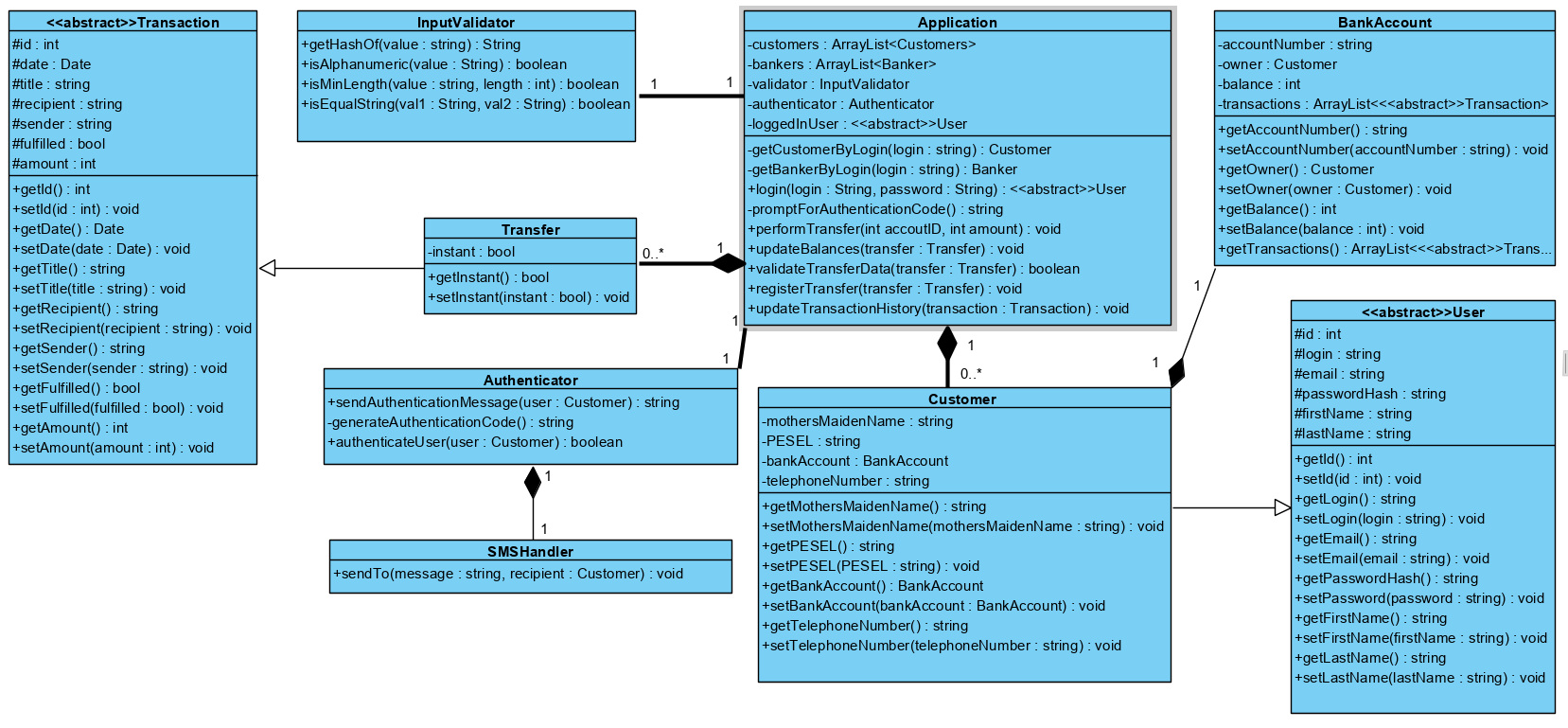
Inżynieria Oprogramowania

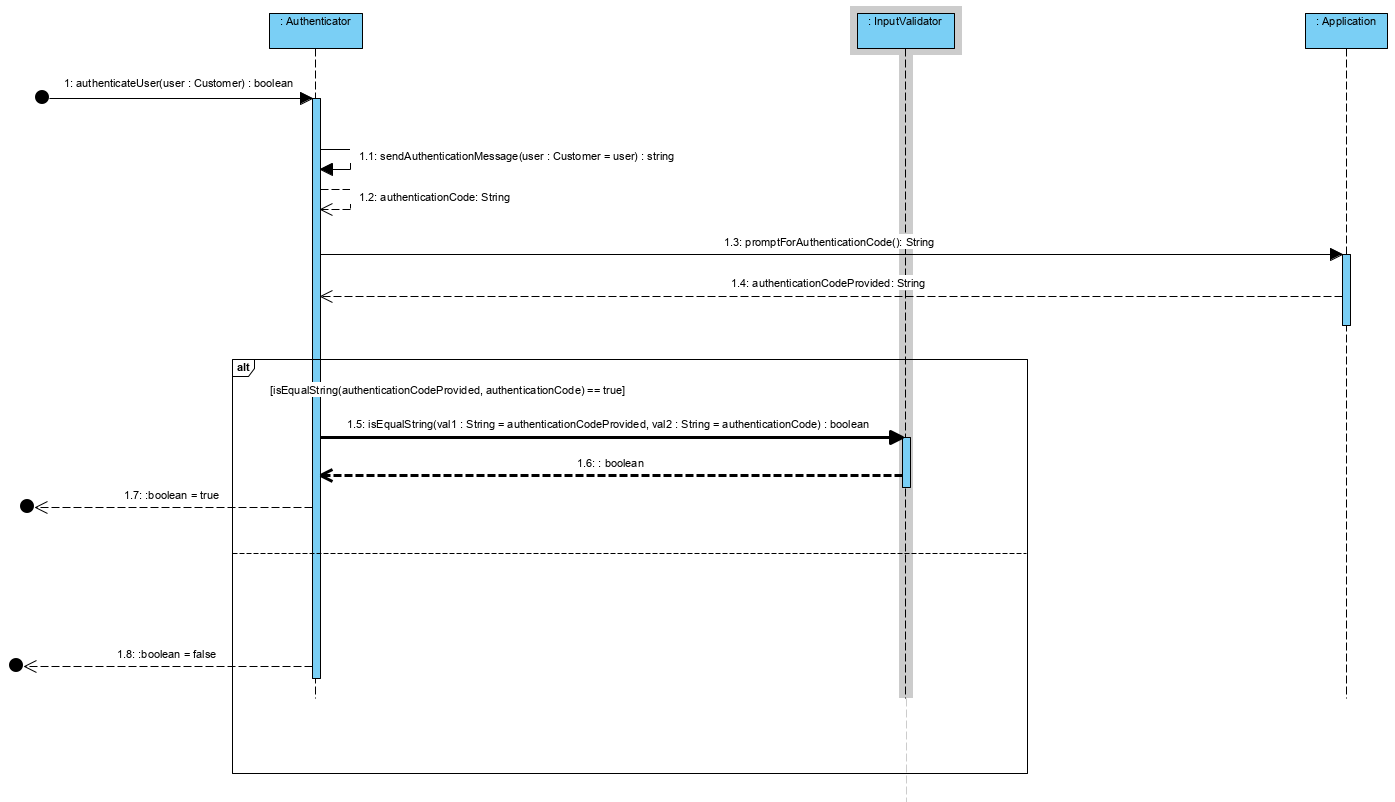
Sprawozdanie

Daniel Król

Artur Boryś

Kod całego projektu znajduje się pod linkiem: https://github.com/SheCanWait/IO





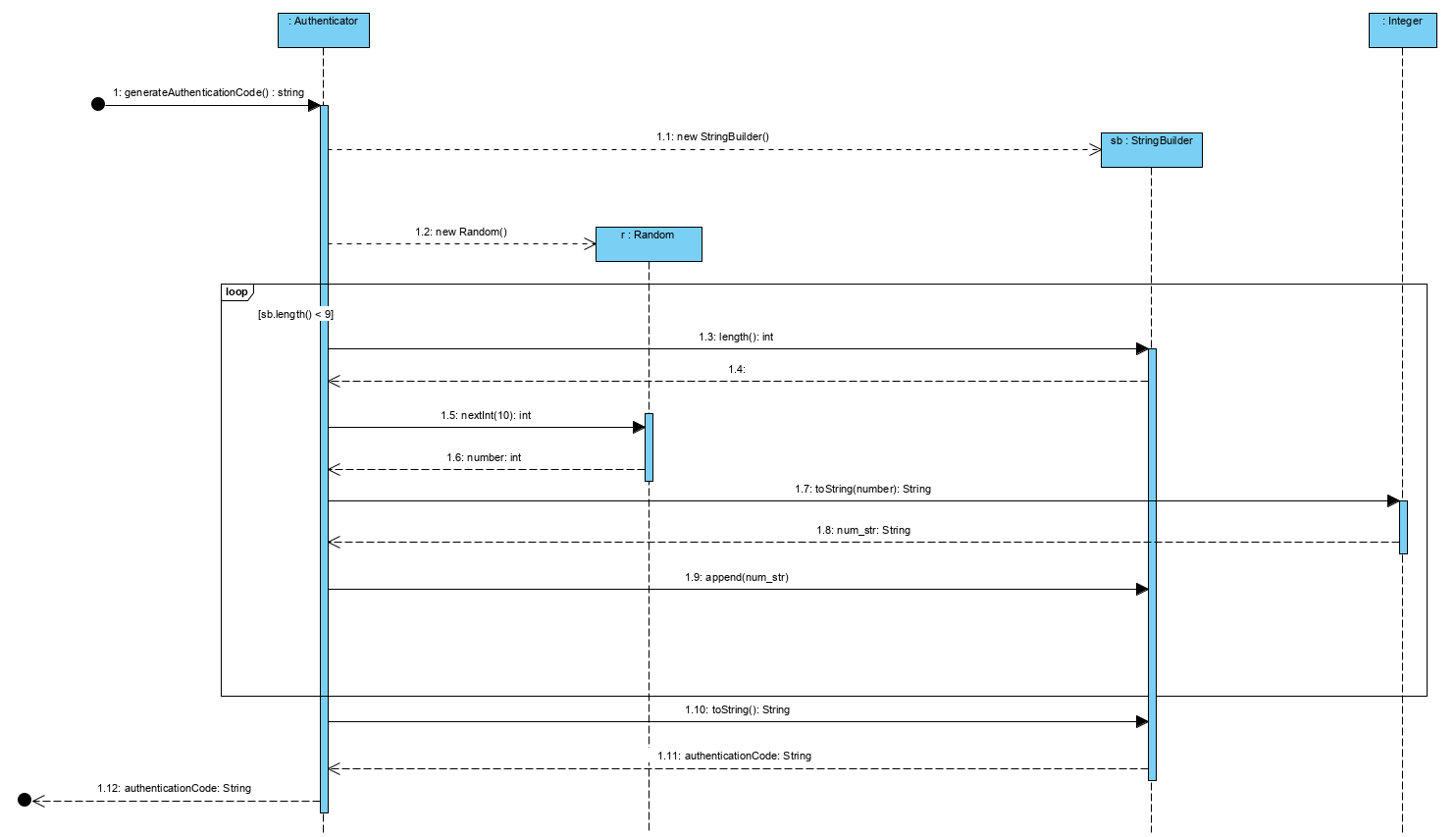
public static boolean authenticateUser(Customer user) {

String authenticationCode = sendAuthenticationMessage(user);

String authenticationCodeProvided = Application.promptForAuthenticationCode();

return InputValidator.isEqualString(authenticationCodeProvided, authenticationCode);

}



private static String generateAuthenticationCode() {

StringBuilder sb = new StringBuilder();

Random r = new Random();

while(sb.length() < 9) {

int number = r.nextInt(10);

String num\_str = Integer.toString(number);

sb.append(num\_str);

}

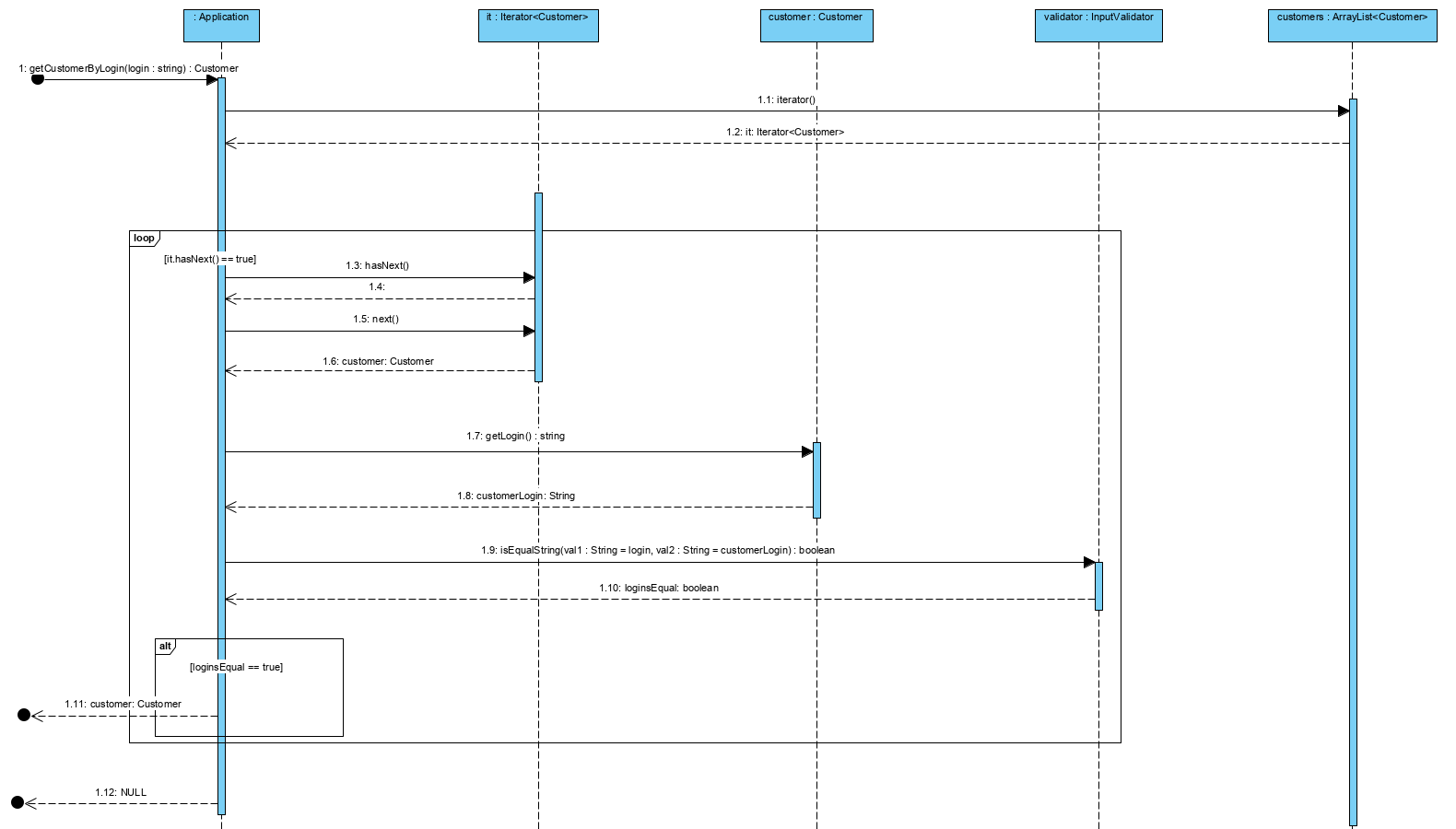
String authenticationCode = sb.toString();

// Potrzebne do zasymulowania wpisywania kodu weryfikacyjnego przez użytkownika

Application.lastAuthenticationCode = authenticationCode;

return authenticationCode;

}



private Customer getCustomerByLogin(String login) {

Iterator<Customer> it = customers.iterator();

while(it.hasNext()) {

Customer customer = it.next();

String customerLogin = customer.getLogin();

boolean loginsEqual = InputValidator.isEqualString(login, customerLogin);

if(loginsEqual) {

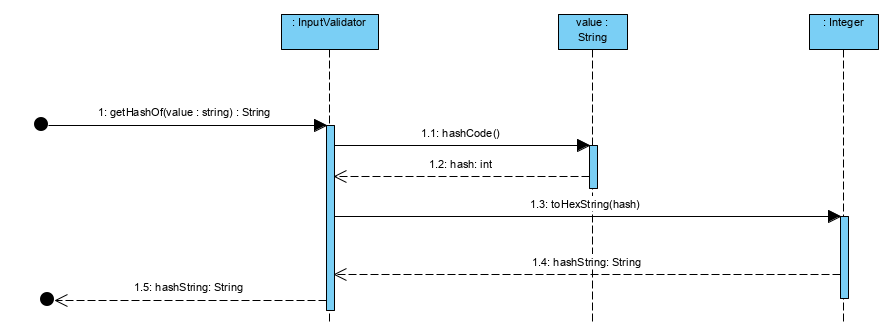
return customer;

}

}

return null;

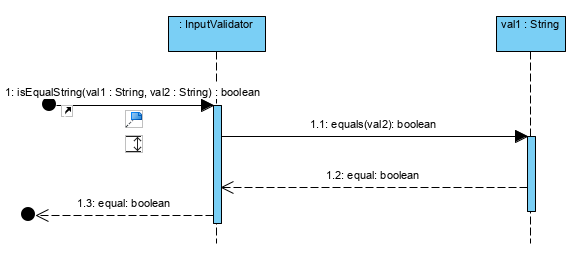
}



static String getHashOf(String value) {

return Integer.toHexString(value.hashCode());

}

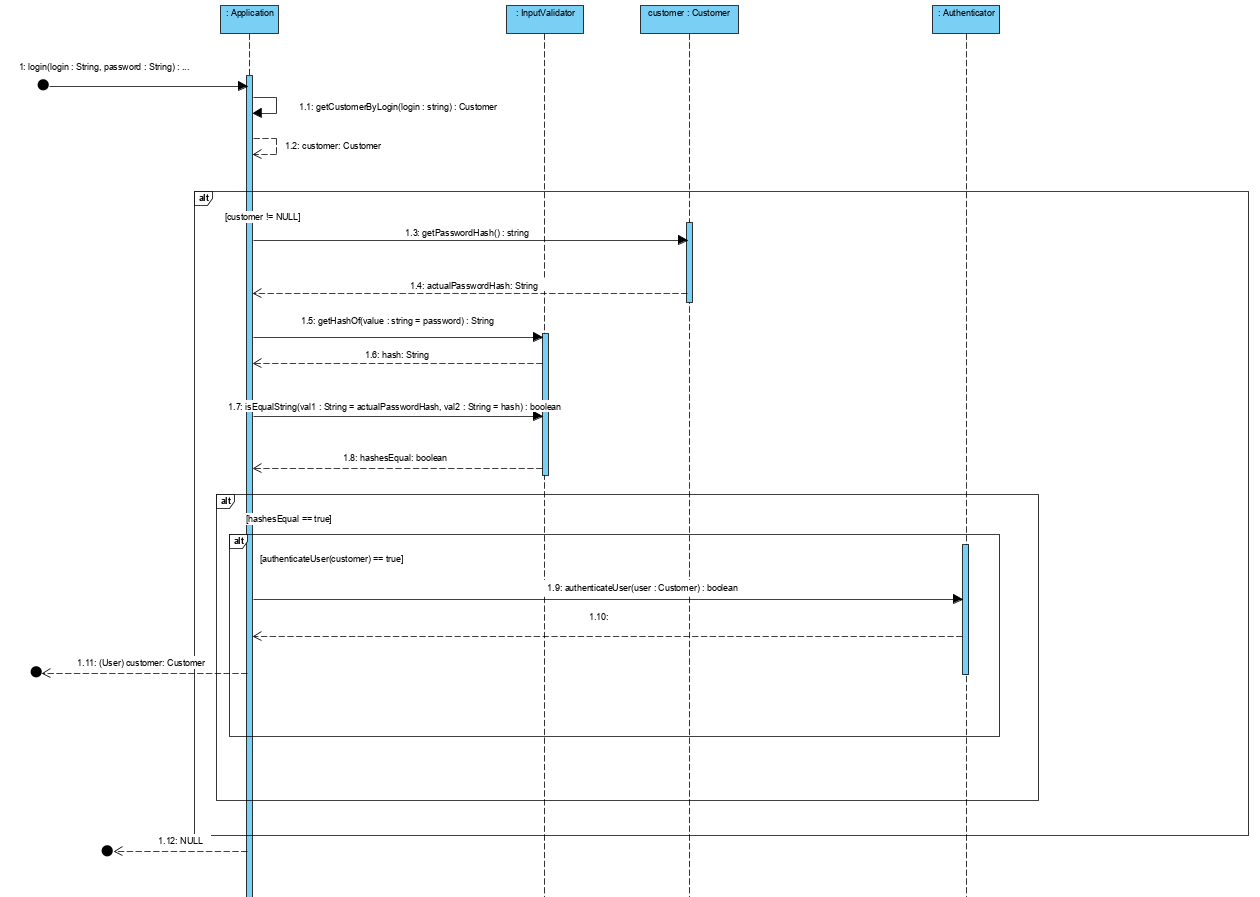


static boolean isEqualString(String val1, String val2) {

boolean equal = val1.equals(val2);

return equal;

}



public User login(String login, String password) {

Customer customer = getCustomerByLogin(login);

if(customer != null) {

String actualPasswordHash = customer.getPasswordHash();

String hash = InputValidator.getHashOf(password);

boolean hashesEqual = InputValidator.isEqualString(actualPasswordHash, hash);

if(hashesEqual) {

if(Authenticator.authenticateUser(customer) == true) {

return customer;

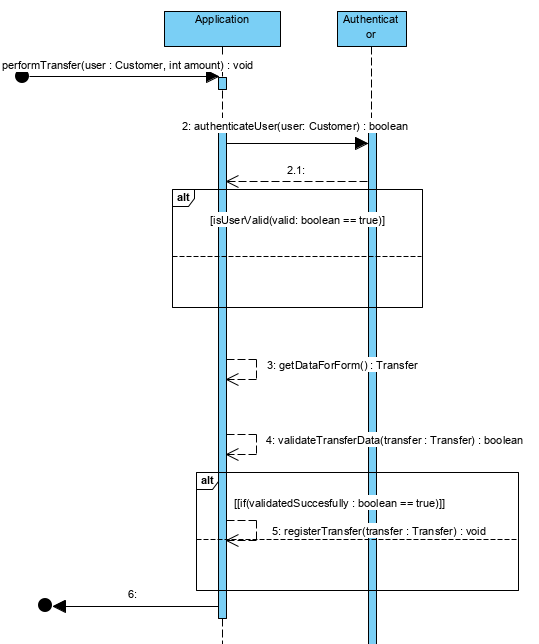
}

}

}

return null;

}



public void performTransfer(Customer user, int amount) {

if(Authenticator.authenticateUser(user) == true) {

Transfer transfer = getDataForForm();

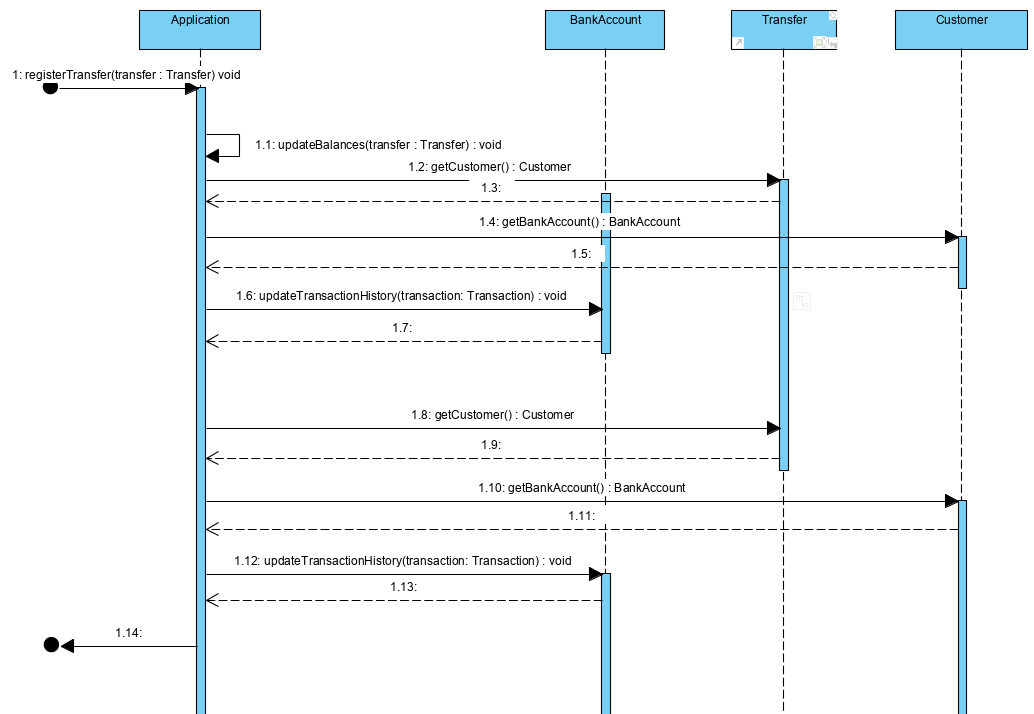
if(validateTransferData(transfer) == true) {

registerTransfer(transfer);

}

}

}



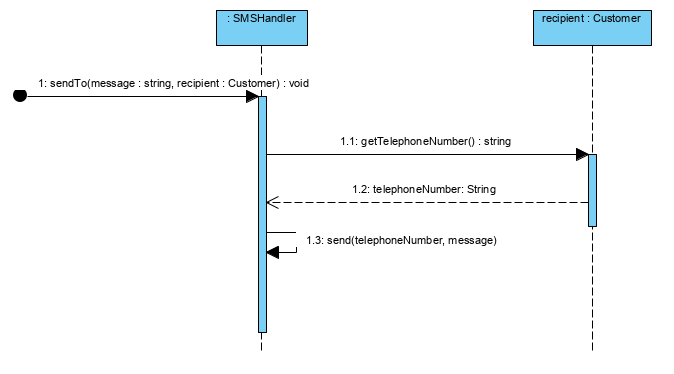
private void registerTransfer(Transfer transfer) {

updateBalances(transfer);

transfer.recipient.getBankAccount().updateTransactionHistory(transfer);

transfer.sender.getBankAccount().updateTransactionHistory(transfer);

}



private static String sendAuthenticationMessage(Customer user) {

String authenticationCode = generateAuthenticationCode();

StringBuilder sb = new StringBuilder();

sb.append("Twoj kod weryfikacyjny: ");

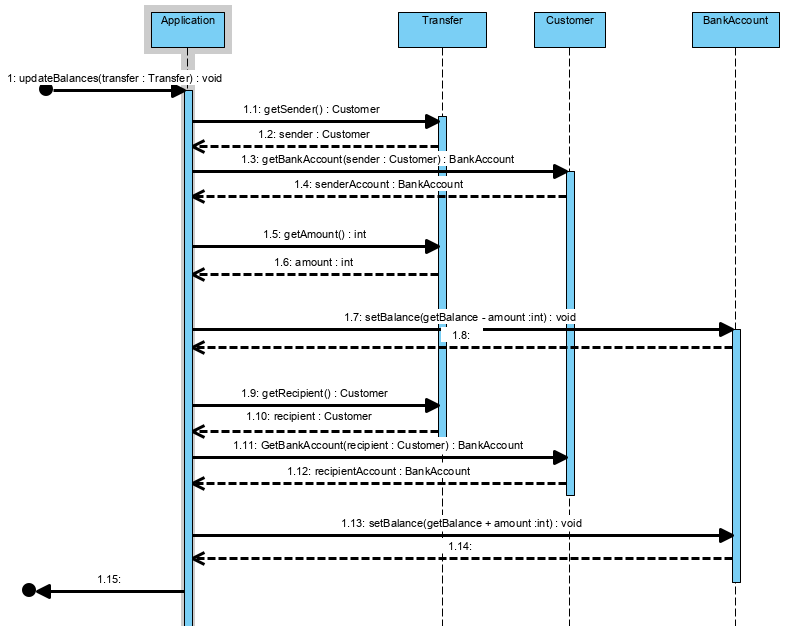
sb.append(authenticationCode);

String message = sb.toString();

SMSHandler.sendMessageTo(message, user);

return authenticationCode;

}



private void updateBalances(Transfer transfer) {

Customer sender = transfer.getSender();

BankAccount senderAccount = sender.getBankAccount();

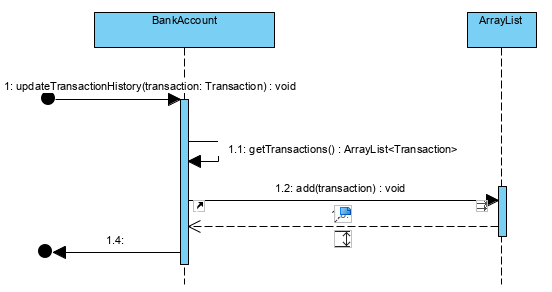
senderAccount.setBalance(senderAccount.getBalance() - transfer.amount);

Customer recipient = transfer.getSender();

BankAccount recipientAccount = recipient.getBankAccount();

recipientAccount.setBalance(recipientAccount.getBalance() + transfer.amount);

}



public void updateTransactionHistory(Transfer transfer) {

getTransactions().add(transfer);

}