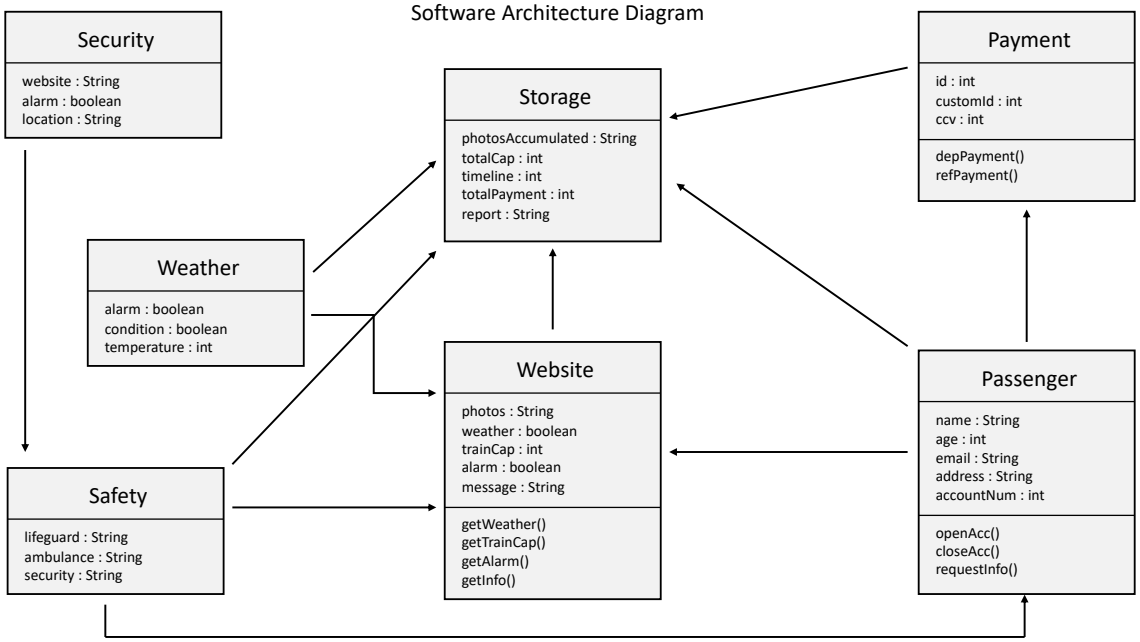
Data Management Strategy

Group 6: Chanine Malong, Isaac Pompa, Janelle Kwofie, Maron Vincent Ejanda

Software Architecture Diagram



Data Management Dictionary

Storage

photosAccumulated : String
totalCap : int
timeline : int
totalPayment : int
report : String
website : String
alarm : boolean
location : String
condition : boolean
temperature : int
lifeguard : String
ambulance : String
security : String
photos : String
weather : boolean
trainCap : int
message : String
id : int
customId : int
ccv : int
name : String
age : int
email : String
address : String
accountNum : int

Weather Database Description:

The main concept for the weather database is to store and collect the upcoming and current weather for the train tracks, this will be the key if the bluffs will be safe for the day. The weather diagram will be SQL because it can contain a good amount of data regarding the temperature of location in the train station. The alarm and the condition should not be too much to store because it only says if there is a good or a bad weather. This will have a trigger bool alarm ()- gives the alarms to the security and the safety for the weather, bool condition ()- a the current condition for the weather and in temperature ()- ties into the condition for the train ride and for the bluffs. Overall, the weather will be SQL because it represents a straightforward attribute for the objects and will only contain data for certain time for the different temperature in the location of the train station.

Website Database Description:

The website will have a lot of data to store and to read throughout the day. The website will be a NoSQL database because it will contain all the data gathered from the safety, security, weather, payment, and passenger. The website will be split into 4 different documents; the weather, train capacity, alarm and the information that will be passed down to the upcoming people for the train ride. The website will consist of the following, string photos ()- large amount of data that shows the progression of the bluffs, Boolean weather () – from the weather database and its contents, int TrainCap () – the current capacity of the train and will be outputting a continuous number depends on the people will ride the train, and string message () – a good amount of data that will stored and sent out to the passengers .Due to the constant change in the incoming data, this will be a NoSQL because of the different people will be using the trains and the information that will be passed down to them. The website will hold the information regarding the management of the whole project, all the concepts and the contents will soon be stored in the storage documentation.

Safety Database Description:

The purpose of Safety is to notify lifeguards and ambulance the location of the unstable rocks and grant them access to the passengers data. The Safety database will be No-SQL due to the fact that Safety will be constantly updated because of the change in new data like different people on the train and different information passed down to the lifeguards and ambulance. This database will store String lifeguard, String ambulance, and String report. Safety database will also call storage to obtain the passengers information and store reports.

Passenger Database Description:

The role of Passengers is to carry information required by the user to take out a payment. Since Passenger is something that executes Payment it has a Many to One Relationship with Passenger and it as well as a Many to Many Relationship with Website where Passengar develops Website. Furthermore, when it comes to the specific type that the Passengar Database is because of its size and relational goals with Website and Passanger, it is NoSQL. As it not only stores: name : String, age : int, email : String, address : String, and accountNum : int openAcc(), closeAcc(), and requestInfo(), but it sends this information to Website and Payment. These variables also would be Public mechanisms because they are foundational to the rest of the program running hence why it must be NoSQL. To be more technical especially since the Passengar is the main user of the application their data is a necessity and needs to be explored in a detailed manner.

Payment Database Description:

The role of Payment is to carry information required by the user to take out a Payment. Since Payment is something done by the Passenger it has a Many to One Relationship with Passenger and specifically is executed by the Passenger. Furthermore, when it comes to the specific type that the Payment Database is because of its size and relational goals, it is SQL. As it only stores: id : int, customId : int, ccv : int, depPayment(), refPayment(). Which are all private variables and methods belonging to Payment, that will not interact with any other database's objects.