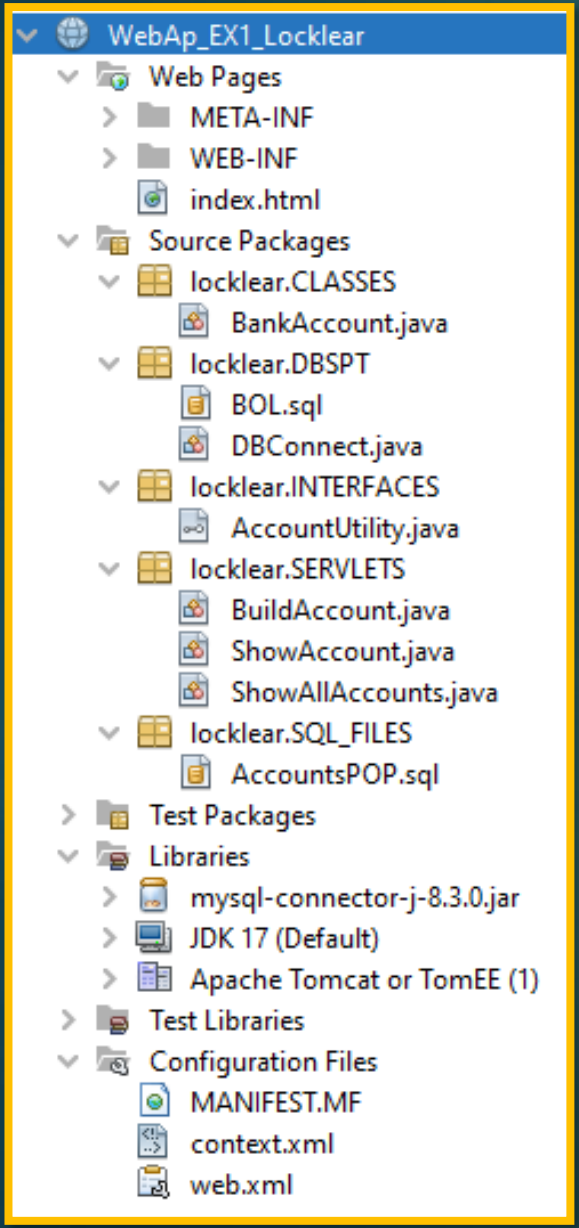
Dynamic Web Development in JSP
Spring 2024

Assignment 2: Simple Application Development

Professor HG Locklear

Program Structure



Organize your applications structure exactly as shown

BankAccount Class

BankAccount	
-String accountNumber -String lastName -String firstName -String taxIDNumber -double checkingBalance -double minCheckingBalance -double maxCheckingWithdrawal -double savingsBalance -double minSavingBalance -double maxSavingWithdrawal -double savingsInterestRate +static int AccountCount = 0	
BankAccount(String accountNumber, String lastName, String firstName, String taxIDNumber, double checkingBalance, double savingsBalance, double savingsInterestRate)	
+displayAccountInfo(): void +depositToChecking(double d): boolean +depositToSavings(double d): boolean +withdrawFromChecking(double d): boolean +withdrawFromSavings(double d): boolean	+toString(): String +toHTML(): String +toSQL(): String +writeToSQLFile(): void +writeToDatabase(): void +showAccount(): String

Constructor Specification

- You may only have one constructor with the signature specified and ALL data fields must be initialized by the constructor.

- Data fields not defined in the constructor parameter list are specified in the following manner:

- minCheckingBalance = 25% of checkingBalance
- maxCheckingWithdrawal = 40% of checkingBalance
- minSavingBalance = 55% of savingsBalance
- maxSavingWithdrawal = 20% of savingsBalance

BankAccount Methods

BankAccount (Method Specifications)
depositToChecking(double d): boolean Adds **d** to the **checkingBalance** and displays ***'$[d] deposited to Checking'*** …returns true
depositToSavings(double d): boolean Adds **d** to the **savingsBalance** and displays ***'$[d] deposited to Savings'*** …returns true
withdrawFromChecking(double d): boolean Subtract **d** from the **checkingBalance** after checking to ensure the amount of **d** is available and that the remaining **checkingBalance** is greater than the **minCheckingBalance** and displays ***'$[d] withdrawn from Checking'*** returns true… if either condition is not met then display message ***'Withdraw cannot be made'*** and returns false.
withdrawFromSavings(double d) : boolean Subtract **d** from the **savingsBalance** after checking to ensure the amount of **d** is available and that the remaining **savingsBalance** is greater than the **minSavingBalance** and displays ***'$[d] withdrawn from Savings'*** returns true… if either condition is not met then display message ***'Withdraw cannot be made'*** and returns false.

BankAccount Methods

BankAccount (Method Specifications)
toString(): String Returns a String representation of the BankAccount object formatted in any manner.
toHTML(): String Returns a HTML-formatted String representation of the BankAccount object formatted as shown on Slide 9.
toSQL(): String[ ] Returns a String array containing the SQL INSERT statements for the BankAccount object that can be used individually as queries to add the BankAccount object's information to the appropriate tables in the BOL database.
writeToSQLFile(): void Writes the Bank Account object's toSQL() String array elements to a file.
writeToDatabase(): void Utilizes the Bank Account object's toSQL() String Array as a queries to insert the BankAccount object into the BOL database.
showAccount(): void Returns a HTML-formatted String representation of the BankAccount object formatted as shown on Slide 10.

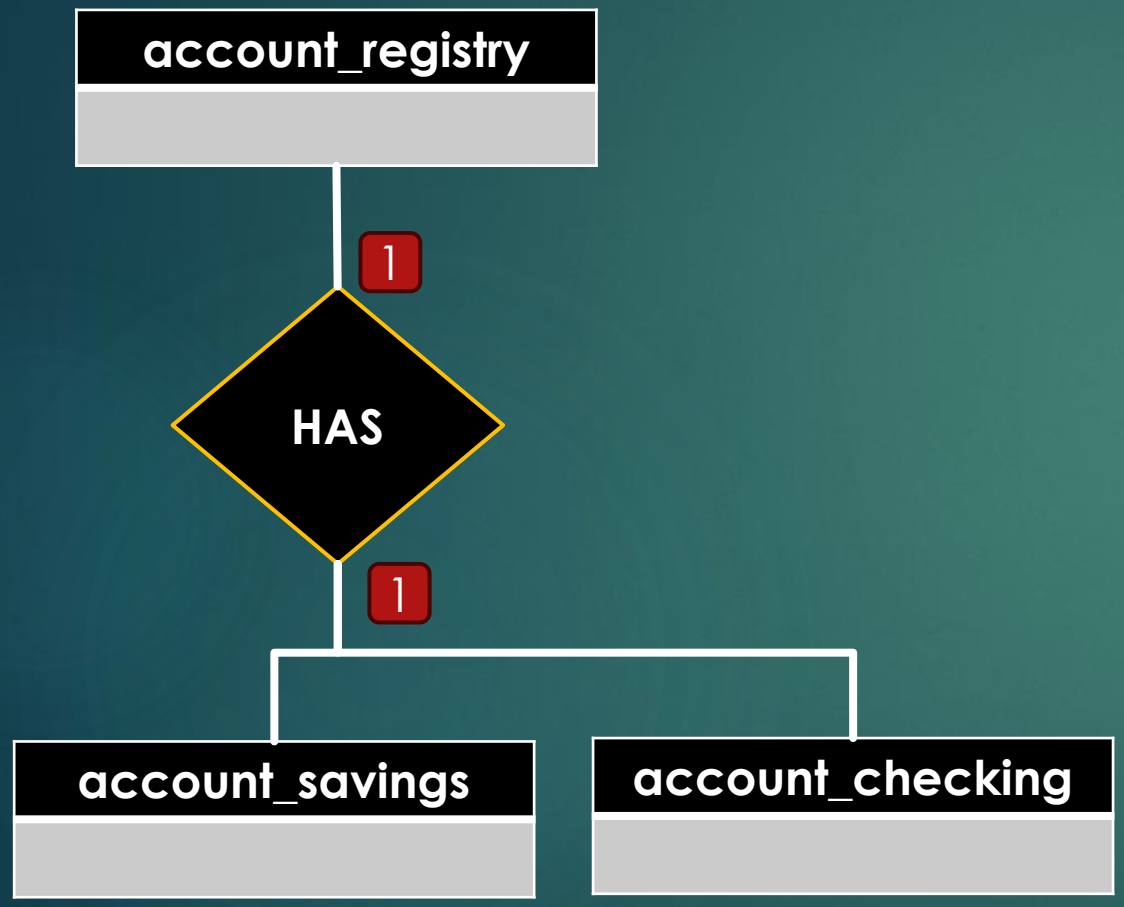
AccountUtility Interface

<<AccountUtility>>
+ static String dir = [your **SQL_FILES** directory] + static buildAccountFromQuery(String acct): BankAccount + static buildAllAccounts(): ArrayList<BankAccount>

AccountUtility (Method Specifications)
static buildAccountFromQuery(String acct): BankAccount Accepts an account number and creates a BankAccount object from the data in the BOL database corresponding to the specified account number and <u>returns</u> the BankAccount object
static buildAllAccounts(): ArrayList<BankAccount> Queries the BOL database and returns the data to create BankAccount objects for all accounts in the BOL database, stores each object in an ArrayList and <u>returns</u> the ArrayList of BankAccount objects.

BOL Database

- The Bank of Locklear database (BOL) has the schema shown below.



BOL Database Relations

- The Relations in the BOL database have the schema shown below.

Field	Type	Null	Key	Default	Extra
AccountNumber	varchar(50)	NO	PRI	NULL	
LastName	varchar(50)	YES		NULL	
FirstName	varchar(50)	YES		NULL	
TaxIDNumber	varchar(50)	YES		NULL	

account_registry

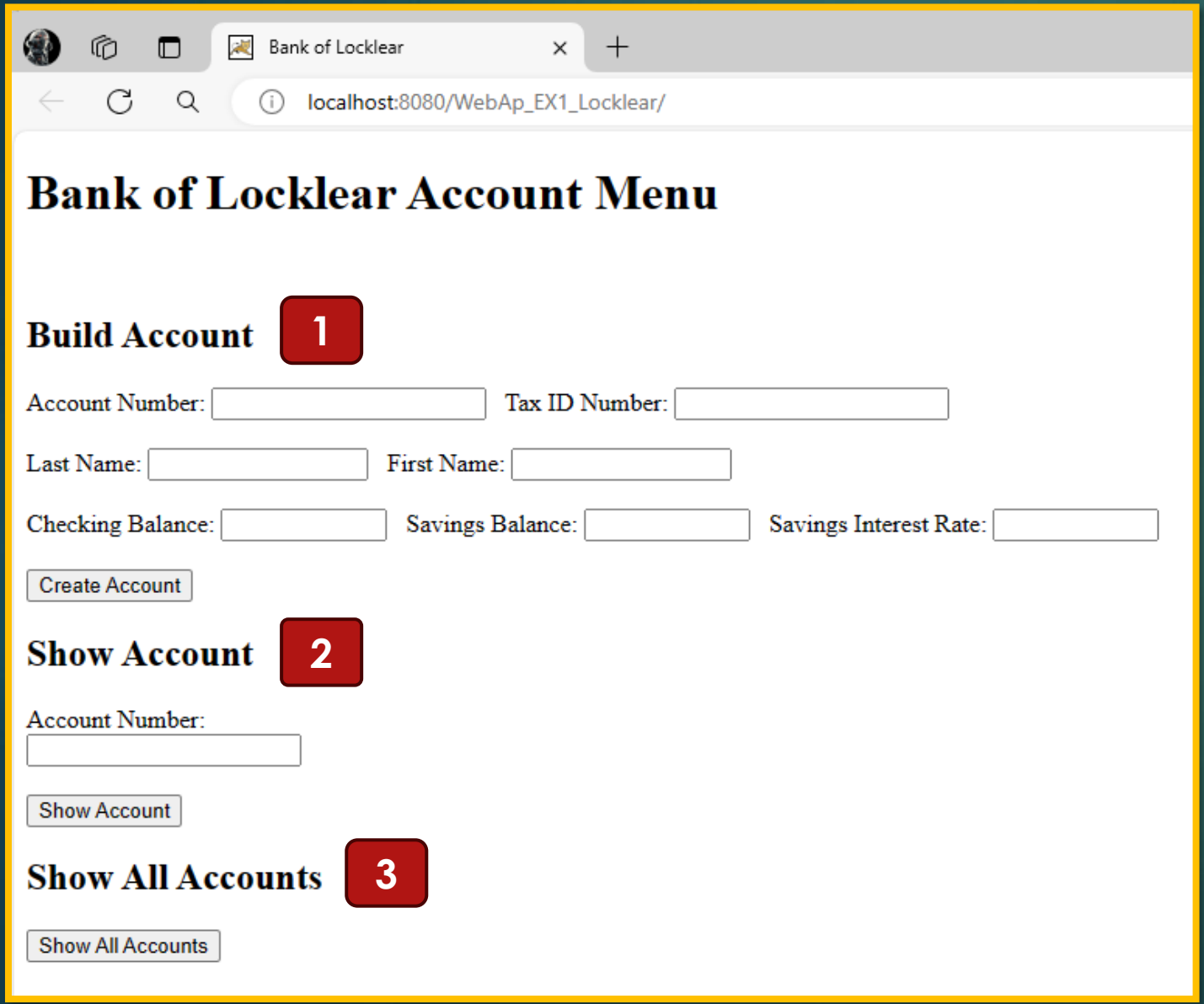
Field	Type	Null	Key	Default	Extra
AccountNumber	varchar(50)	NO	PRI	NULL	
CheckingBalance	varchar(50)	YES		NULL	
MinCheckingBalance	decimal(10,2)	YES		NULL	STORED GENERATED
MaxCheckingWithdrawal	decimal(10,2)	YES		NULL	STORED GENERATED

account_checking

Field	Type	Null	Key	Default	Extra
AccountNumber	varchar(50)	NO	PRI	NULL	
SavingsBalance	decimal(10,2)	YES		NULL	
MaxSavingWithdrawal	decimal(10,2)	YES		NULL	STORED GENERATED
MinSavingBalance	decimal(10,2)	YES		NULL	STORED GENERATED
SavingsInterestRate	decimal(10,2)	YES		NULL	

account_savings

Application GUI



GUI Specification

- **PART 1:** Allow the user to enter the required information for a Bank Account. This operation displays the new Account information as shown on Slide 9 , enters the information into the **BOL** database, and creates the appropriate entry into the **AccountsPOP.sql** file.
- **PART 2:** Allow the user to enter the Account Number for an existing Bank Account and displays the Bank Account information as shown on Slide 10.
- **PART 3:** Allows the user to show all Bank Accounts in the **BOL** database in the format shown on Slide 11.

Application GUI

Source History Connection:

```
INSERT INTO account_registry VALUES('A101','Gene','Locklear','T101');
INSERT INTO account_checking (accountNumber,checkingBalance) VALUES('A101',5000.0);
INSERT INTO account_savings (accountNumber,savingsBalance,savingsInterestRate) VALUES('A101',10000.0,3.0);
```

Servlet BuildAccount

localhost:8080/WebAp_EX1_Locklear/BuildAccount?acctnumber=A101&taxid=T101&lname=Lockle

Bank Account

Last Name: Locklear
First Name: Gene

Account Number: A101
Tax ID Number: T101

*****CHECKING*****

Checking Balance: $5,000.00 Min Checking Balance: $1,000.00 Max Checking Withdrawal: $2,000.00

*****SAVINGS*****

Savings Balance: $10,000.00 Min SavingBalance: $5,500.00 Max Saving Withdrawal: $2,000.00 Savings Interest Rate: 3.00%

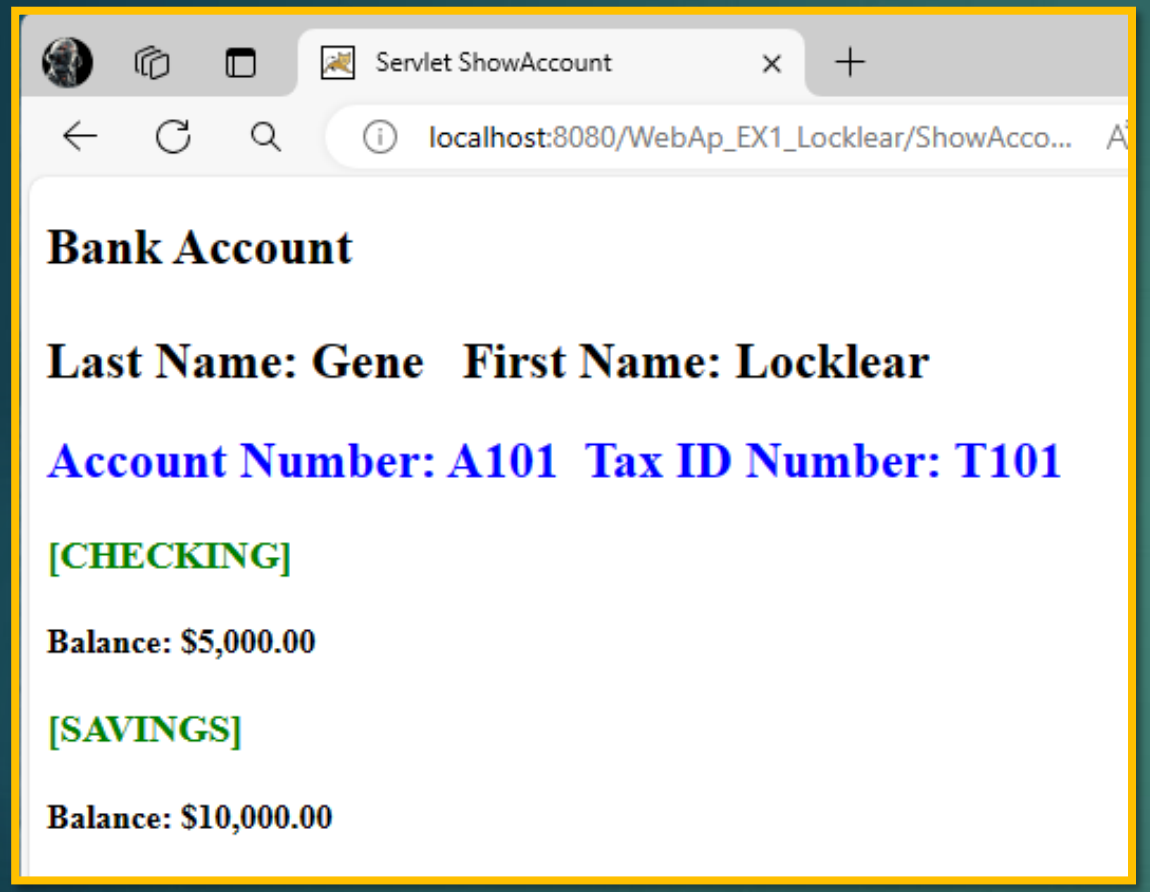
AccountNumber	LastName	FirstName	TaxIDNumber
A101	Gene	Locklear	T101

AccountNumber	CheckingBalance	MinCheckingBalance	MaxCheckingWithdrawal
A101	5000.0	1000.00	2000.00

AccountNumber	SavingsBalance	MaxSavingWithdrawal	MinSavingBalance	SavingsInterestRate
A101	10400.00	2080.00	5720.00	3.00

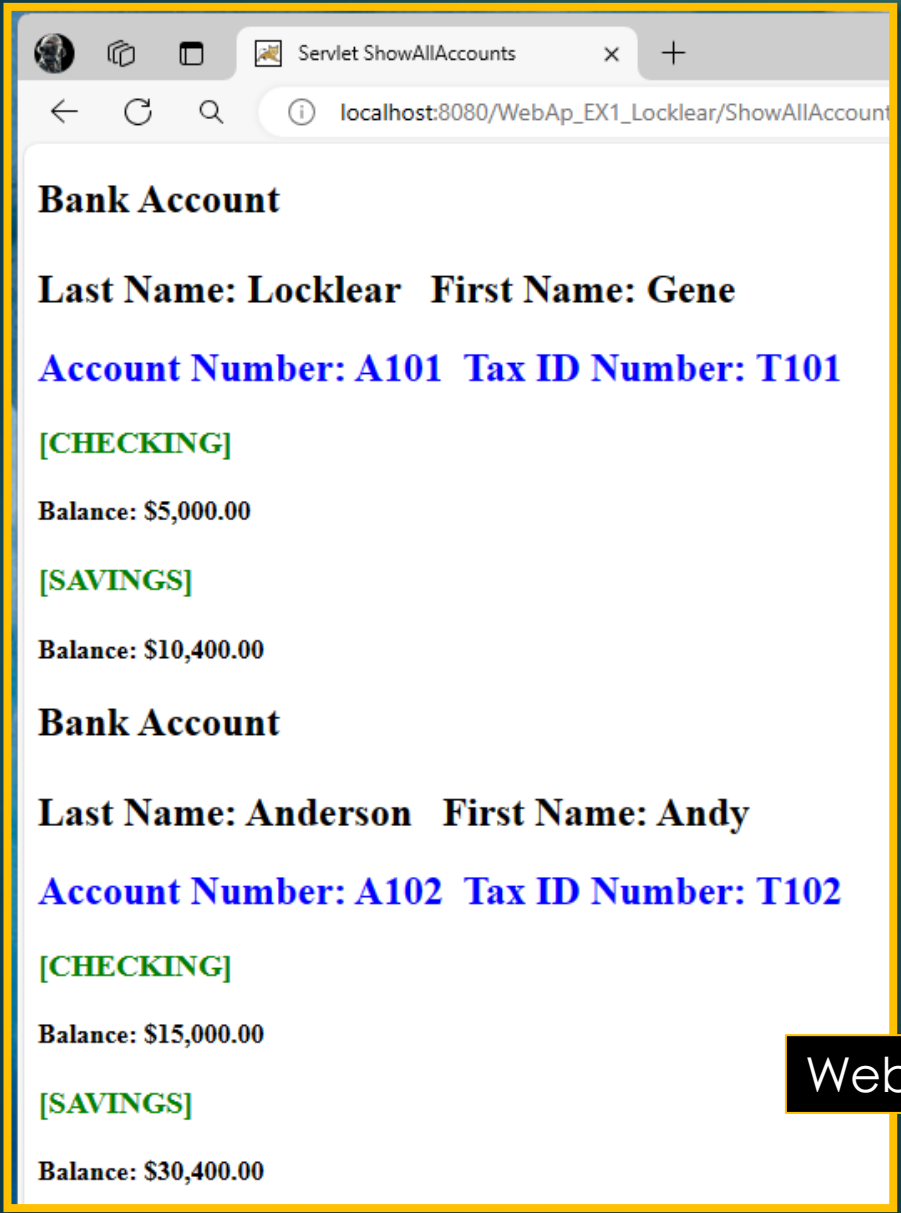
Webpage returned by BuildAccount servlet

Application GUI



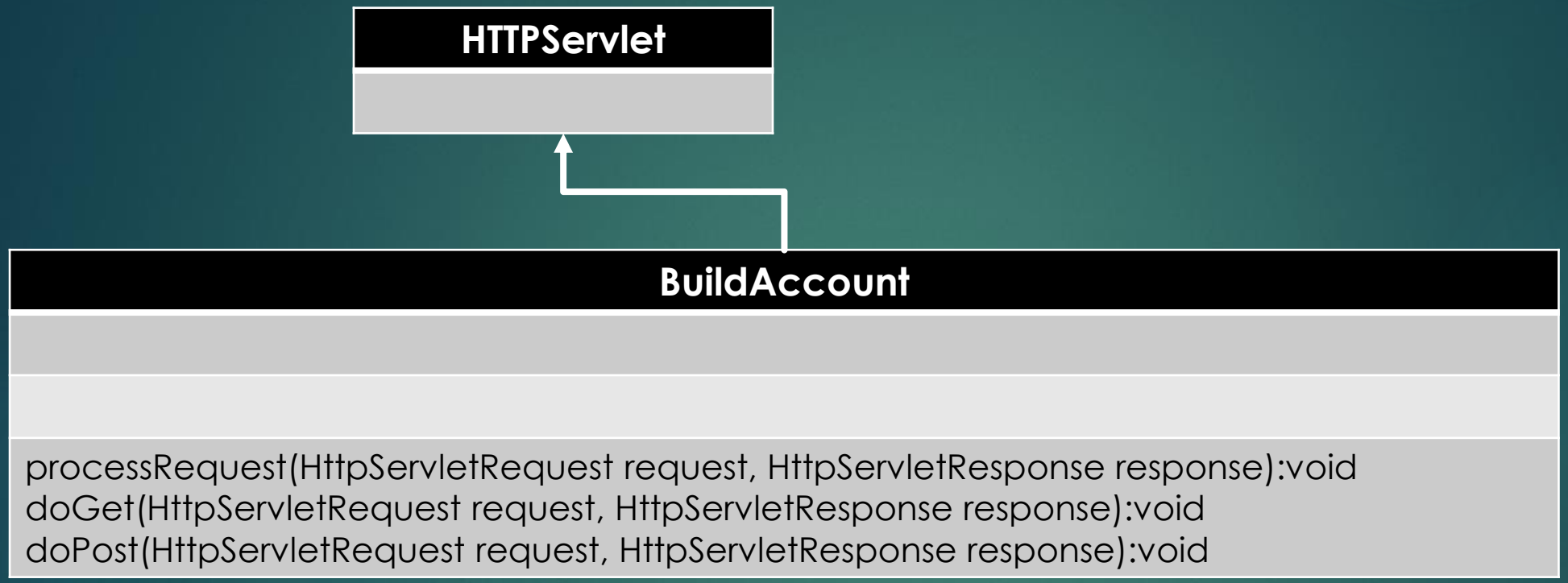
Webpage returned by ShowAccount servlet

Application GUI



Webpage returned by ShowAllAccounts servlet

Application Servlets

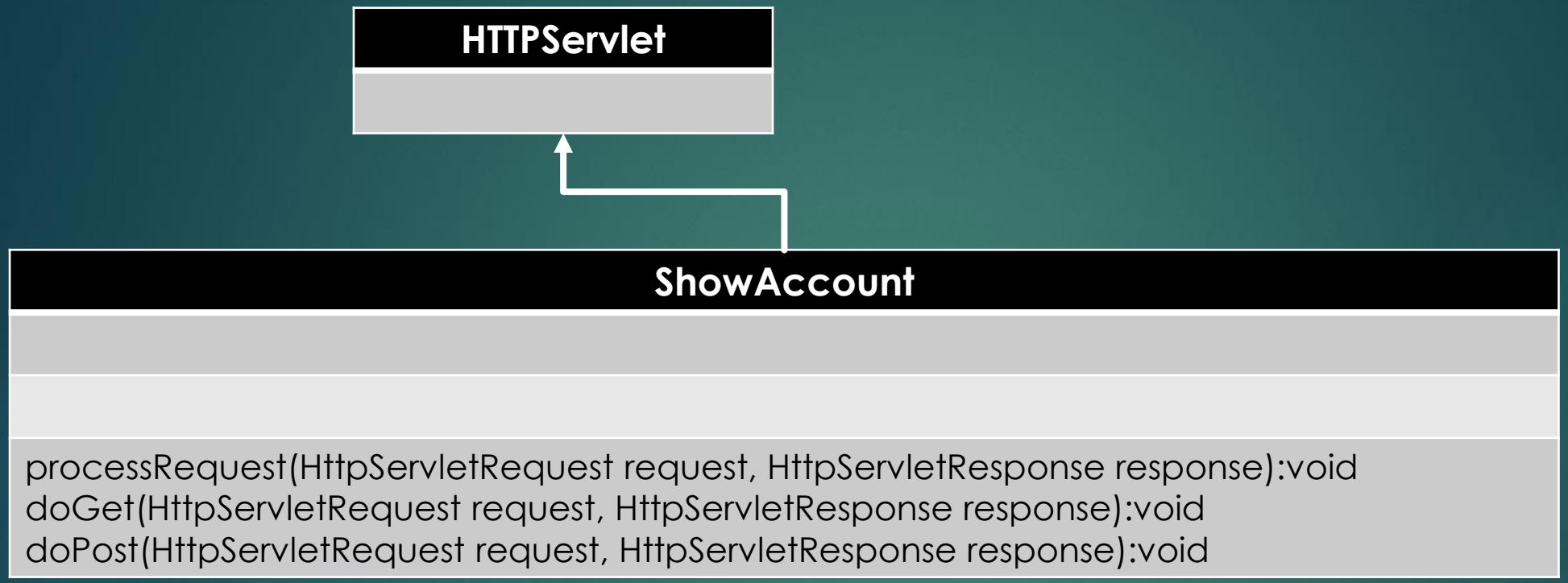


Method Specifications

processRequest(HttpServletRequest request, HttpServletResponse response):void
Accept User Input as parameters to the BankAccount object constructor and creates the specified BankAccount object, stores the object in the BOL database, writes the appropriate INSERT statements to the **AccountPOP.sql** file and displays the BankAccount object as shown on Slide 10.

***doPost** and **doGet** call the **processRequest** method

Application Servlets

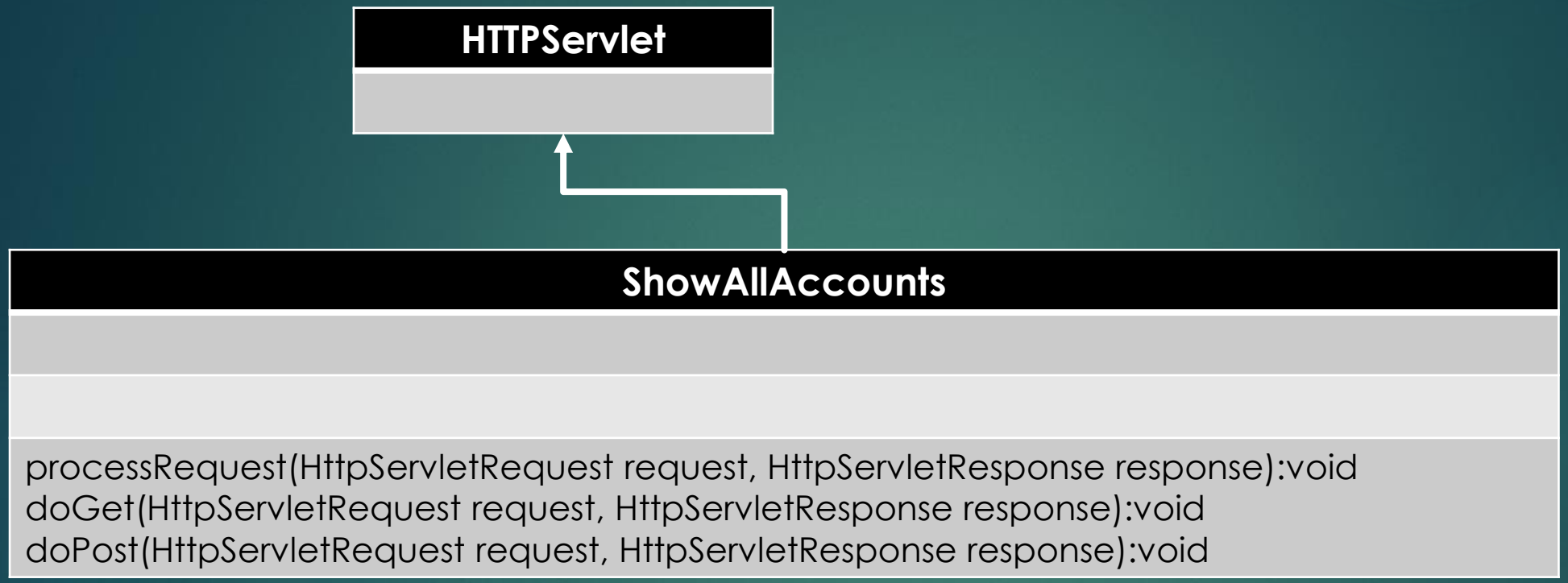


Method Specifications

processRequest(HttpServletRequest request, HttpServletResponse response):void
Accept User Input and displays the appropriate account information from the BOL database in the format shown on Slide 11.

***doPost** and **doGet** call the **processRequest** method

Application Servlets



Method Specifications
processRequest(HttpServletRequest request, HttpServletResponse response):void Displays all accounts in the BOL database in the format shown on Slide 12. ***doPost** and **doGet** call the **processRequest** method