**CS 3365 Software Engineering I**

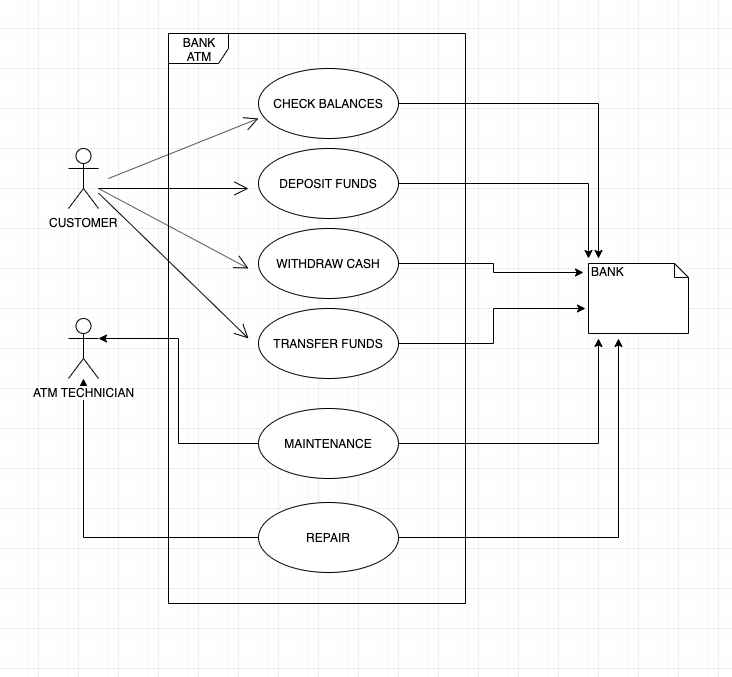
**Homework 02**

**Due Date: 11:59 pm, Wednesday, 10/23/2019**

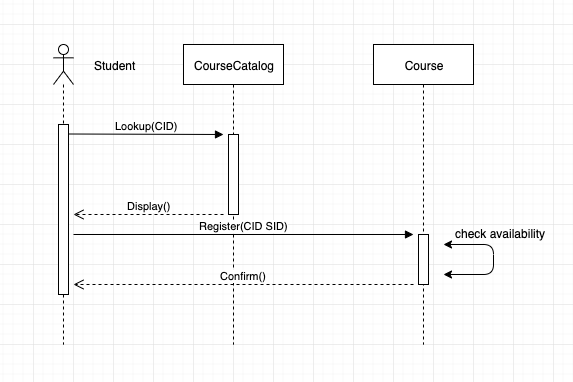
Name: \_\_\_\_\_Chen\_Zhang\_\_\_\_\_\_\_\_\_\_

**NOTE: use draw.io or similar tools to draw your UML diagrams then insert them into your answers.**

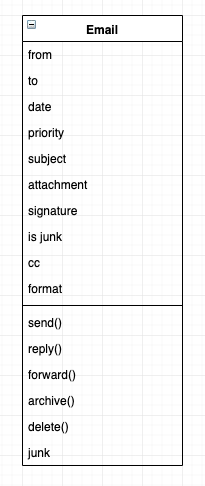
1. Using your knowledge of how an ATM is used, develop a set of use cases (minimum 4 use cases) that could serve a basis for understanding the requirements for an ATM system.



2. Develop a sequence diagram showing the interactions involved when a student registers for a course in a university. Courses may have limited enrollment, so the registration process must include checks that places are available. Assume that the student is already authorized to access an electronic course catalog to find out about available courses and is also authorized to register an available course.



3. Based on your knowledge of an email message, model an EmailMessage class in a class diagram. Minimum required number of attributes: 10. Minimum required number of methods: 6



4. Name three (3) benefits of class diagrams.

1. Illustrate data models for information systems, no matter how simple or complex.
2. Better understand the general overview of the schematics of an application.
3. Visually express any specific needs of a system and disseminate that information throughout the business.

5. What is Model-Driven Architecture (MDA)? Name four (4) factors that have limited the adoption of MDA.

MDA is a model-focused approach to software design and implementation that uses a subset of UML models to describe a system.

1. Specialized tool support is required to convert models from one level to another.
2. There is limited tool availability and organizations may require tool adaptation and customization to their environment.
3. For the long-lifetime systems developed using MDA, companies are reluctant to develop their own tools or rely on small companies that may go out of business
4. Models are a good way of facilitating discussions about a software design. However, the abstractions that are useful for discussions may not be the right abstractions for implementation.