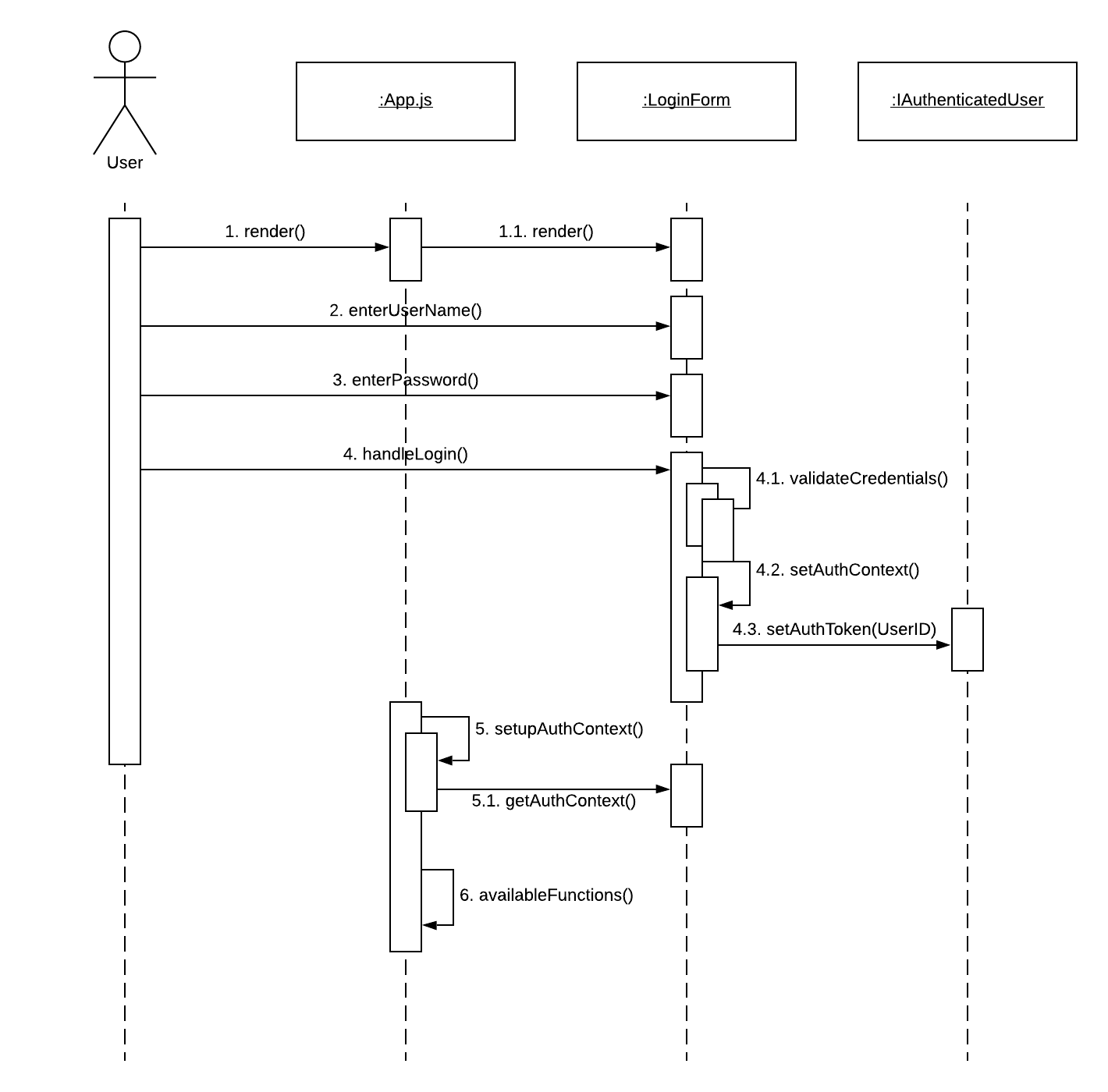
|  |  |  |
| --- | --- | --- |
| **Analysis Mechanism** | **Design Mechanism** | **Implementation Mechanisms** |
| **Persistency** | **RDBMS** | **JDBC** |
| **Security** |  | **Reverse – Engineered Secure.java elements.** |

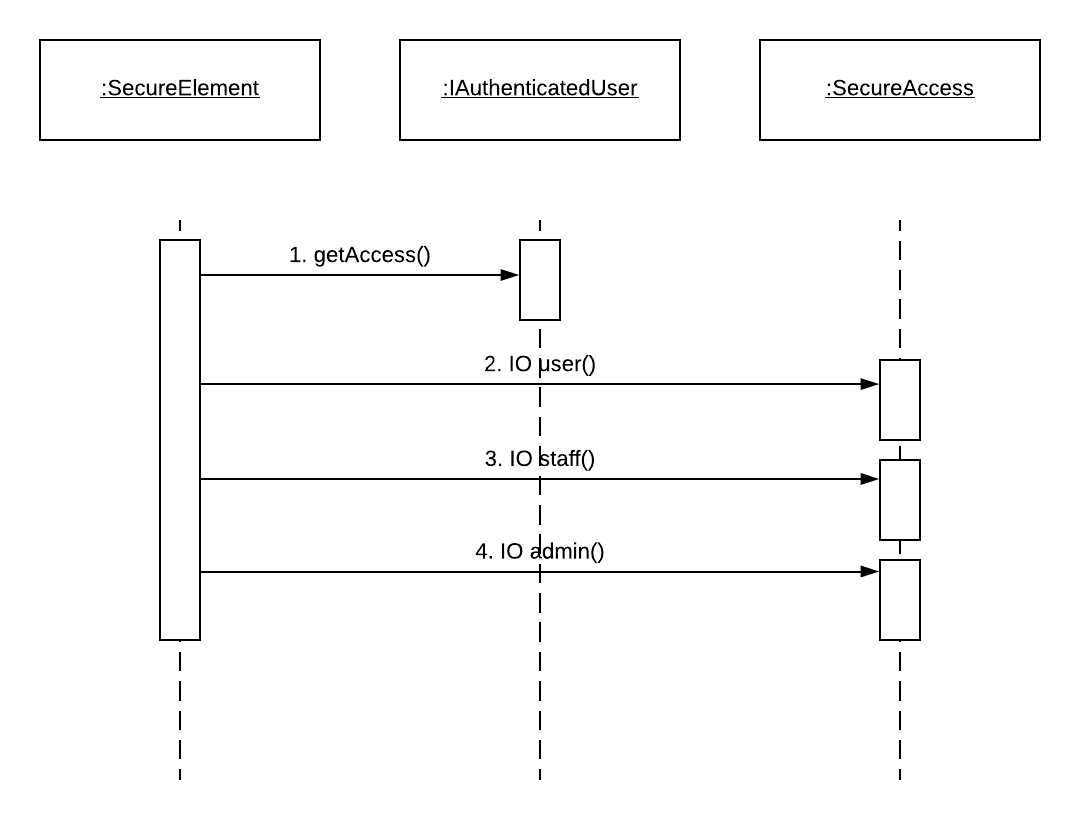
* 1. **Security Mechanism:**

**Static View: Security**

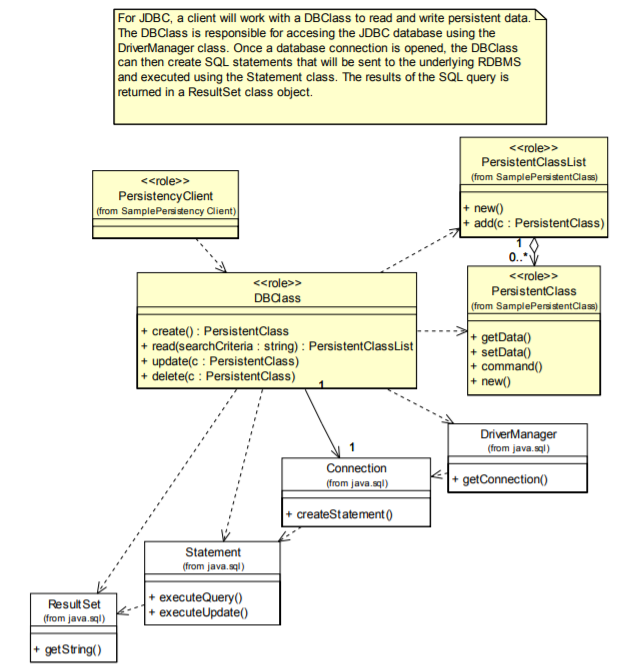
**Class Descriptions:**

* **LoginForm: Analysis Mechanisms: Security; Self - explanatory**
* **IAuthenticatedUser: Analysis Mechanisms: Security; Main interface on handling User’s SecureAccess and AuthToken**
* **AuthContext: Analysis Mechanisms: Security; Provide Contexts for SecureElements**
* **IsecureElement: Analysis Mechanisms: Security; Interface for Elements requiring SecureAccess**
* **authToken: Analysis Mechanisms: Security; Token for proof of access**
* **SecureAccess: Analysis Mechanisms: Security; Self – explanatory**

**Dynamic View: Secure User Login**

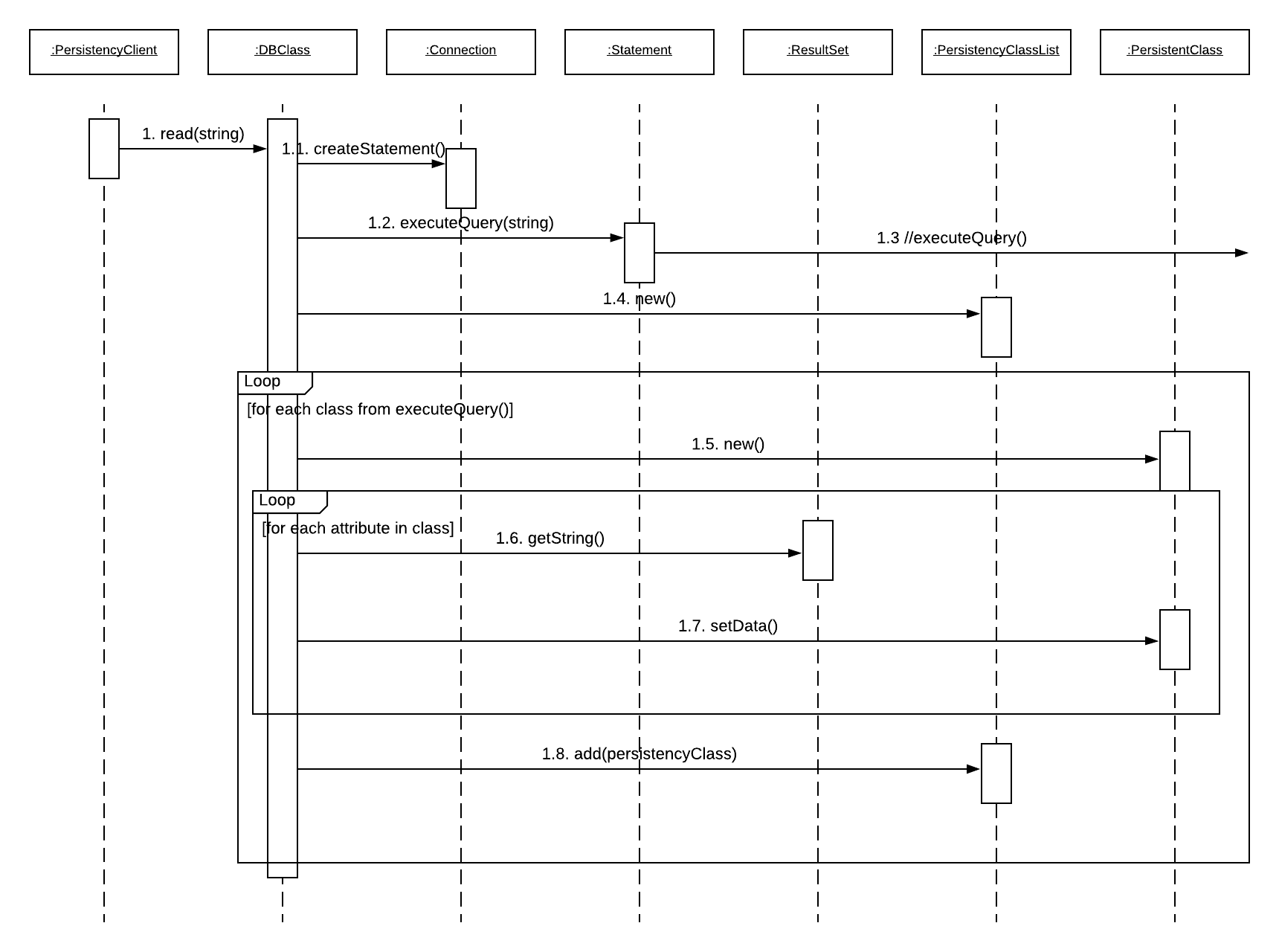
**Dynamic View: Secure Access**

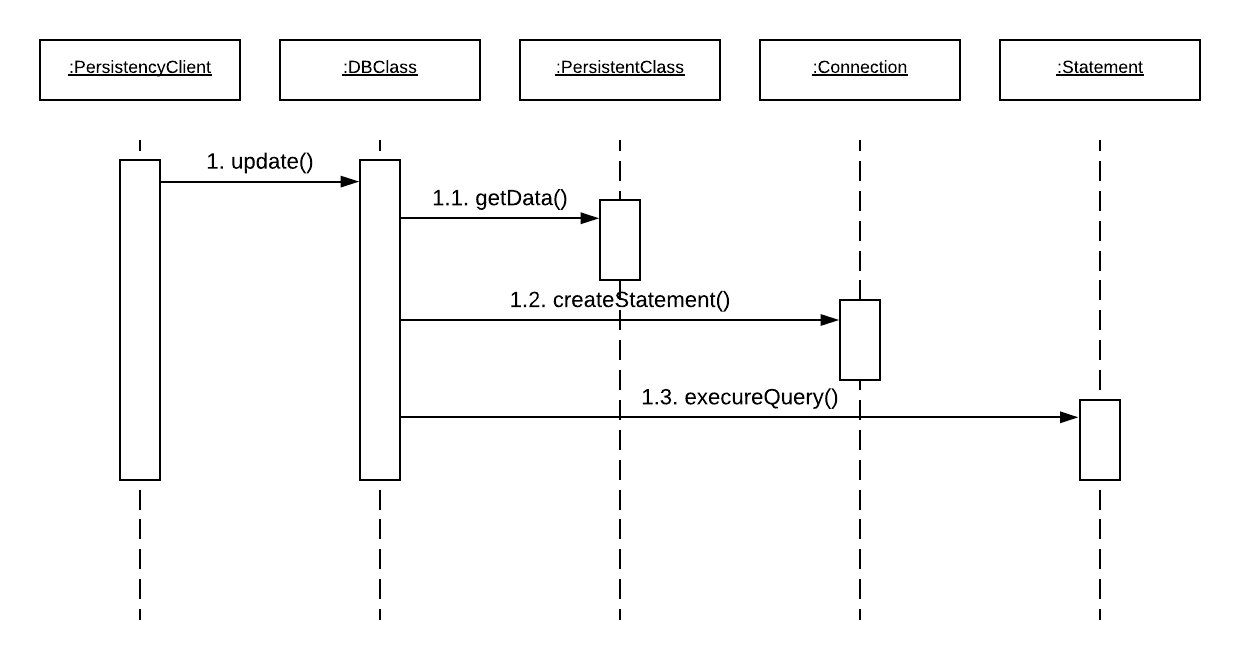
* 1. **Persistency Mechanism:**

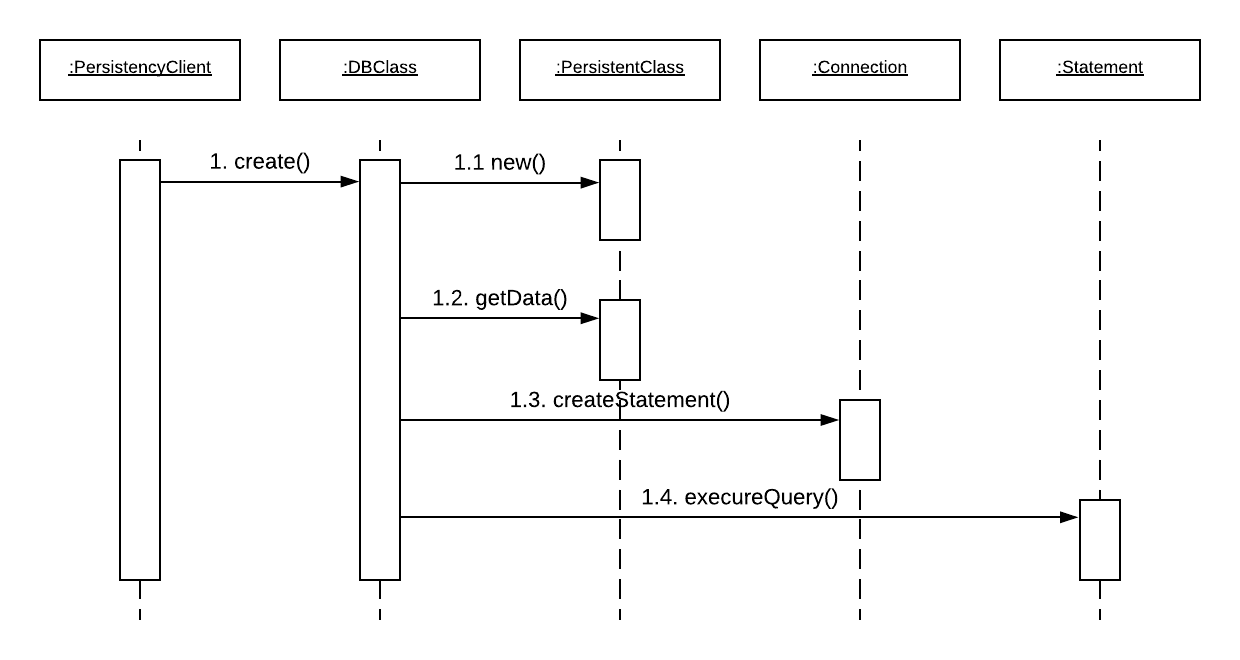
**Static View: Persistency JDBC**

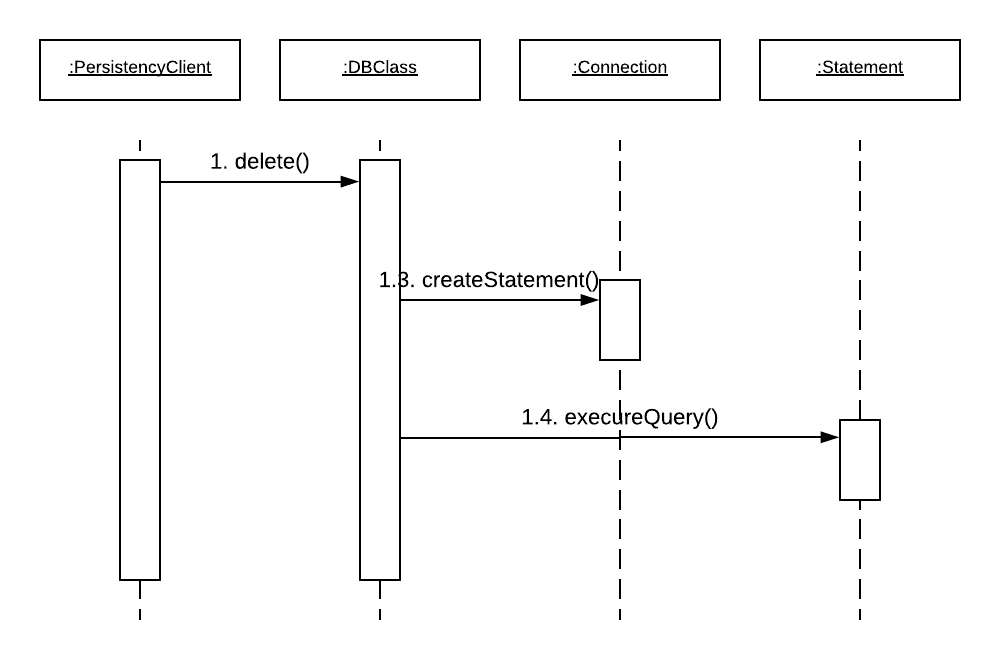
**Class Descriptions:**

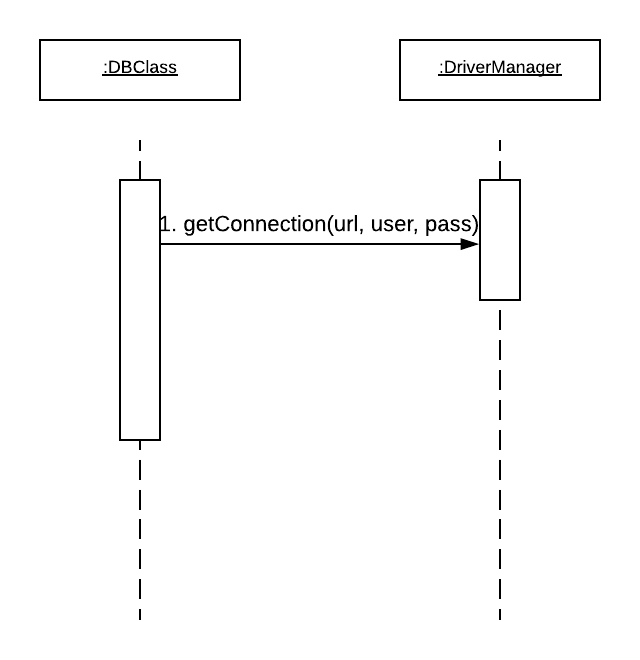
* **PersistencyClient:** An example of a client of a persistent class.
* **PersistentClass:** An example of a class that's persistent
* **PersistentClassList: Self - explanatory**
* **Statement:** The class used for executing a static SQL statement and obtaining the results produced by it. SQL statements without parameters are normally executed using Statement objects.
* **DBClass:** A sample of a class that would be responsible for making another class persistent. Every Class that's persistent will have a corresponding DBClass (e.g., Student will have a DBStudent class).
* **Connection:** A connection (session) with a specific database. Within the context of a Connection, SQL statements are executed, and results are returned.
* **ResultSet:** A ResultSet provides access to a table of data. A ResultSet object is usually generated by executing a Statement.
* **DriverManager:** The basic service for managing a set of JDBC drivers.

**Dynamic View: JDBC RDBMS Read**

**Dynamic View: JDBC RDBMC Update**

**Dynamic View: JDBC RDBMC Create**

**Dynamic View: JDBC RDBMC Delete**

**Dynamic View: JDBC RDBMC Initialize**