North South University
Department of Electrical and Computer Engineering
CSE327 - Software Engineering
Midterm, Summer 2020
Total Marks - 40

Name:	**Mohammed Rakib**
Student ID:	**1731176042**
Section:	**01**
Date:	**31st August 2020**

Instructions: All Questions are based on the following case study.

Case study:

P1	You are required to develop a versatile online data annotation tool. The main task of a data annotation tool is to assign a label to a sample, and repeat the process for samples in a data set. Most data annotation tools are restricted to samples of one type. However, you are required to design the system so that it can handle text samples, document samples, images, audio clips, and series data. In the future, other sample types may be added. Each sample type can have different presentations, e.g. text samples can be shown in total, documents must be shown in parts, images can be viewed, audio must be played on the speakers, etc. Labels can also be of different types, but for now, you need to handle integers, real numbers, and text labels. Each sample in a dataset can be associated with one or more labels. A dataset usually has many samples that need to be associated with a label by human beings, which is the crux of an annotation task.
P2	The data annotation tools has multiple types of users. The administrators are the super users and have access to almost any functionality. However, their main task is related to discipline, which means reviewing offence reports and blocking offending users. They (the admins) can also create new user accounts and give users discounts in their data collection efforts.
P3	Anyone who wants to annotate their dataset can use this system. They must sign up to the system with their email addresses or log in if they already have an account. Once logged in, the system should give the users the option to create a new annotation task. The cost of an annotation task depends on the number of samples and the number of annotations per sample required by the creator. This cost is deducted from the task creator's credit, which he/she can purchase. When creating the new annotation task, the system will ask

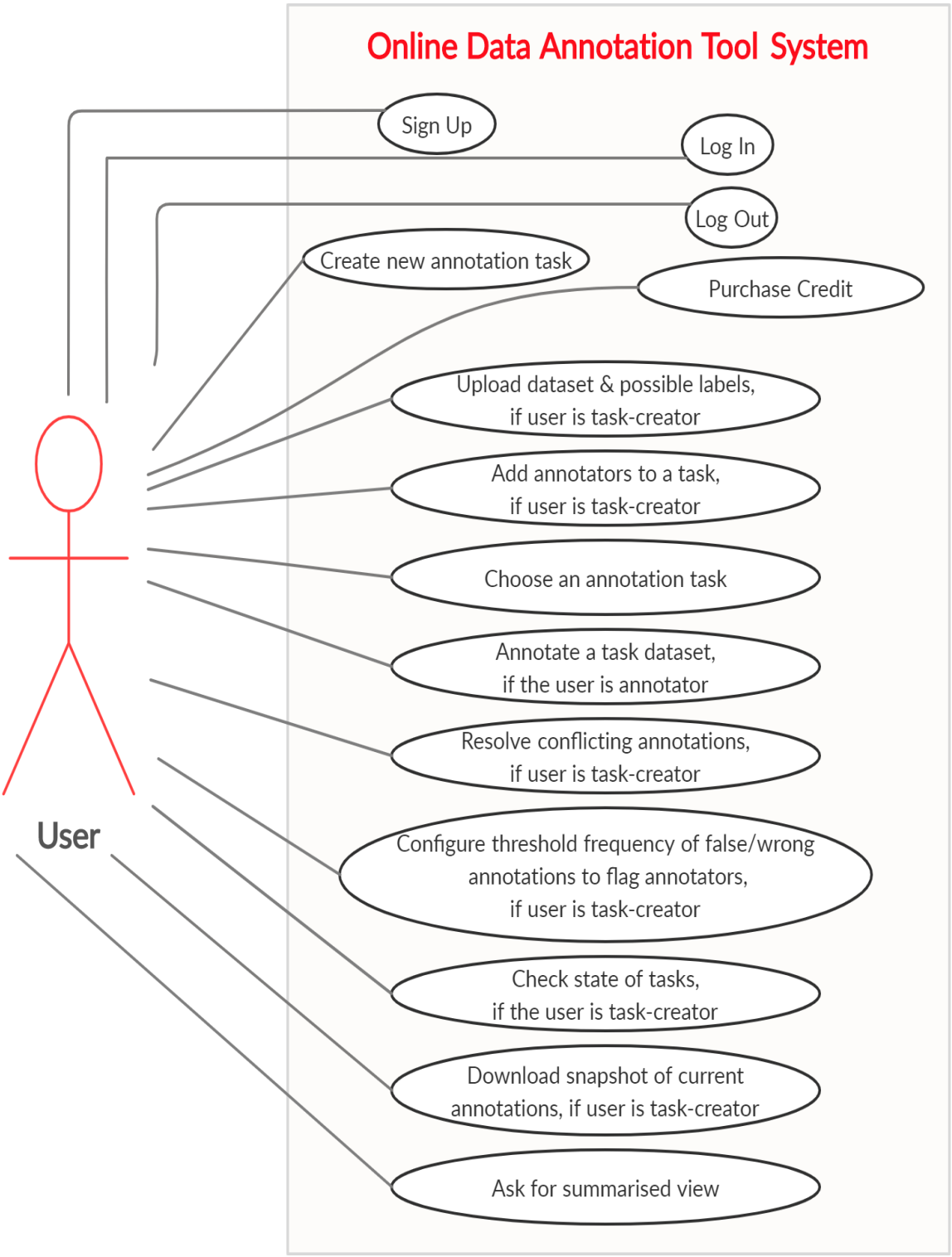
	for the data set, which has to be uploaded as well as the possible labels. There should also be an option to add a list of annotators to each annotation task. Tasks that do not have a list of annotators are considered to be open and are open for any annotator.
P4	The actual annotation task is a multi-step process per sample. This involves the retrieval of a sample with associated metadata, the display of the sample, the actual input of the annotation, a confirmation that the annotation is indeed the one desired by the annotator, and finally storing the annotation information (data sets, sample number, annotation, annotator). These steps are fixed. However, the actual implementation of each step may vary (i.e. the retrieval of a document is different from the retrieval of an image).
P5	When an annotator logs into the system, he/she is presented with a list of annotation tasks. After choosing anyone, the system shows one or more samples of the tasks data set and presents labels (for each one) for the annotator to annotate. The inputs are stored in the database. The system should ensure that more than one annotator is annotating each sample in the dataset.
P6	The system should keep track of conflicting annotations of the same sample. In such cases, if a correct annotation is decided (usually by the task creator), then the system should record the others as false/wrong annotation. An annotator who has a high frequency of false/wrong annotations should be flagged as an offender and a notification has to be sent to the administrator(s). The threshold frequency for this task should be configurable on a per annotation task basis.
P7	The task creator can log in at any time and check the state of the tasks. They can download a snapshot of the current annotations or ask for a summarised view, which essentially gives a list of samples and the percentage of responses for each label.

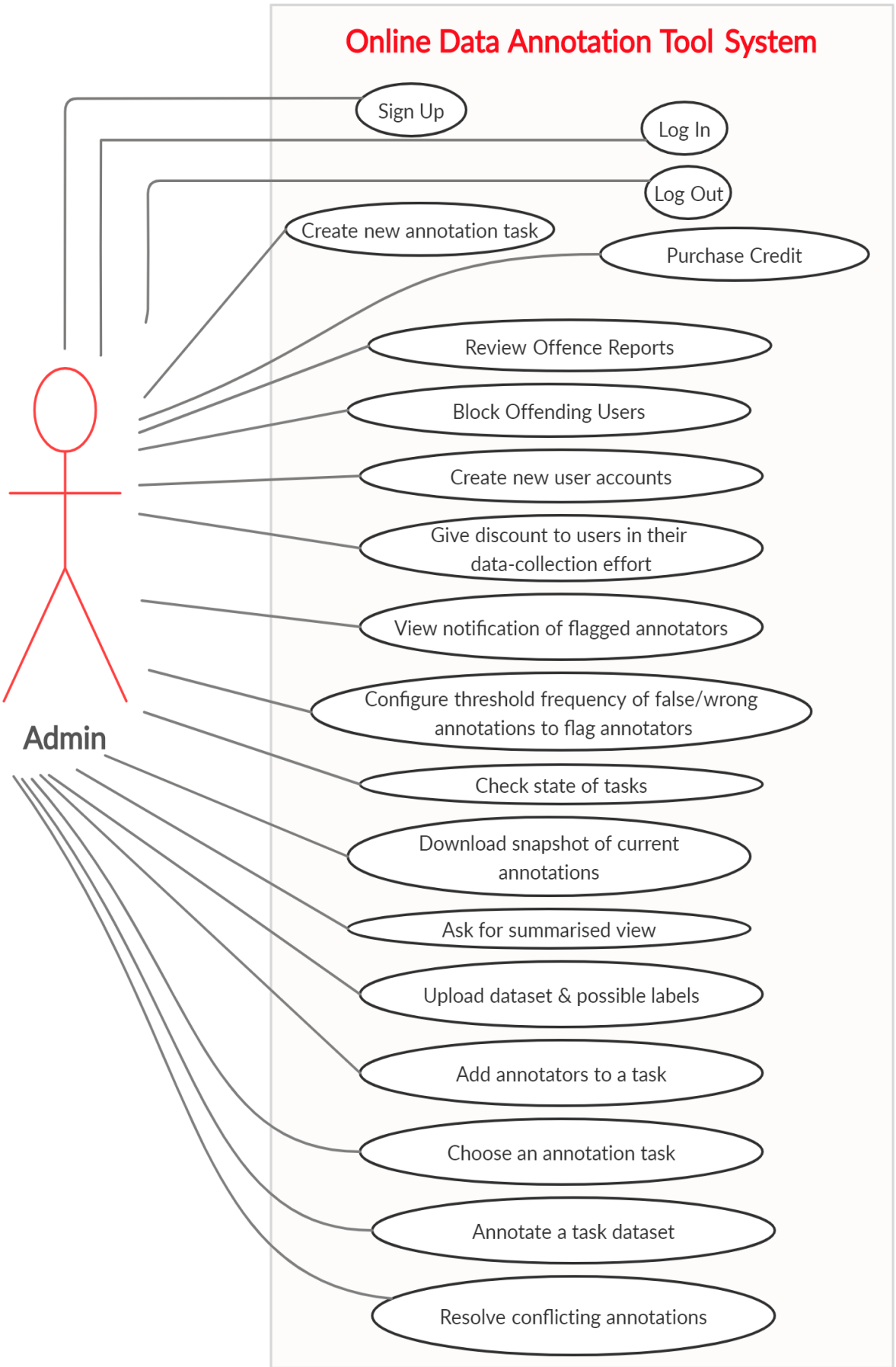
Questions:

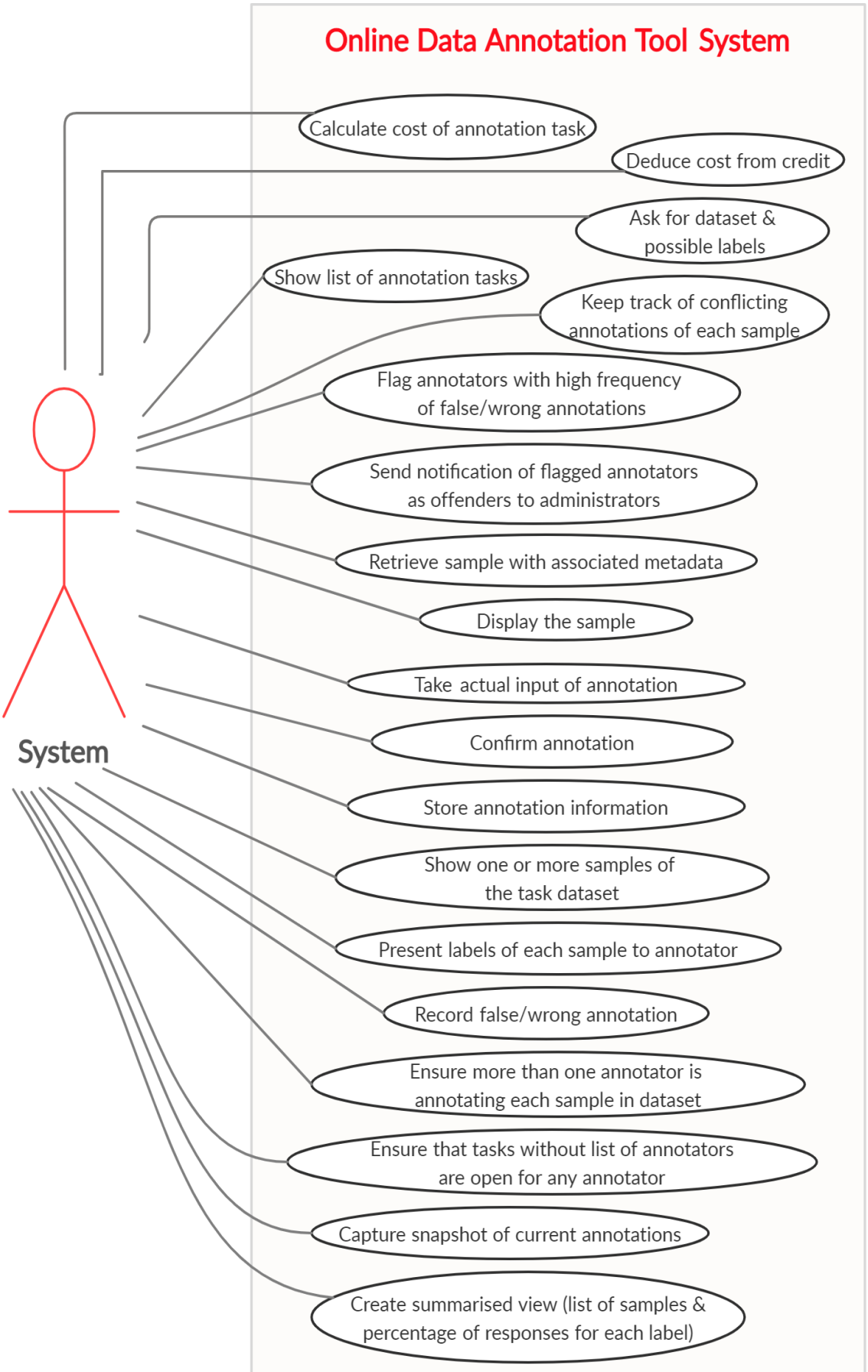
Q1. Draw the use case diagram of the system. (10)

Answer to the question no. 1

The use case diagrams of the system are given below:



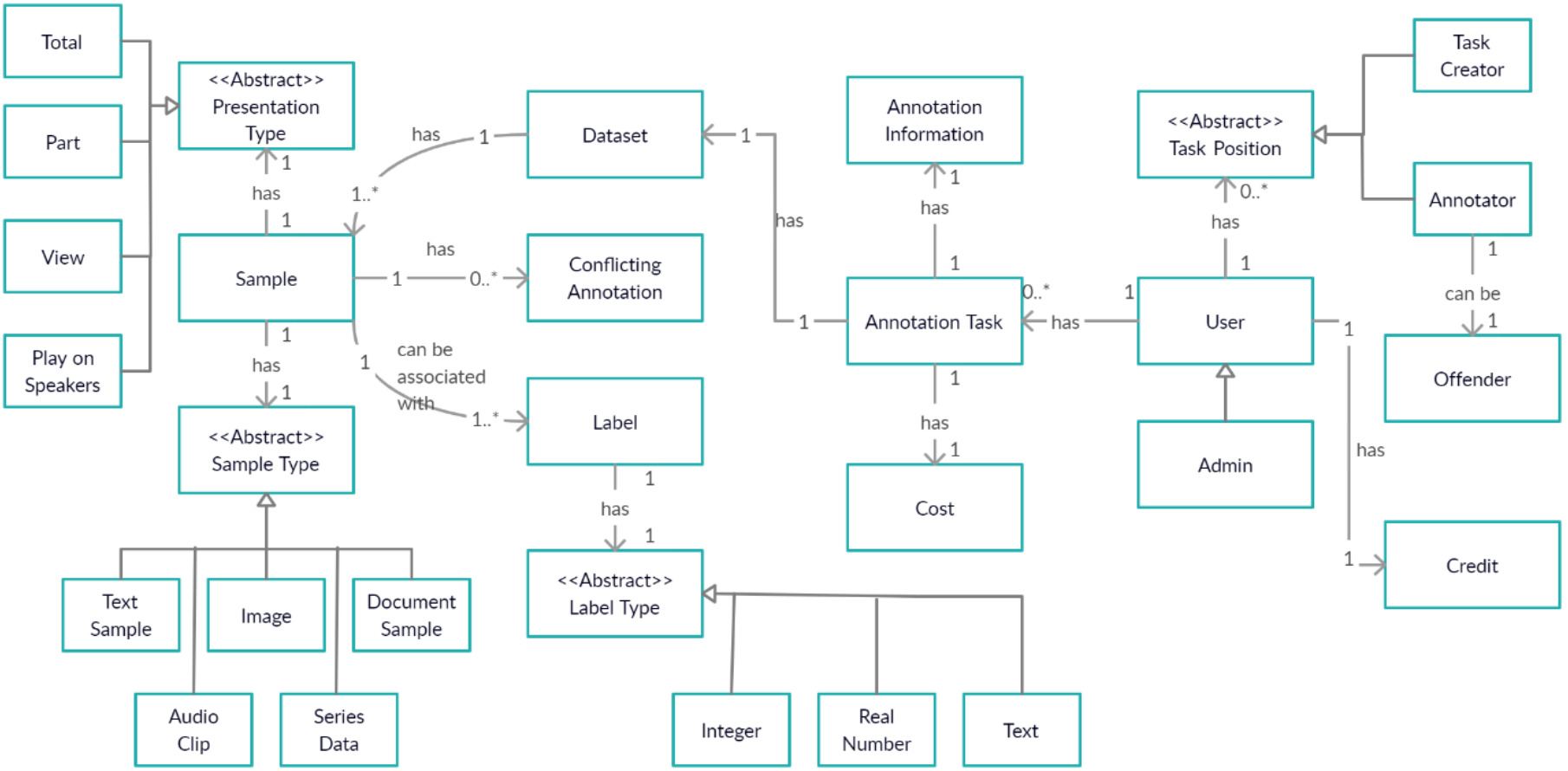

Online Data Annotation Tool System
Sign Up
Log In
Log Out
Create new annotation task
Purchase Credit
Review Offence Reports
Block Offending Users
Create new user accounts
Give discount to users in their data-collection effort
View notification of flagged annotators
Configure threshold frequency of false/wrong annotations to flag annotators
Check state of tasks
Download snapshot of current annotations
Ask for summarised view
Upload dataset & possible labels
Add annotators to a task
Choose an annotation task
Annotate a task dataset
Resolve conflicting annotations
Admin


Online Data Annotation Tool System
Calculate cost of annotation task
Deduce cost from credit
Ask for dataset & possible labels
Show list of annotation tasks
Keep track of conflicting annotations of each sample
Flag annotators with high frequency of false/wrong annotations
Send notification of flagged annotators as offenders to administrators
Retrieve sample with associated metadata
Display the sample
Take actual input of annotation
Confirm annotation
Store annotation information
Show one or more samples of the task dataset
Present labels of each sample to annotator
Record false/wrong annotation
Ensure more than one annotator is annotating each sample in dataset
Ensure that tasks without list of annotators are open for any annotator
Capture snapshot of current annotations
Create summarised view (list of samples & percentage of responses for each label)
System

Q2. Assuming an MVC architectural pattern, draw a high-level class diagram depicting your proposed design for the model classes of the system. (10)

Answer to the question no. 2

High-level Class Diagram of Model Classes of Online Data Annotation Tool System

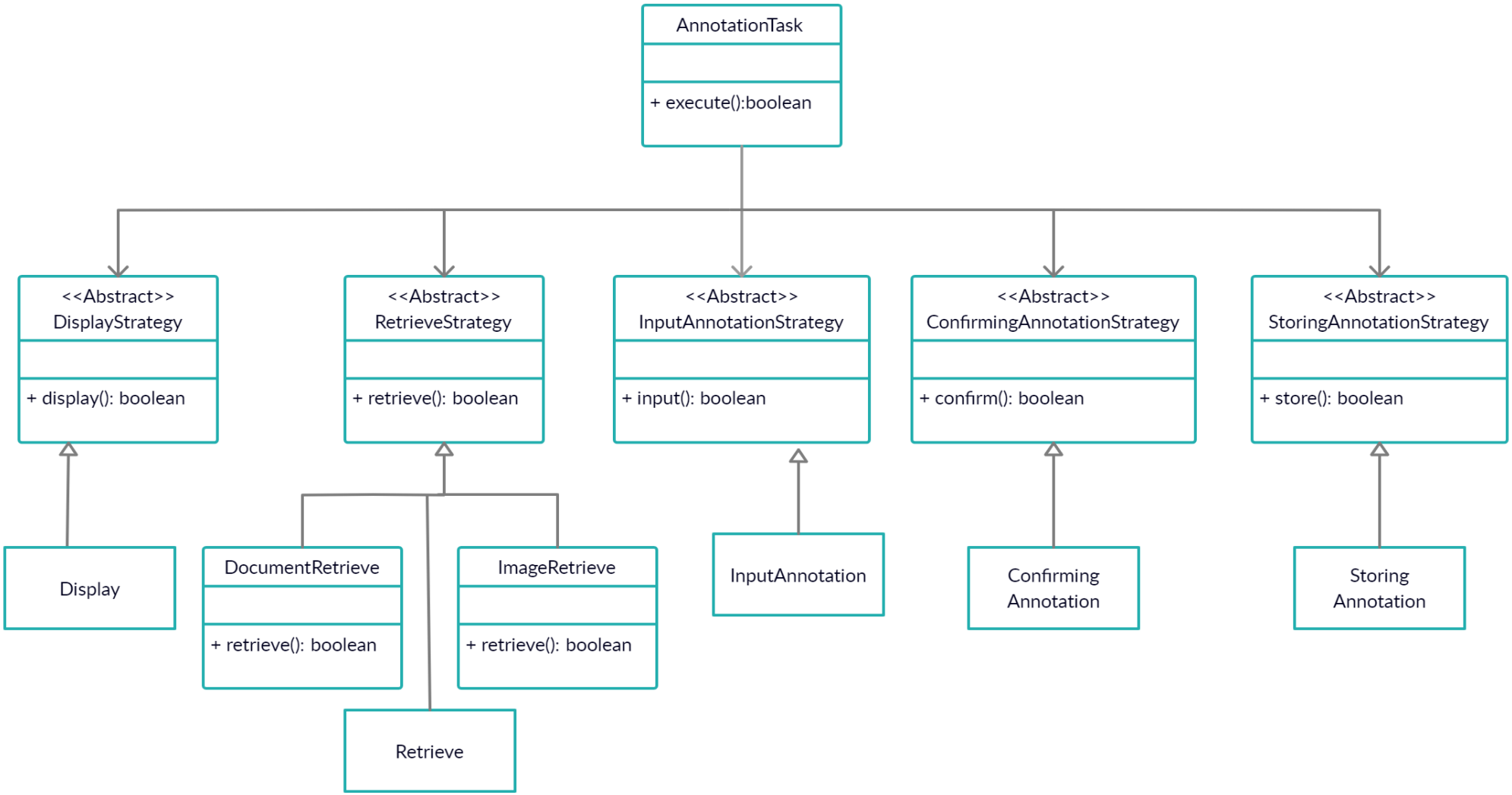


Q3. Draw a high-level UML class diagram depicting the design of your solution to the problem described in paragraph 4 (P4). (10)

Answer to the question no. 3

High-level UML Class Diagram depicting the design of solution to P4.

I have used Strategy Design Pattern



Q4. Write the code for the controller responsible for *annotating an image sample.* (10)
Your code must be consistent with your answers for Questions 2 and 3. In addition, your code must clearly show the class, method name, parameters, and workflow in order to be eligible for any marks.

Answer to the question no. 4

Code for annotating an image sample:

```
public class AnnotationTask {

	RetrieveStrategy rs;
	DisplayStrategy ds;
	InputAnnotationStrategy ias;
	ConfirmingAnnotationStrategy cas;
	StoringAnnotationStrategy sas;

	public AnnotationTask(RetrieveStrategy rs, DisplayStrategy ds,
		  InputAnnotationStrategy ias, ConfirmingAnnotationStrategy cas,
		  StoringAnnotationStrategy sas) {
		this.rs = rs;
		this.ds = ds;
		this.ias = ias;
		this.cas = cas;
		this.sas = sas;
	}

	public boolean execute()
	{
		if(this.rs.retrieve())
		{
			if(this.ds.display())
			{
				if(this.ias.input())
				{
					if(this.cas.confirm())
					{
						return this.sas.store();
					}
				}
			}
		}
	   return false;
	}
}


public class Main {
	public static void main(String[] args) {

		AnnotationTask at = new AnnotationTask(new ImageRetrieve(),
		  new Display(), new InputAnnotation(), new ConfirmingAnnotation(),
		  new StoringAnnotation());

		at.execute();
	}
}
```

THE-END