IN2013 Object-Oriented Analysis and Design
2025/26 Coursework Assignment 1 (Analysis)

Z-Flexi Credit Card Mobile App

Assignment

Dr Peter T. Popov

Version 1.1

September 11th, 2025

Document History

1. A revision of the initially released text (v1.0).
2. Changes made:
    a. Fixed inconsistency in stating the submission deadline on p. 7. The text "Sunday, the 14th of November" has been changed to "Friday, the 14th of November" to reflect that the 14th of November is a Friday.

Introduction

There will be two pieces of coursework for IN2013, each worth 45% of the total mark for the module.

This coursework assignment tests your knowledge on ***object-oriented analysis***. You are asked to analyse a scenario and then develop a set of user requirements and analysis models for the software system described in the scenario.

The **deadline** for this coursework is **17:00 on 14th of November 2025**, and all work must be submitted electronically in accordance with the guidelines below.

Scenario: Z-Flexi Credit Card Mobile App[1]

Zonder, a financial organisation based in the UK, came up with an innovative idea of a "credit card with flexible benefits", Z-Flexi Card. The benefits are similar to other popular schemes, such as the "Avios" offered by Airlines in partnership with credit card providers, whereby Avios are earned when a credit card is used to purchase goods and services. Avios can later be used to purchase flight tickets.

A card holder of Z-Flexi card would earn "reward points" which can later be used to pay for flights, and visits of cinemas, concerts, restaurants, etc, in the city where the card holder lives. Additional, benefits from the Z-Flexi card are: i) access to passenger lounges at airports, and ii) no transaction/currency conversion fees for using the card oversees. Z-Flexi is issued as a digital/virtual card, suitable for online purchases only, but the holder can request a physical credit card, too.

Zonder are looking for a credible software company to develop a prototype mobile application, Z-Client, and a complementary cloud-based web-service, Z-Server, which Z-Client will rely on.

The Z-Client mobile app

The core functionality of the mobile app is summarised below:

- ***Registration.*** A Customer willing to register with the service applies for a Z-Flexi credit card by completing an application form using a web-browser. The application form shall include essential personal details such as full name, home address, email address and mobile number, which Zonder will use to check the Customer credit score. Processing the application may take up to 2 working days. The outcome will be communicated to the applicant by email. The outcome can be:
  - Application is accepted. A new customer account is setup on the Z-Server with a unique CustomerID, the customer details as provided in the application form (name, address, email and mobile number), and a reward points register. An email to the applicant will be sent (by the Z-Server) confirming successful application. The email will contain the URL of the Z-Server, a password for the initial login, and detailed instructions about how to activate the Customer Account. The customer will need to install the Z-Client app on their mobile and to complete an initial login using the email they have provided in the application form together with the password included in the email they have received confirming the successful card application[2]. Upon successful initial Login the following will take place "in the background" via interaction between Z-Client and Z-Server:
    - the customer account is activated. The customer is offered the option to change their password as part of the initial workflow.
    - A Digital Credit Card is generated[3] with a unique card number, expiry date, security code (CVV/CSC), credit limit and a balance register, and is linked to the customer account. The card can be used immediately with popular mobile payment methods such as Apple Pay, Google Wallet, and others.
    - Z-Client and Z-Server exchange encryption keys needed for secure communication.
  - Application is rejected. The customer will receive an email with the reasons for rejection, and the application is closed. All personal details included in the application form are destroyed.
- ***Login/Logout.*** The functionality of the Z-Client will be available to use only after a successful Login. For logins the Customer can use either their email or Customer ID, and the

[1] The scenario is inspired by the "Yonder Full Credit Card", but required functionality is a fraction of the functionality offered by the Yonder's mobile app.

[2] For increased security, the password may be made valid for a limited period, e.g., a couple of hours, but this additional complexity is not required for the prototype.

[3] The card will be generated on the Z-Server. The Customer will have access to it via the Z-Client app.

password stored in their account. To avoid misuse of Z-client (e.g. if the mobile phone is lost or stolen) an automatic Logout is triggered if the Z-Client has not been used for more than 15 min.

- ***Account access*** (after a successful login):
  - The User shall be able to view their account details, change password, address, email address, view the list and details of the "benefits" currently available. Some benefits (cinemas, restaurants, etc.) will change over time. The info on these will indicate the period the benefits are available. Other benefits, such as access to passenger lounges at airports are not time restricted, but may have certain restrictions applied to them, e.g. no more than certain number of visits of passenger lounges within a calendar year. For such benefits the customer shall be able to check the number of visits they have left on their account using the Z-Flexi app.
  - Should the User decide to visit any of the available venues, they shall make a reservation and pay for it.
  - The user shall be able to view the credit card details, credit limit, and current balance.
  - The Customer shall be able to request a physical credit card. It will have the same details as the digital card and all transactions paid for with the physical card will be processed in the same way as the transactions paid with the digital credit card.
- ***Make payments.*** Unless due payments are received on time, the card will be blocked until all outstanding payment(s) are received in full.
  - Annual/monthly fee. This can be done "manually" via the Z-Flexi app[4].
  - Monthly payment for the use of Z-Flexi card. The minimum amount due is computed (by the Z-server app) and shown via the Z-Client. The User can make a larger payment, if they so choose.
- ***Confirmation of payment method.*** Whenever a payment is attempted with the Z-Flexi card (virtual or physical), the Z-Server will trigger a dialog with the Customer's Z-Client app to establish the payment method (card or reward points) the customer selects for their transaction. Z-Server will confirm by email the chosen payment method and the amount charged.

All communications between Z-Client and Z-Server must be encrypted to the strongest encryption standards. The data held on the mobile app and on the cloud must be encrypted, too, with efficient mechanisms in place to guarantee data recovery, e.g. when the customer replaces their mobile handset, or in case of encryption keys being lost or compromised.

We assume that both Z-Server and Z-Client are always available.

The Z-Server cloud application (given for information, not used in the assignment)

The core functionality of the server application deployed in the cloud is summarised below:

- ***Accept applications*** and store on cloud storage the application forms received from prospective new card holders.
- ***Create a customer account***. Upon satisfactory credit check, a member of Zonder staff will create a new Customer account with a unique CustomerID, and details as provided by the customer in their card application form. A reward points register will be added to the account, too. An email will be generated and sent to the customer with the details as listed above. The customer account will remain inactive until the customer logs in for the first time via Z-Client.
- ***Allow legitimate customers access*** to their accounts. The access will offer the following options:

[4] Direct debit is also an option, but this should be done via customer bank, and is not part of Z-Client app.

- Retrieve the Account details. This feature will be accessed via Z-Clients.
- Change Account details: password, mobile number, email address, or address.
- Retrieve the Z-Flexi card details (both virtual and physical).
- Accept a reservation request for a visit of a listed venue (entertainment, restaurant, etc.)
- Make payments of i) monthly/annual fee, and ii) of minimum monthly amount.

- ***Payment clearance.*** Upon receiving a payment clearance request (e.g. from a retailer's POS[5]), the Z-Server app will interact with Z-Client app of the respective customer to establish the method of payment (card or reward points) the customer selects for the transaction. Choosing a card payment will increase the card balance by the amount of the new transaction and increasing the reward points (£1 spent leads to an extra reward point after rounding down the transaction amount), accordingly. Selecting payment with reward points will lead to reduction of the reward points accordingly (in this case the rounding up of transaction amount will be applied). Should the available reward points be insufficient for full payment, the payment will be split into two parts – one using the available points and another one adding the difference between the transaction amount and the reward points to the credit card balance.

[5] Point-of-Sale.

Assignment

You are expected to develop a set of requirements and UML models of Z-Client software and answer the following questions.

Question 1: User requirements

Using the Volere Template, introduced in Lecture 1, specify 1 functional and 1 non-functional requirement for the Z-Client software using the provided scenario.

(5 marks)

Question 2: Use case Diagram

Draw a use case diagram for *Z-Client software*, which covers the functionality described in the scenario[6]. The diagram should include:

- The Actors (primary and secondary) of *Z-Client software*. Consider the users of *Z-Client software* and the external systems *Z-Client software* relies upon.
- The use cases, which capture the main services provided by *Z-Client software* to the respective actors.
- The generalization relationships between Actors.
- The relationships between the use cases (<<include>>, <<extend>> and generalization).

(20 marks)

Question 3: Use case Specifications

An important part of core functionality of Z-Client software is processing payments made with Z-Flexi card. As spelled out in the scenario this will include Z-Server making a request to Z-Client, which will then require a decision on the payment method to use by the holder of the Z-Flexi card. Let's assume that a use case "*ProcessPaymentRequest*" is included in your use case diagram for this purpose. Provide a specification of this use case for both the main and the important alternative flows.

The specifications should cover all the options listed in the scenario and should:

- Spell out the interaction between the actors and the system (the Z-client software):
    - Invocation of Z-Client by Z-Server
    - Selection of the payment method (card or reward points) by the User of Z-Client (i.e. the customer), and
    - Z-Client relaying the choice of the User to the Z-Server. The choice of payment method will trigger amendment of the reward points register and of the credit balance, accordingly, on both the Z-Server and the copy of the data held on Z-Client.

  Recall that according to the scenario we can assume that Z-Client and Z-Server are always available and can communicate. There is, however, a possibility that the transaction amount exceeds the available credit/reward points.
- Capture the important circumstances (e.g., branches in the flow, possible loops, etc.) that might occur while transaction payment clearance is processed.

The use case *ProcessPaymentRequest* might have relationships (<<include>>/<<extend>> or generalization) with other use cases. Providing specifications for these use cases, however, is NOT required.

> **Hint:** Make sure, however, that the specification of *ProcessPaymentRequest* is consistent with the use case diagram: all relationships of *ProcessPaymentRequest* shown in the use case diagram must be matched by "include(…)" steps and "extension point: …" lines in the specification of *ProcessPaymentRequest.*

If you make additional assumptions, please state these in your answer to the question.

(10 marks)

---

[6] Detailing in the use case diagram the functionality of Z-Server is not required.

Question 4: Analysis class diagram

This question is about analysis class diagram and consists of two parts.

a) Develop an analysis class diagram for Z-Client software. Concentrate on the ***problem domain classes***, show their attributes and important operations and the associations between the classes.

- There is no need to include type information, get and set methods, or constructors.
- Consider a minimal set of boundary and control classes that might be needed for the realization of the use case "ProcessPaymentRequest" (as required in Q5).
- Relationships:
  - Use associations in your class models and label them with association or role names, as appropriate, show the association directions, and multiplicities, but don't worry about navigability.
  - Use generalization (inheritance) between classes, where appropriate.
  - Don't bother with dependency relationships.

(30 Marks)

b) ***Substantiate*** your answer by demonstrating competence with the taught techniques for identifying analysis classes and their relationships as follows:

b.1. Apply the noun/verb analysis to the following fragment from the provided scenario:

> - The User shall be able to view their account details, change password, address, email address, view the list and details of the "benefits" currently available. Some benefits (cinemas, restaurants, etc.) will change over time. The info on these will indicate the period the benefits are available. Other benefits, such as access to passenger lounges at airports are not time restricted, but may have certain restrictions applied to them, e.g. no more than certain number of visits of passenger lounges within a calendar year. For such benefits the customer shall be able to check the number of visits they have left on their account using the Z-Flexi app.
> - Should the User decide to visit any of the available venues, they shall make a reservation and pay for it.
> - The user shall be able to view the credit card details, credit limit, and current balance.
> - The Customer shall be able to request a physical credit card. It will have the same details as the digital card and all transactions paid for with the physical card will be processed in the same way as the transactions paid with the digital credit card."

b.2. Demonstrate the use of CRC cards technique on the following *3 classes from the problem domain*: Customer, CreditCard, Transaction.

b.3. Apply Robustness analysis to the analysis class model by adding to the class diagram control and boundary classes sufficient for the realisation of use case *ProcessPaymentRequest* and checking if the associations between the classes in the class diagram satisfy the robustness analysis rules.

(10 marks)

Question 5: Use case realisation (sequence diagram)

Draw a sequence diagram that realizes the use case "ProcessPaymentRequest". The diagram should cover all possible branches (extensions, alternative flows, and if-else's) and possible loops as defined in your answer to Q3. Make sure that your sequence diagram is consistent with the class diagram developed in Q4 and, of course, with the use case specification developed in Q3.

> ***Hint:*** Note that as a result of developing the sequence diagram your analysis class diagram *may change* – you may need to add new operations to some of the classes or even add new classes to the class diagram. The use case specifications may change, too: you may discover that the flows (the main and/or the alternatives) may need to be modified.
> The UML models included in the submission ***must be consistent***. The simplest way to achieve consistency

is to develop *all UML diagrams* and only then take a snapshot (i.e. export as images) the class, use case and of sequence diagrams and include these in your CW submission.

(25 marks)

Submission guidelines

1) Submissions can **only** be made **electronically** via **Moodle**, using the Coursework submission area for the IN2013 module.

2) The **deadline** for submission is **17.00 on Friday, the 14th of November 2025**.

3) Moodle will adhere to the cut-off date/time and automatically prevent you from attempting to submit your work after the deadline.

4) I suggest that you do not even attempt to work right up to the deadline and instead recommend that you get your submission in well before the cut-off time. The last thing you need is the stress and worry of watching the clock tick and then encountering a problem that delays you. It can and does happen!

Late submission policy

In accordance with the usual policy on coursework submission, **late submission will receive 0%** unless there are **extenuating circumstances with supporting evidence**, which must be notified to the Programme's Office in **advance of the deadline**.

Feedback

You will be provided with individual feedback (e.g. as an audio file) with details comments on your submission and a provisional mark via the grade book in Moodle within 3 weeks. Feedback on the common mistakes found in the submissions will be explained in class on the 9th of December 2025, the last lecture in the module.

Format

All UML diagrams must be created using a UML tool and exported as images for subsequent inclusion in the submission document. Diagrams drawn without a tool will be penalized by ***deducting 20%*** from the awarded mark.

Your coursework must be submitted as a **single** PDF file. Export your diagrams from the UML tool you have used as images and then assemble all your answers – text and diagrams – in a single word processor file (e.g. Word) and convert it to a PDF file. Make sure your diagrams are eligible.

You may also submit your entire Visual Paradigm project (if you have used Visual Paradigm for the diagrams, of course) to help the marker with the assessment in case some of the diagrams are not legible in the submitted pdf file.

UML Diagrams

You can install a copy of Visual Paradigm (VP) on your personal machine (Windows, Mac, or Linux) by following the instructions provided on Moodle. If you have used VP before you may need to update the license file for your installation by obtaining the new activation key from Moodle.

Note that you may come across variations in UML syntax on websites and within certain textbooks. This is of no consequence for the purposes of this coursework.