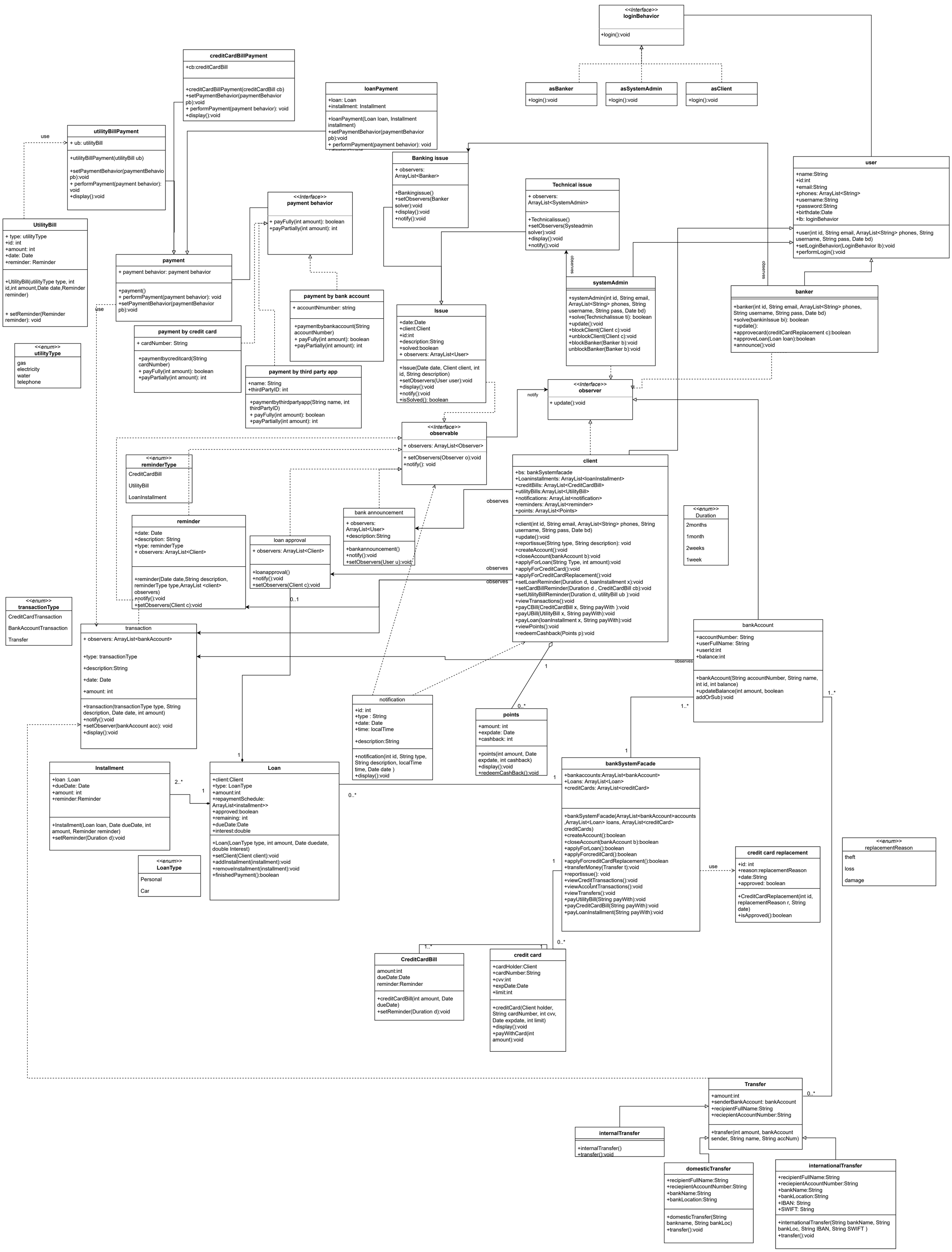
<<Interface>>
loginBehavior
+login():void
asBanker
+login():void
asSystemAdmin
+login():void
asClient
+login():void
creditCardBillPayment
+cb:creditCardBill
+creditCardBillPayment(creditCardBill cb)
+setPaymentBehavior(paymentBehavior pb):void
+ performPayment(payment behavior): void
+display():void
loanPayment
+loan: Loan
+installment: Installment
+loanPayment(Loan loan, Installment installment)
+setPaymentBehavior(paymentBehavior pb):void
+ performPayment(payment behavior): void
use
utilityBillPayment
+ ub: utilityBill
+utilityBillPayment(utilityBill ub)
+setPaymentBehavior(paymentBehavio pb):void
+ performPayment(payment behavior): void
+display():void
Banking issue
+ observers: ArrayList<Banker>
+Bankingissue()
+setObservers(Banker solver):void
+display():void
+notify():void
user
+name:String
+id:int
+email:String
+phones: ArrayList<String>
+username:String
+password:String
+birthdate:Date
+lb: loginBehavior
+user(int id, String email, ArrayList<String> phones, String username, String pass, Date bd)
+setLoginBehavior(LoginBehavior lb):void
+performLogin():void
<<Interface>>
payment behavior
+ payFully(int amount): boolean
+payPartially(int amount): int
Technical issue
+ observers: ArrayList<SystemAdmin>
+Technicalissue()
+setObservers(Systeadmin solver):void
+display():void
+notify():void
UtilityBill
+ type: utilityType
+id: int
+amount: int
+date: Date
+reminder: Reminder
+UtilityBill(utilityType type, int id,int amount,Date date,Reminder reminder)
+ setReminder(Reminder reminder): void
payment
+ payment behavior: payment behavior
+payment()
+ performPayment(payment behavior): void
+setPaymentBehavior(paymentBehavior pb):void
observes
systemAdmin
+systemAdmin(int id, String email, ArrayList<String> phones, String username, String pass, Date bd)
+solve(Technichalissue ti): boolean
+update():void
+blockClient(Client c):void
+unblockClient(Client c):void
+blockBanker(Banker b):void
unblockBanker(Banker b):void
banker
+banker(int id, String email, ArrayList<String> phones, String username, String pass, Date bd)
+solve(bankinIssue bi): boolean
+update():
+approvecard(creditCardReplacement c):boolean
+approveLoan(Loan loan):boolean
+announce():void
payment by bank account
+ accountNmumber: string
+paymentbybankaccount(String accountNumber)
+ payFully(int amount): boolean
+payPartially(int amount): int
Issue
+date:Date
+client:Client
+id:int
+description:String
+solved:boolean
+ observers: ArrayList<User>
+Issue(Date date, Client client, int id, String description)
+setObservers(User user):void
+display():void
+notify():void
+isSolved(): boolean
payment by credit card
+ cardNumber: String
+paymentbycreditcard(String cardNumber)
+ payFully(int amount): boolean
+payPartially(int amount): int
<<enum>>
utilityType
gas
electricity
water
telephone
payment by third party app
+name: String
+thirdPartyID: int
+paymentbythirdpartyapp(String name, int thirdPartyID)
+ payFully(int amount): boolean
+payPartially(int amount): int
notify
<<Interface>>
observer
+ update():void
<<Interface>>
observable
+ observers: ArrayList<Observer>
+ setObservers(Observer o):void
+notify(): void
<<enum>>
reminderType
CreditCardBill
UtilityBill
LoanInstallment
client
+bs: bankSystemfacade
+Loaninstallments: ArrayList<loanInstallment>
+creditBills: ArrayList<CreditCardBill>
+utilityBills:ArrayList<UtilityBill>
+notifications: ArrayList<notification>
+reminders: ArrayList<reminder>
+points: ArrayList<Points>
+client(int id, String email, ArrayList<String> phones, String username, String pass, Date bd)
+update():void
+reportissue(String type, String description): void
+createAccount():void
+closeAccount(bankAccount b):void
+applyForLoan(String Type, int amount):void
+applyForCreditCard():void
+applyForCreditCardReplacement():void
+setLoanReminder(Duration d, loanInstallment x):void
+setCardBillReminder(Duration d , CreditCardBill cb):void
+setUtilityBillReminder(Duration d, utilityBill ub):void
+viewTransactions():void
+payCBill(CreditCardBill x, String payWith):void
+payUBill(UtilityBill x, String payWith):void
+payLoan(loanInstallment x, String payWith):void
+viewPoints():void
+redeemCashback(Points p):void
observes
<<enum>>
Duration
2months
1month
2weeks
1week
reminder
+date: Date
+description: String
+type: reminderType
+ observers: ArrayList<Client>
+reminder(Date date,String description, reminderType type,ArrayList <client> observers)
+notify():void
+setObservers(Client c):void
bank announcement
+ observers: ArrayList<User>
+description:String
+bankannouncement()
+notify():void
+setObservers(User u):void
loan approval
+ observers: ArrayList<Client>
+loanapproval()
+notify():void
+setObservers(Client c):void
observes
observes
0..1
<<enum>>
transactionType
CreditCardTransaction
BankAccountTransaction
Transfer
transaction
+ observers: ArrayList<bankAccount>
+type: transactionType
+description:String
+date: Date
+amount: int
+transaction(transactionType type, String description, Date date, int amount)
+notify():void
+setObserver(bankAccount acc): void
+display():void
bankAccount
+accountNumber: String
+userFullName: String
+userId:int
+balance:int
+bankAccount(String accountNumber, String name, int id, int balance)
+updateBalance(int amount, boolean addOrSub):void
observes
1
1..*
1..*
notification
+id: int
+type : String
+date: Date
+time: localTime
+description:String
+notification(int id, String type, String description, localTime time, Date date)
+display():void
0..*
points
+amount: int
+expdate: Date
+cashback: int
+points(int amount, Date expdate, int cashback)
+display():void
+redeemCashBack():void
1
1
Installment
+loan :Loan
+dueDate: Date
+amount: int
+reminder:Reminder
+Installment(Loan loan, Date dueDate, int amount, Reminder reminder)
+setReminder(Duration d):void
2..*
1
Loan
+client:Client
+type: LoanType
+amount:int
+repaymentSchedule: ArrayList<installment>>
+approved:boolean
+remaining: int
+dueDate:Date
+interest:double
+Loan(LoanType type, int amount, Date duedate, double Interest)
+setClient(Client client):void
+addInstallment(installment):void
+removeInstallment(installment):void
+finishedPayment():boolean
0..*
1
bankSystemFacade
+bankaccounts:ArrayList<bankAccount>
+Loans: ArrayList<Loan>
+creditCards: ArrayList<creditCard>
+bankSystemFacade(ArrayList<bankAccount>accounts ,ArrayList<Loan> loans, ArrayList<creditCard> creditCards)
+createAccount():boolean
+closeAccount(bankAccount b):boolean
+applyForLoan():boolean
+applyForcreditCard():boolean
+applyForcreditCardReplacement():boolean
+transferMoney(Transfer t):void
+reportissue(): void
+viewCreditTransactions():void
+viewAccountTransactions():void
+viewTransfers():void
+payUtilityBill(String payWith):void
+payCreditCardBill(String payWith):void
+payLoanInstallment(String payWith):void
use
credit card replacement
+id: int
+reason:replacementReason
+date:String
+approved: boolean
+CreditCardReplacement(int id, replacementReason r, String date)
+isApproved():boolean
<<enum>>
replacementReason
theft
loss
damage
<<enum>>
LoanType
Personal
Car
1
0..*
1..*
1
CreditCardBill
amount:int
dueDate:Date
reminder:Reminder
+creditCardBill(int amount, Date dueDate)
+setReminder(Duration d):void
credit card
+cardHolder:Client
+cardNumber:String
+cvv:int
+expDate:Date
+limit:int
+creditCard(Client holder, String cardNumber, int cvv, Date expdate, int limit)
+display():void
+payWithCard(int amount):void
Transfer
+amount:int
+senderBankAccount: bankAccount
+recipientFullName:String
+reciepientAccountNumber:String
+transfer(int amount, bankAccount sender, String name, String accNum)
0..*
internalTransfer
+internalTransfer()
+transfer():void
domesticTransfer
+recipientFullName:String
+reciepientAccountNumber:String
+bankName:String
+bankLocation:String
+domesticTransfer(String bankname, String bankLoc)
+transfer():void
internationalTransfer
+recipientFullName:String
+reciepientAccountNumber:String
+bankName:String
+bankLocation:String
+IBAN: String
+SWIFT: String
+internationalTransfer(String bankName, String bankLoc, String IBAN, String SWIFT)
+transfer():void