Design

[1. The solution and system architecture involving the system with interfacing and deployment scenarios](#_igt2a5z5ubm1)

[2. High level requirements analysis and the assumptions you have made](#_rvtbm4ul38s0)

[Component #1: Banking portal](#_pyodopj66jfv)

[Use Case #1.1: Customer registration](#_92ioj1humt4g)

[Use Case #1.2: Customer login](#_3wbvlxjkrc8m)

[Use Case #1.3: Transferring money](#_mz8wx8hpacbt)

[Component #2: ATM application](#_3tz2m66mdqtk)

[Use Case #2.1: Customer login](#_7h1qn9jdld30)

[Use Case #2.2: Get balance](#_u3kl3sio1f7)

[Use Case #2.3: Withdrawal](#_r7e7js87xvsq)

[Use Case #2.4: PIN change](#_y1giu1w3y8r7)

[Component #3: Web service](#_hbbthf1mknjf)

[Use Case #3.1: Online payment](#_6t6s3ce8yhu)

[3. High level presentation of the data model](#_otmum8qqw16h)

[4. If necessary, architecture diagrams describing the composition and working of the system, explaining the component interaction and process, control and data flows](#_ujsvo84ia5ej)

[5. Technical implementation summary along with the design patterns involved and with reasons that justify your choices.](#_b0sf7zflk2kg)

[6. Use both visual elements (diagrams) and text descriptions to maximize the amount of information conveyed while keeping the document as compact as possible (about 10 pages)](#_gpf3o5204xm)

# 1. The solution and system architecture involving the system with interfacing and deployment scenarios

* **Database.** MSSQL environment. Hosted on provider. It makes sense to deploy it on separate server[s] to increase system performance.
* **Web service.** WCF service. It makes sense to deploy it on separate server to increase system performance. Has the connection to database. It contains both services for ATM application and bill payment methods for external websites.
* **Banking portal.** ASP.NET MVC web site. Hosted on provider. It makes sense to deploy it on separate server to increase system performance. Has the connection to database and to the Yahoo finance web service for loading currency rates.
* **ATM application.** WPF desktop application. Hosted on ATM terminals. Has the connection to the web service above.

# 2. High level requirements analysis and the assumptions you have made

## **Component #1:** Banking portal

### **Use Case #1.1:** Customer registration

**Primary Actor:** Web site visitor (Unregistered user)

**Basic flow:**

1. The user visits the web site.
2. The user selects the option *“Register”*.
3. The system opens the “*Register page”*.
4. The user fills the mandatory fields.
5. The user submits the register information.
6. The system authenticates and authorizes the user.
7. The system presents the updated view as far as user is authenticated.

**Extensions:**  
а. Cancel the registration

1. The user selects “*Cancel”*.
2. The system goes to step 1.

b. No required data

1. The user doesn’t entered any of mandatory fields.
2. The user submits the register information.
3. The system shows validation error: “Please fill the required field <<Field name>>”.
4. The system doesn’t submit the data.

c. User is already existed

1. The user entered all of mandatory fields.
2. The user submits the register information.
3. The system detects that user with entered login is already registered.
4. The system shows validation error: “The user with this login is already registered. Try to invent unique login.”.
5. The system doesn’t submit the data.

### **Use Case #1.2:** Customer login

**Primary Actor:** Web site visitor (Unregistered user)

**Basic flow:**

1. The user visits the web site.
2. The user selects the option *“Login”*.
3. The system opens the “*Login page”*.
4. The user fills the fields: “*Login” and “Password”*.
5. The user submits the information.
6. The system authenticates and authorizes the user.
7. The system presents the updated view as far as user is authenticated.

**Extensions:**  
а. Cancel the login

1. The user selects “*Cancel”*.
2. The system goes to step 1.

b. No required data

1. The user doesn’t entered any of required fields: “*Login” or “Password*.”
2. The user submits the register information.
3. The system shows validation error: “Please fill the required field <<Field name>>”.
4. The system doesn’t authenticates the user.

c. User is not exists

1. The user entered all of required fields: “*Login” and “Password”*.
2. The user submits the register information.
3. The system detects that user with entered login and password is not exists.
4. The system shows validation error: “The user with this login or password is not registered.”.
5. The system doesn’t authenticates the user.

d. Login via Facebook

1. The user selects option “Login via Facebook”.
2. The user authenticates via its existing FB account .
3. The system created new user with FB data and presents the updated view as far as user is authenticated.

### **Use Case #1.3:** Transferring money

**Primary Actor:** Member (Registered user)

**Preconditions:**

* The web site visitor authenticates as bank’s client (UC #1.2).

**Basic flow:**

1. The user visits the web site.
2. The user selects the option *“Transfer money”*.
3. The system opens the “*Transfer money”* screen.
4. The user enters all the options for money transferring.
5. The system transfers money.
6. The system sends an email to user with transfer information.

**Extensions:**  
а. Cancel the transferring

1. The user selects “*Cancel”*.
2. The system goes to step 1.

b. No required data

1. The user doesn’t entered any of required fields.
2. The user tries to make transferring.
3. The system shows validation error: “Please fill the required field <<Field name>>”.
4. The system doesn’t transfer money.

c. Not enough money

1. The user entered all of required fields.
2. The user tries to make transferring.
3. The system shows validation error: “Not enough money to transfer”.
4. The system doesn’t transfer money.

## **Component #2:** ATM application

### **Use Case #2.1:** Customer login

### **Use Case #2.2:** Get balance

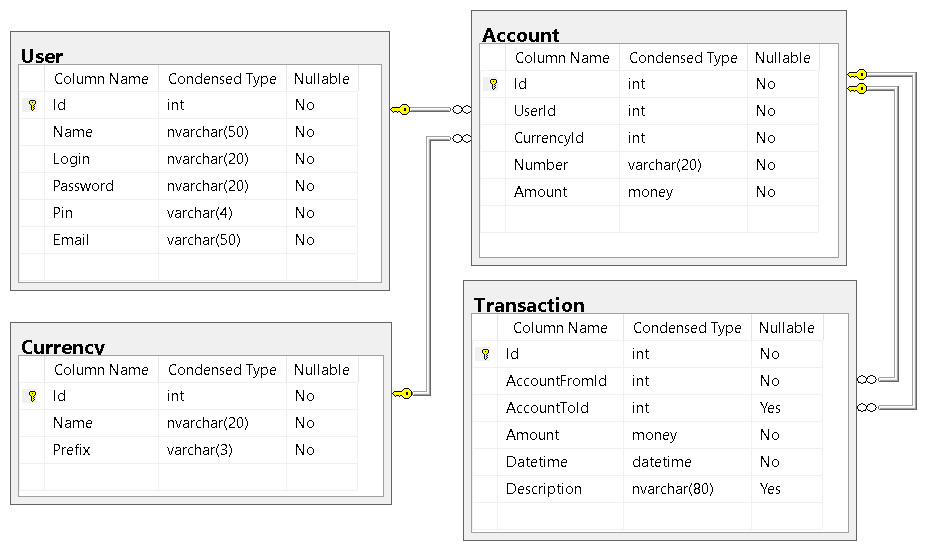
### **Use Case #2.3:** Withdrawal

### **Use Case #2.4:** PIN change

## **Component #3:** Web service

### **Use Case #3.1:** Online payment

# 3. High level presentation of the data model



# 4. If necessary, architecture diagrams describing the composition and working of the system, explaining the component interaction and process, control and data flows



# 5. Technical implementation summary along with the design patterns involved and with reasons that justify your choices.

N/A

# 6. Use both visual elements (diagrams) and text descriptions to maximize the amount of information conveyed while keeping the document as compact as possible (about 10 pages)

Done. See above.