13th November 2024

* We decided on the project idea which was a bank management system using java
* And decide to spit it down the middle and do two packages each for a total 4 packages

20th November 2024

* Started working on the UML diagram
* Decided the names and theme of the 4 packages Booking, Transaction, Customers, Insurance

27th November 2024

* Shae defines the classes for each package in the UML, and finalizes design e.g. extend arrows and associations arrows
* Matthew codes the Booking and Transactions packages and classes
* Shae codes the Customers and Insurance packages and classes

4th December 2024

* Matthew finishes the team’s document
* Matthew Changes the UML to fully match his code
* Shae Changes the UML to fully match his code
* Review the code before submission

Matthew Code Location:

* The Classes are Booking, Staff, BookingInfo in Booking & Transaction, Withdraw, Deposit, Loan in Transaction
* Withdraw and Deposit inherit from Transaction and BookingInfo inherit from Booking
* Staff is an object in Booking called banker
* Booking uses blank constructor and staff, deposit, withdraw uses a normal constructor, transaction has a constructor to allow for chaining
* Constructor chaining is used for inheritance in withdraw and deposit from Transaction
* currentAmount is public, and there’s a mix of private and protected variables used throughout both packages
* Every private variable has a get/setter as well as protected date in booking to allow it to be added even with a blank constructor
* The packages are called Bookings and Transactions
* Enum are in both Booking and Transaction Package
* Transaction and Booking class is abstract
* BookingInfo is a final class
* NegativeNumExepection is the exception class and is applied to loan if the number is less than 0
* Interface is the CalAmount class and is applied to withdraw and deposit for calculation