

Table of Contents

1	Requirements.....	3
1.1	Problem statement	3
1.1.1	Addressing the problem.....	3
1.1.2	Solution.....	3
1.2	Glossary.....	4
1.3	Supplementary specifications.....	4
1.4	Use-case models	6
1.4.1	Approve Advertisements	6
1.4.2	Create Account	7
1.4.3	Delete Advertisement	8
1.4.4	Delete User	9
1.4.5	Edit Advertisement	10
1.4.6	Log In.....	11
1.4.7	Process User Reports	12
1.4.8	Publish Advertisement.....	13
1.4.9	Report Advertisement.....	13
1.4.10	Search For Advertisements.....	14
1.4.11	Update Account	15
1.4.12	View Advertisement	16
2	Use-case analysis	17
2.1	Architectural analysis	17
2.1.1	High-level organisation of the model	17
2.1.2	Key abstractions.....	17
2.2	Use-case realisations	18
2.2.1	Use-case realisations: sequence diagrams	18
2.2.2	Use-case realisations: views of participating classes	27
2.2.3	Describe analysis mechanism.....	33
3	Use-case design.....	34
3.1	Architectural refinement.....	34
3.1.1	Identify design elements	34
3.2	Identify design mechanisms	37

3.3	Describe the run-time architecture	38
3.4	Describe distribution	38
3.5	Use-case design	39
3.5.1	Design sequence diagrams.....	39
3.5.2	Design views of participating classes	46
3.6	Subsystem design	46
3.7	Class design	47
3.8	Database design	53

1 Requirements

1.1 Problem statement

1.1.1 Addressing the problem

As technology develops, online shopping/trading is becoming increasingly popular. It is a much more convenient and easier way to shop as buyers don't have to leave their house, they can buy whatever they want without ever standing in a queue or crowd. This evokes the need for a system to help the communication between sellers and buyers. Sellers need a medium to market their products to a wide range of people, thus to increase the number of potential customers. Buyers need a place to look for what they need to buy and to compare products of multiple suppliers. This has come to a demand for a system to solve this problem.

1.1.2 Solution

Chotot.com is built as an online trading system serving as a medium to directly connect buyers and sellers.

Description

The system will be developed as a web application. The end users will interact with the system over the Internet via a wide range of devices (laptop, PC, tablet, smartphones).

Users and Visitors can search for advertisements of products that they are interested in. They can also view information of products whose advertisements have been published by shop owners and report spam/misleading advertisements.

People can register for an account and log into the system to publish an advertisement about a product, edit their published advertisements and update their profile.

Administrator can log into the system to perform their monitoring tasks. They can approve advertisements submitted by Users, view reports made by Users, delete spam advertisements and delete users with harmful activities.

End users

The system's end users are as follow:

- *Users and Visitors*
Users and Visitors can search for and view advertisements. Also, users can report possibly spam or harmful advertisements to Administrators.
After registering an account, a user can publish advertisements about their products and manage those advertisements (edit, delete).
- *Administrators*
Administrators are responsible for monitoring all user accounts and advertisements. All advertisements published by Users must be approved by Administrator to ensure the system's reliability. Administrator can view reports made by Users, delete spam advertisements and delete users with abusive behaviours.

1.2 Glossary

Introduction

This document is used to define terminology specific to the problem domain, explaining terms, which may be unfamiliar to the reader of the use-case descriptions or other project documents. Often, this document can be used as an informal data dictionary, capturing data definitions so that use-case descriptions and other project documents can focus on what the system must do with the information.

Definitions

The glossary contains the working definitions for the key concepts in the Chotot.com website.

Account

A record about a user/administrator containing information about his/her name, e-mail address, password, phone number and optional self-introduction. Each account has a unique user ID and a password, which are used to identify the user/administrator and grant them access to secure parts of the system.

Administrator

A person whose job is to ensure that the site is free of spam advertisements or abusive behaviours. This entails approving advertisements before they are published, deleting reported advertisements and deleting users with abusive behaviours.

Advertisement

An announcement posted by a registered user about the sale of an item or the availability of a service offered by that user. An advertisement contains information about the place, price and description of the item or service.

Report

A notification sent from a registered user to the administrators about an advertisement's violation of the website's policy.

User

Any person who has a registered account on the website but is not an administrator. Users are able to perform multiple tasks with their account and advertisements.

Visitor

A person who is interested in viewing advertisements on the website but does not have an account.

1.3 Supplementary specifications

Objectives

The purpose of this document is to define requirements of the Chotot.com system. This Supplementary Specification lists the requirements that are not rapidly captured in the use case of the use-case model. The Supplementary Specification and the use-case model together capture a complete set of requirements on the system.

Scope

This Supplementary Specification applies to the Chotot.com system, which is an online advertisement website in Vietnam.

This specification defines the non-functional requirements of the system: such as reliability, usability, performance, and supportability, as well as functional requirements that are common across a number of use cases. (The functional requirements are defined in the Use Case Specification.)

References

None.

Functionality

Multiple users must be able to perform their work concurrently.

Usability

The software must be easy to use so that a new user can learn how to use the system within 1 hour.

The user interface has to be friendly and intuitive.

Reliability

The system must be available 24 hours a day, 7 days a week. The system must also have less than 5% downtime.

Performance

The system shall support up to 2000 simultaneously users against the central database, and up to 100 simultaneous users against the local servers at any given time.

The system shall provide access to the database with no more than 5 seconds latency.

The system must be able to complete 90% of all transactions within 30 seconds.

Supportability

None.

Security

The system must prevent users from logging in if they do not have a password.

Only Administrators can delete User accounts.

An advertisement can only be edited by its owner and deleted by its owner or an Administrator.

Only Administrators can approve pending advertisements.

Design Constraints

The system must provide a responsive web-based interface usable on computers and mobile devices.

1.4 Use-case models

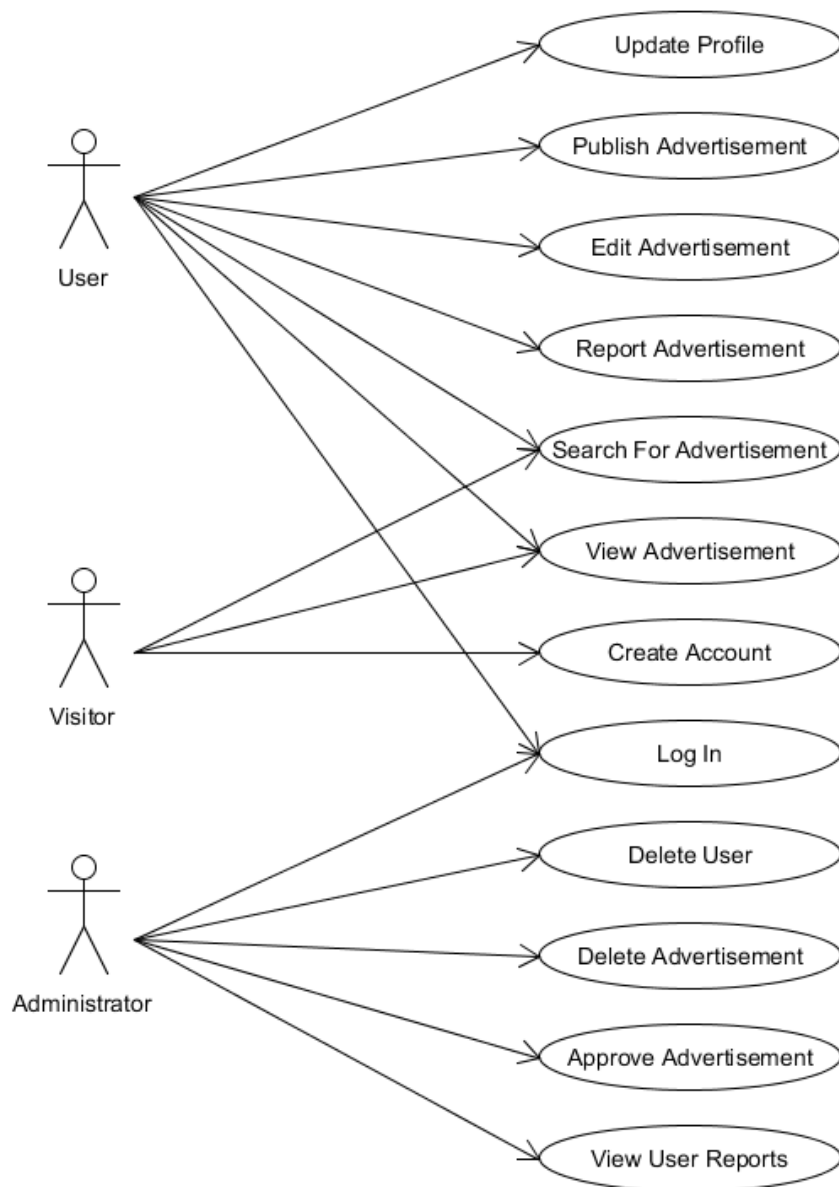


Figure 1-1. Use-case model for the Chotot.com website system

1.4.1 Approve Advertisements

Brief Description

This use case describes how an Administrator views the list of advertisements waiting for approval and approves or deletes them.

Flow of Events

Basic Flow

This use case starts when the Administrator requests to view the list of pending advertisements.

1. The system displays a list of unapproved advertisements, showing the information about the advertisement and user specified in the **View Advertisement** use case. The system also displays two actions, “Publish” or “Delete”.

2. If the Administrator selects “Publish”, the **Publish Advertisement** sub-flow is executed. If the Administrator selects “Delete”, the **Delete Advertisement** sub-flow is executed.

Approve Advertisement

1. The system asks the Administrator to confirm the decision.
2. The Administrator verifies the approval of the advertisement.
3. The system changes the advertisement’s status to “approved”.

Delete Advertisement

1. The system asks the Administrator to confirm the decision.
2. The Administrator verifies the deletion of the advertisement.
3. The system deletes the advertisement.

Alternative Flows

Operation Cancelled

In one of the sub-flows, if the Administrator selects “Cancel”, the advertisement is not published or deleted.

Special Requirements

None.

Pre-Conditions

The Administrator must be logged into the system.

Post-Conditions

If the use case is successful, the advertisement is published or removed. Otherwise, it is unchanged.

Extension Points

None.

1.4.2 Create Account

Brief Description

This use case describes how a Visitor registers for an account on the Chotot.vn website.

Flow of Events

Basic Flow

This use case starts when the Visitor requests to create an account on the website.

1. The system displays a form that asks the Visitor to enter the following information:
 - Unique user ID
 - Password
 - Name
 - Phone number
 - Email address
 - Brief self-introduction (optional)

2. Once the Visitor provides the requested information, the system verifies that the user ID is unique and all required fields are specified. It then adds a new account with the specified information to the user database.

Alternative Flows

Missing Information

If any of the above fields (except for self-introduction) are not filled in, the system displays an error message. The Visitor can continue making changes to the registration form or cancel the registration, at which point the use case ends.

User ID Already Exists

If the specified user ID already exists, the system displays an error message. The Visitor can continue making changes to the registration form or cancel the registration, at which point the use case ends.

Special Requirements

None.

Pre-Conditions

None.

Post-Conditions

If the use case was successful, a new user is added to the system. Otherwise, the system state remains unchanged.

Extension Points

None.

1.4.3 Delete Advertisement

Brief Description

This use case allows an Administrator to remove an Advertisement from the system.

Flow of Events

Basic Flow

This use case starts when the Administrator wishes to remove an Advertisement from the system.

1. The system requests that the Administrator specify the advertisement ID.
2. The Administrator enters the advertisement ID. The system retrieves and display the advertisement's information.
3. The system prompts the Administrator to confirm the deletion of the advertisement.
4. The Administrator verifies the deletion.
5. The system deletes the advertisement from the system.

Alternative Flows

Advertisement Not Found

If the advertisement with the specified ID does not exist, the system displays an error message. The Administrator can then enter a different ID or cancel the operation, at which point the use case ends.

Delete Cancelled

If the Administrator decides not to delete the advertisement, the removal is cancelled and the use case is re-started at the beginning.

Pre-Conditions

The Administrator must be logged onto the system before this use case begins.

Post-Conditions

If the use case was successful, the advertisement is removed from the system. Otherwise, the system state is unchanged.

Extension Points

None.

1.4.4 Delete User

Brief Description

This use case allows an Administrator to remove a User from the system.

Flow of Events

Basic Flow

This use case starts when the Administrator wishes to remove a User from the system.

1. The system requests that the Administrator specify the user ID.
2. The Administrator enters the user ID. The system retrieves and display the user's information.
3. The system prompts the Administrator to confirm the deletion of the user.
4. The Administrator verifies the deletion.
5. The system deletes the user from the system.

Alternative Flows

User Not Found

If the user with the specified ID does not exist, the system displays an error message. The Administrator can then enter a different ID or cancel the operation, at which point the use case ends.

Delete Cancelled

If the Administrator decides not to delete the user, the removal is cancelled and the use case is re-started at the beginning.

Pre-Conditions

The Administrator must be logged onto the system before this use case begins.

Post-Conditions

If the use case was successful, the user and the user's published advertisements are removed from the system. Otherwise, the system state is unchanged.

Extension Points

None.

1.4.5 Edit Advertisement

Brief Description

This use case describes how a User edits an advertisement that he/she previously published.

Flow of Events

Basic Flow

This use case starts when the User requests to edit an advertisement posted by him/her.

1. The system displays the advertisement information in editable fields.
2. If the User selects "Unpublish", the **Unpublish Advertisement** subflow is executed. Otherwise, the **Update Advertisement Information** subflow is executed.

Unpublish Advertisement

1. The system asks the User to confirm the deletion of the advertisement.
2. The User verifies the deletion.
3. The system removes the advertisement from the system.

Update Advertisement Information

1. The User makes the desired changes to the advertisement information, including any of the information specified in the Publish Advertisement use case.
2. The User selects "Update".
3. The system updates the advertisement record and marks the advertisement as not approved, effectively scheduling it for approval.

Alternative Flows

Edit Cancelled

If the user selects "Cancel", the advertisement is not updated or deleted and the use case ends.

Missing Advertisement Information

If, in the **Update Advertisement Information** sub-flow, the User did not specify all the required information, the system displays an error message. The use case continues.

Unpublish Cancelled

If, in the **Unpublish Advertisement** sub-flow, the User decides not to unpublish the advertisement, the advertisement is not removed from the system and the use case continues.

Special Requirements

None.

Pre-Conditions

The user must be logged onto the system and an advertisement he/she has published is being displayed.

Post-Conditions

If the use case is successful, the advertisement is updated or removed. Otherwise, it is unchanged.

Extension Points

None.

1.4.6 Log In

Brief Description

This use case describes how a User or an Administrator logs into the Website.

Flow of Events

Basic Flow

This use case starts when the actor wishes to log into the Website.

1. The actor enters his/her ID and password.
2. The system validates the entered ID and password and logs the actor into the system.

Alternative Flows

Invalid Information

If, in the **Basic Flow**, the actor enters an invalid ID and/or password, the system displays an error message. The actor can choose to either return to the beginning of the **Basic Flow** or cancel the login, at which point the use case ends.

Missing Required Information

If any fields in the form are left empty, the system displays an error message. The actor can continue modifying the form or cancel the operation, at which point the use case ends.

Special Requirements

None.

Pre-Conditions

The system is in the login state and has the login screen displayed.

Post-Conditions

If the use case is successful, the actor is now logged into the system. Otherwise, the system state is unchanged.

Extension Points

None.

1.4.7 Process User Reports

Brief Description

This use case describes how an Administrator views the list of advertisements being reported for abuse and deletes the advertisement or remove the report.

Flow of Events

Basic Flow

This use case starts when the Administrator requests to view the list of reports from users.

1. The system displays a list of reports, showing the information about the advertisement, user and report specified in the **Report Advertisement** use case, with a link to the reported advertisement. The system also displays two actions, “Delete Advertisement” or “Remove Report”.
2. If the Administrator selects “Delete Advertisement”, the **Delete Advertisement** sub-flow is executed. If the Administrator selects “Remove Report”, the **Remove Report** sub-flow is executed.

Delete Advertisement

1. The system asks the Administrator to confirm the decision.
2. The Administrator verifies the deletion of the advertisement.
3. The system removes the advertisement.

Remove Report

1. The system asks the Administrator to confirm the decision.
2. The Administrator verifies the deletion of the report.
3. The system deletes the report.

Alternative Flows

Operation Cancelled

In one of the sub-flows, if the Administrator selects “Cancel”, the advertisement and the report are not removed from the system.

Special Requirements

None.

Pre-Conditions

The Administrator must be logged into the system.

Post-Conditions

If the use case is successful, the advertisement or the report is removed. Otherwise, the system state remains unchanged.

Extension Points

None.

1.4.8 Publish Advertisement

Brief Description

This use case describes how a User publishes an advertisement to the website.

Flow of Events

Basic Flow

This use case starts when the User requests to publish an advertisement.

1. The system displays a form and requests the User to complete the following information:
 - Title
 - Category
 - Province or Municipality
 - Town, District or City
 - Price
 - Description
 - Picture (optional)
2. Once the User provides the requested information, the system generates a new unique advertisement ID and adds a new advertisement to the approval queue.

Alternative Flows

Missing Required Information

If any of the above fields are not filled in or selected (except for Picture), the system displays an error message. The User can continue making changes to the form or cancel the publish, at which point the use case ends.

Special Requirements

None.

Pre-Conditions

The User must be logged into the system.

Post-Conditions

If the use case was successful, an advertisement is created and scheduled for approval. Otherwise, the system state remains unchanged.

1.4.9 Report Advertisement

Brief Description

This use case describes how a person reports an advertisement published on the website.

Flow of Events

Basic Flow

This use case starts when the User clicks on the “Report” button when viewing an advertisement.

1. The system displays a form requesting the User to select a reason for the report (duplicate advertisement, misleading advertisement, the item has been sold or the

publisher cannot be contacted) and enter more information about the report, an email address and a phone number.

2. Once the User provides the requested information, the system generates a new unique report ID and stores the report in the report database.

Alternative Flows

Missing Required Information

If any fields in the form are left empty, the system displays an error message. The User can continue modifying the form or cancel the operation, at which point the use case ends.

Special Requirements

None.

Pre-Conditions

The User is viewing an advertisement that was not published by his/herself and has clicked on the “Report” button for the advertisement.

Post-Conditions

If the use case is successful, a report is generated, containing all the information specified by the user, plus a unique ID and the advertisement’s ID. Otherwise, the system state remains unchanged.

Extension Points

None.

1.4.10 Search For Advertisements

Brief Description

This use case describes how a User or Visitor searches for advertisements that meets their interests.

Flow of Events

Basic Flow

This use case starts when the User/Visitor wishes to search for advertisements.

1. The system display an editable form with the following search criteria:
 - Title
 - Category
 - Province or Municipality
 - Town, District or City
 - Price range
 - Description
 - User ID
2. The User/Visitor enters the title and description; selects the category, address and price range. Once the User chooses to do a search, the system performs a query on the advertisement database and displays a list of result.

Alternative Flows

Missing Required Information

If no search criteria are specified, the system displays an error message. The User/Visitor can adjust their search criteria or cancel the search, at which point the use case ends.

Special requirements

None.

Pre-Conditions

None.

Post-Conditions

If the use case was successful, a list of advertisements satisfying the search criteria is displayed.

1.4.11 Update Account

Brief Description

This use case describes how a User updates his/her account in the system.

Flow of Events

Basic Flow

This use case starts when the User requests to update his/her account information.

1. The system displays the following information in editable fields and asks the User to make changes to their account:
 - Password
 - Name
 - Phone number
 - Email address
 - Brief self-introduction (optional)
2. Once the User provides the requested information, the system verifies that all required fields are specified, and updates the account's information on the user database.

Alternative Flows

Missing Information

If any of the above fields (except for self-introduction) are not filled in, the system displays an error message. The actor can continue making changes to the registration form or cancel the registration, at which point the use case ends.

Special Requirements

None.

Pre-Conditions

The User must be logged into the system.

Post-Conditions

If the use case was successful, the account's information is updated. Otherwise, the system state remains unchanged.

Extension Points

None.

1.4.12 View Advertisement**Brief Description**

This use case describes how a User or Visitor views an advertisement published on the website.

Flow of Events

This use case starts when the User/Visitor goes to the link of an advertisement.

The system displays detailed information about the advertisement as stated in the Publish Advertisement use case and information about the user who published the advertisement, including his/her name, address, phone number and email address.

Special Requirements

None.

Pre-Conditions

None.

Post-Conditions

None.

Extension Points

None.

2 Use-case analysis

2.1 Architectural analysis

2.1.1 High-level organisation of the model

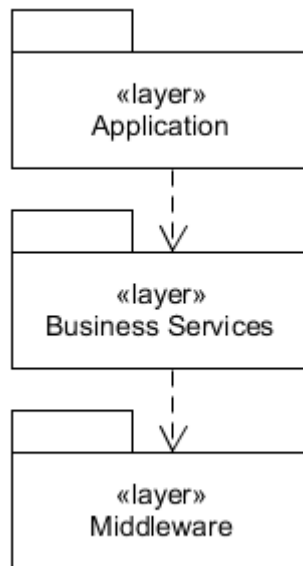


Figure 2-1. Layering approach

The above figure describes the high-level organisation of the software system. The system consists of three layers:

- The *Application* layer contains the design elements that are specific to each use case of the system.
- The *Business Services* layer encapsulates some key abstractions and services common to all use cases. It is accessible from the *Application* layer.
- The *Middleware* layer offers services to enable data communication and management on distributed systems.

2.1.2 Key abstractions

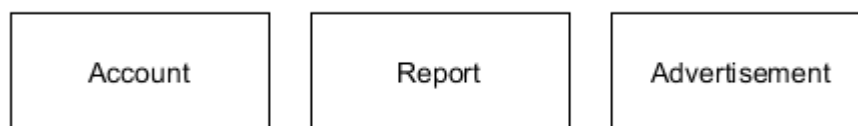


Figure 2-2. Key abstractions used in the application

- *Account*: A record about a user/administrator. Each account has a unique user ID and a password, which is used to identify the user/administrator and grant them access to secure parts of the system.

- *Advertisement*: An announcement posted by a registered user about the sale of an item or the availability of a service offered by that user.
- *Report*: A notification sent from a registered user to the administrators about an advertisement's violation of the website's policy.

2.2 Use-case realisations

2.2.1 Use-case realisations: sequence diagrams

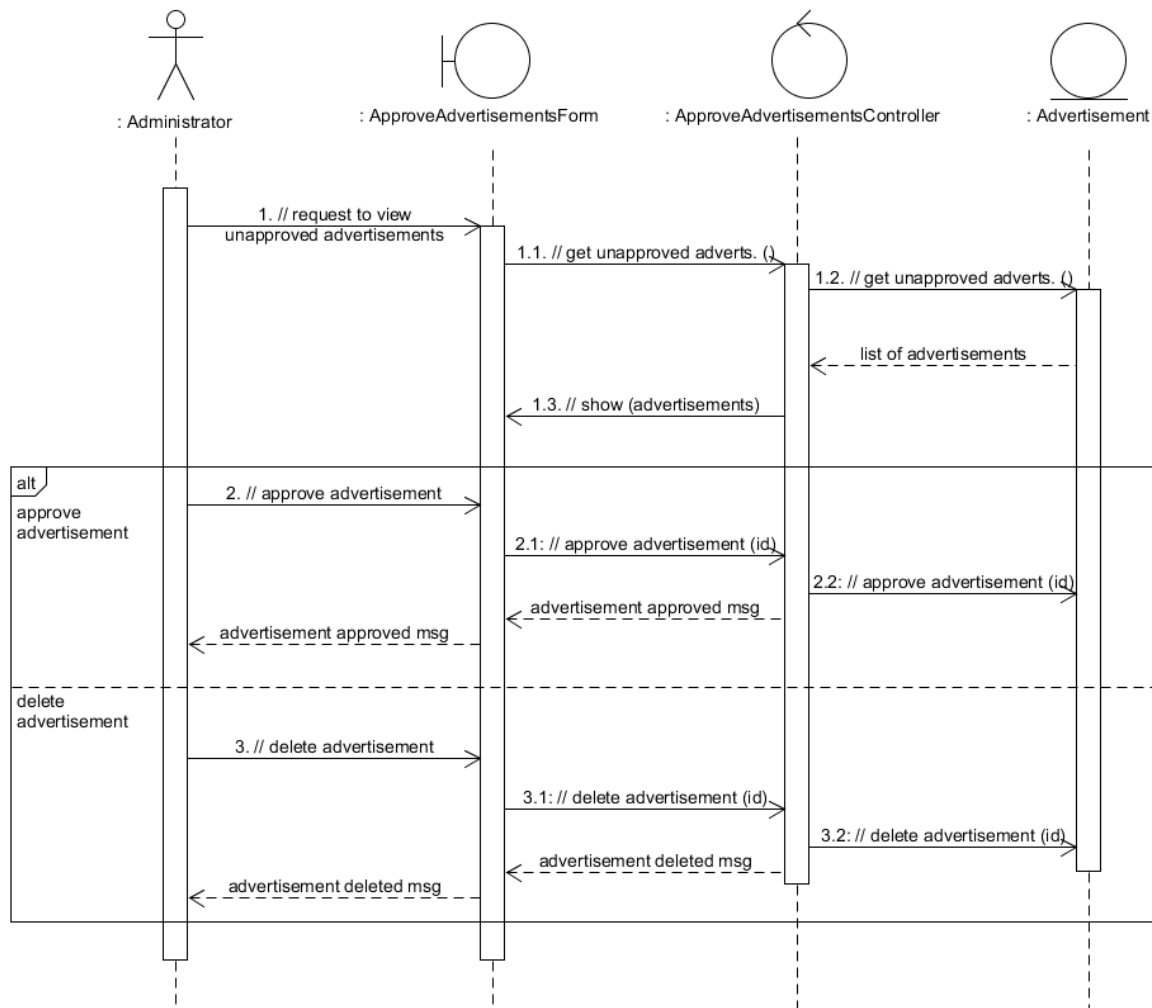


Figure 2-3. Sequence diagram for the Approve Advertisement use case

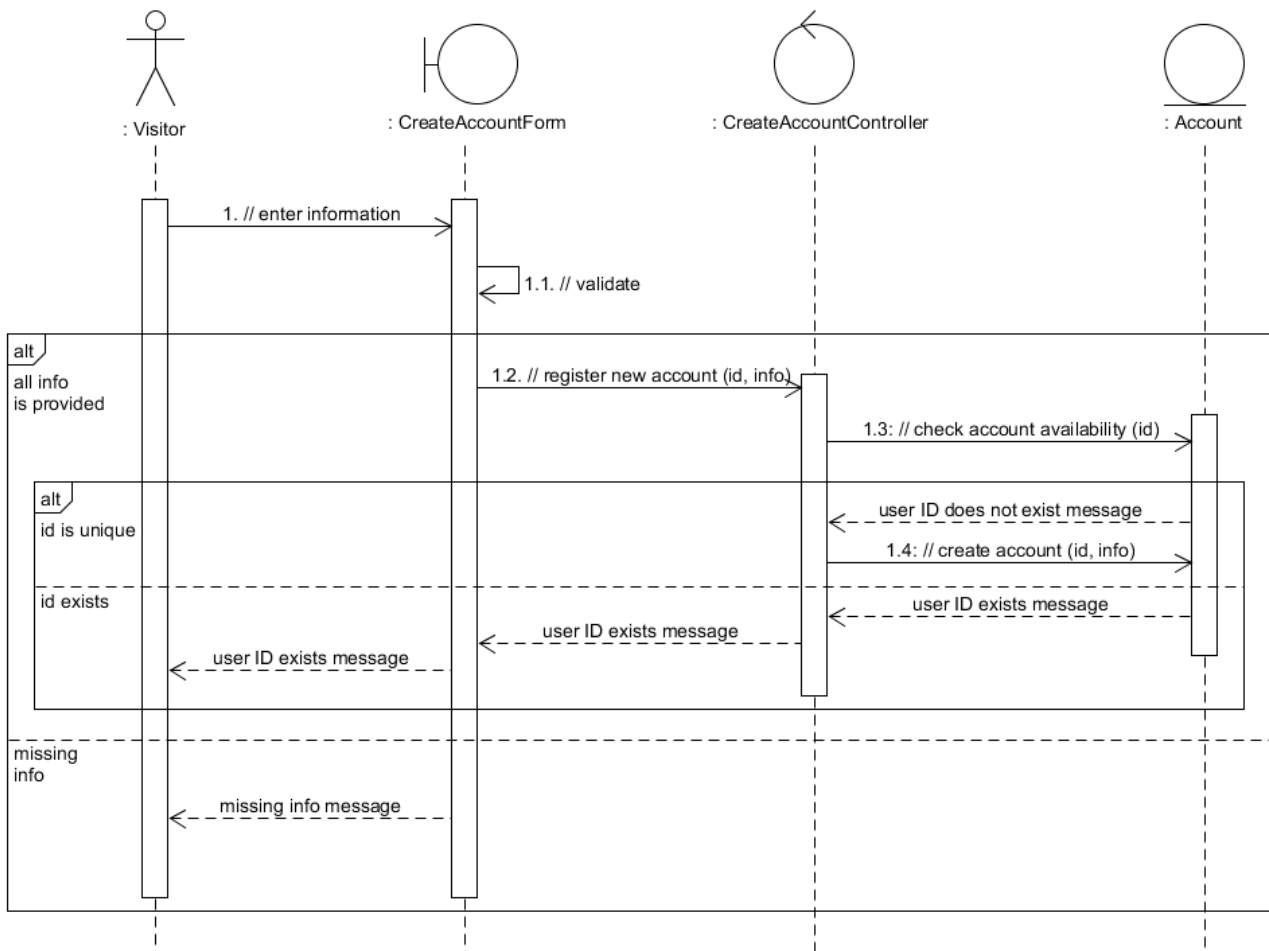


Figure 2-4. Sequence diagram for the Create Account use case

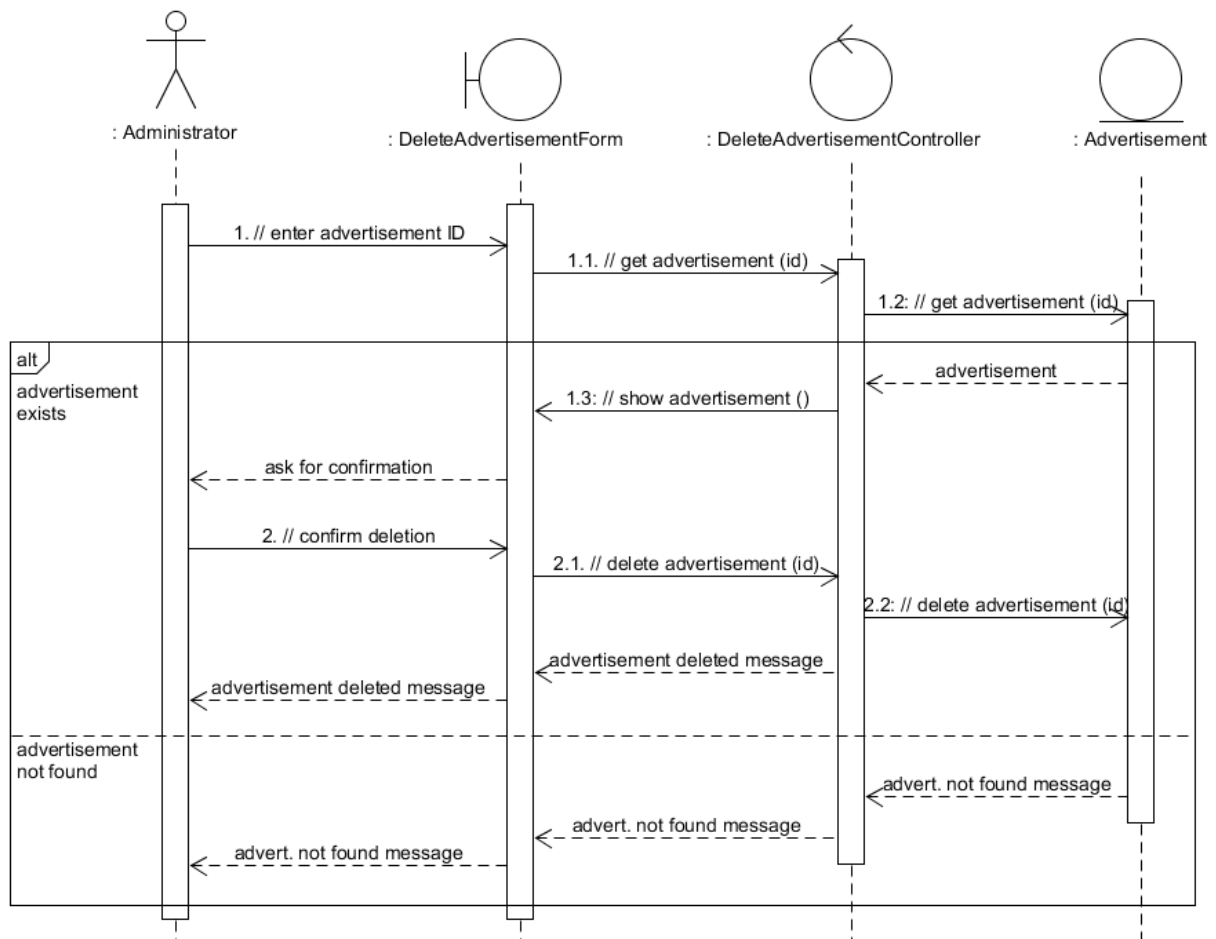


Figure 2-5. Sequence diagram for the Delete Advertisement use case

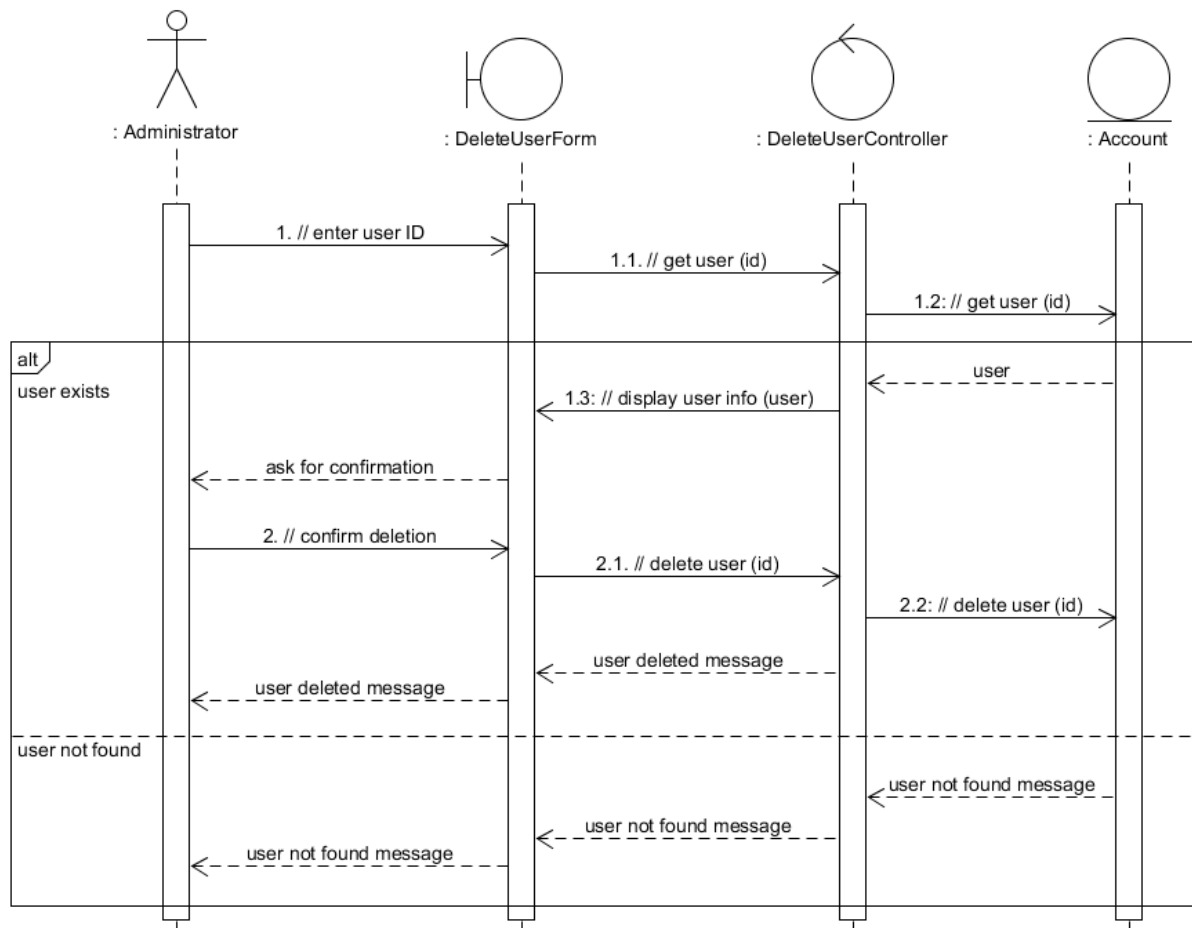


Figure 2-6. Sequence diagram for the Delete User use case

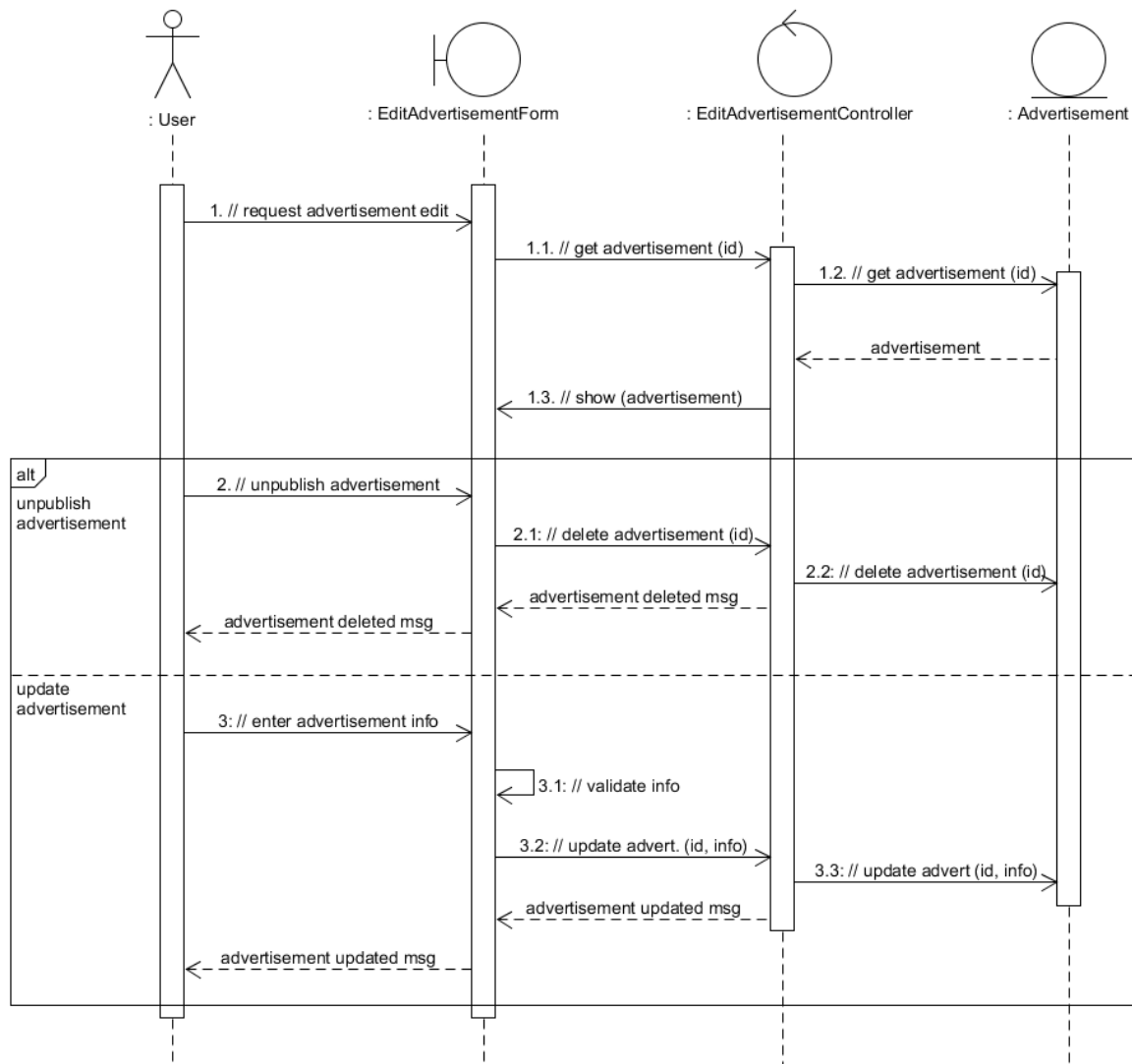


Figure 2-7. Sequence diagram for the Edit Advertisement use case

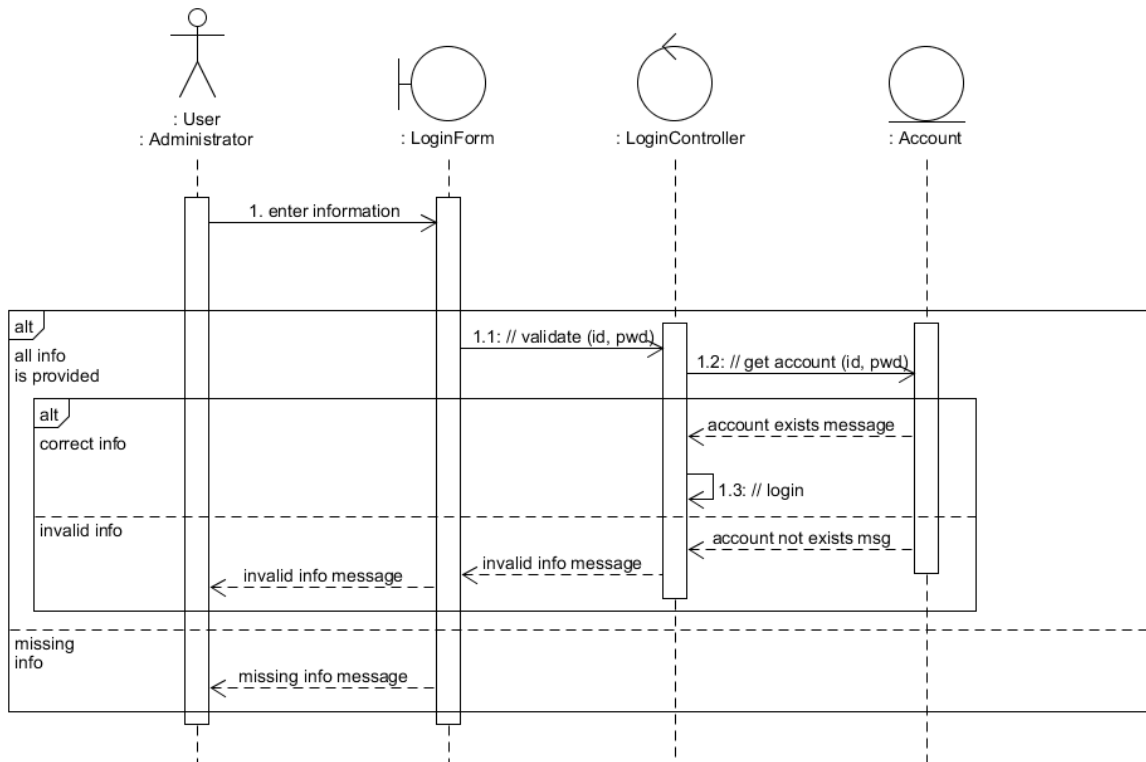


Figure 2-8. Sequence Diagram for the Log In use case

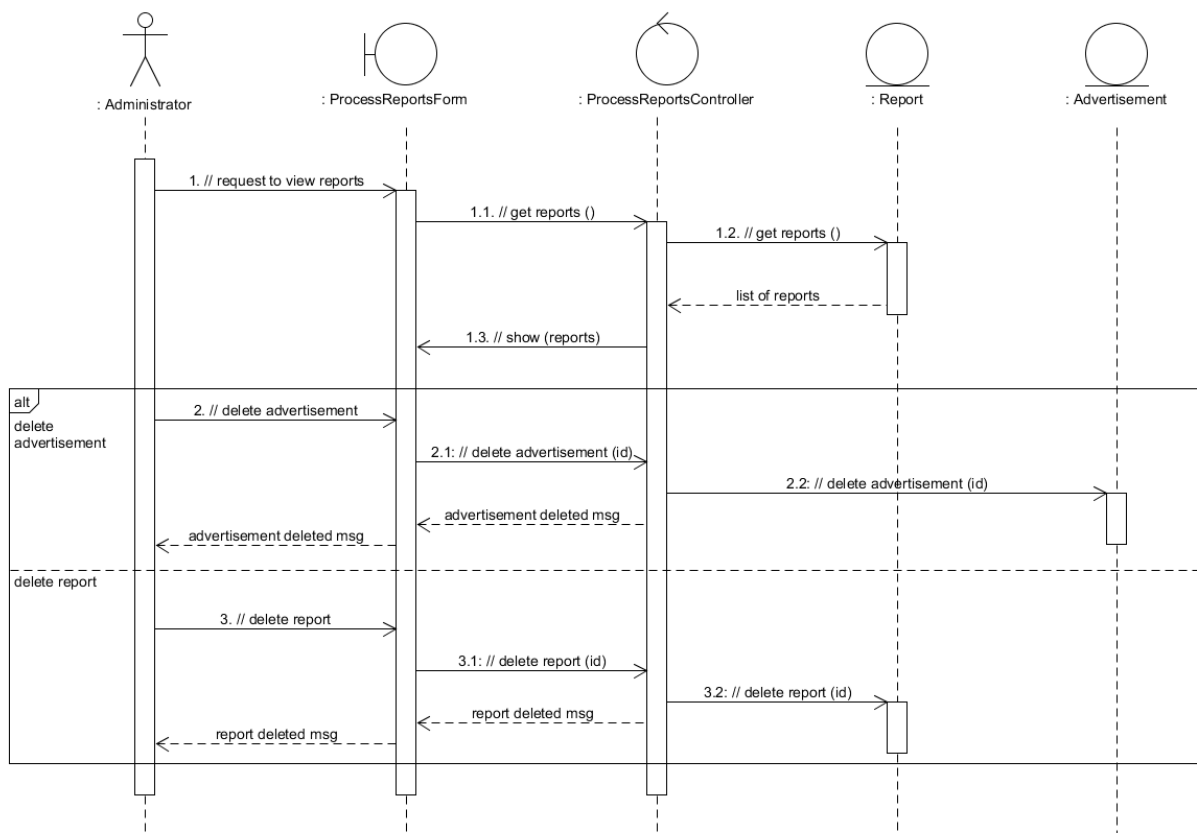


Figure 2-9. Sequence diagram for the Process User Reports use case

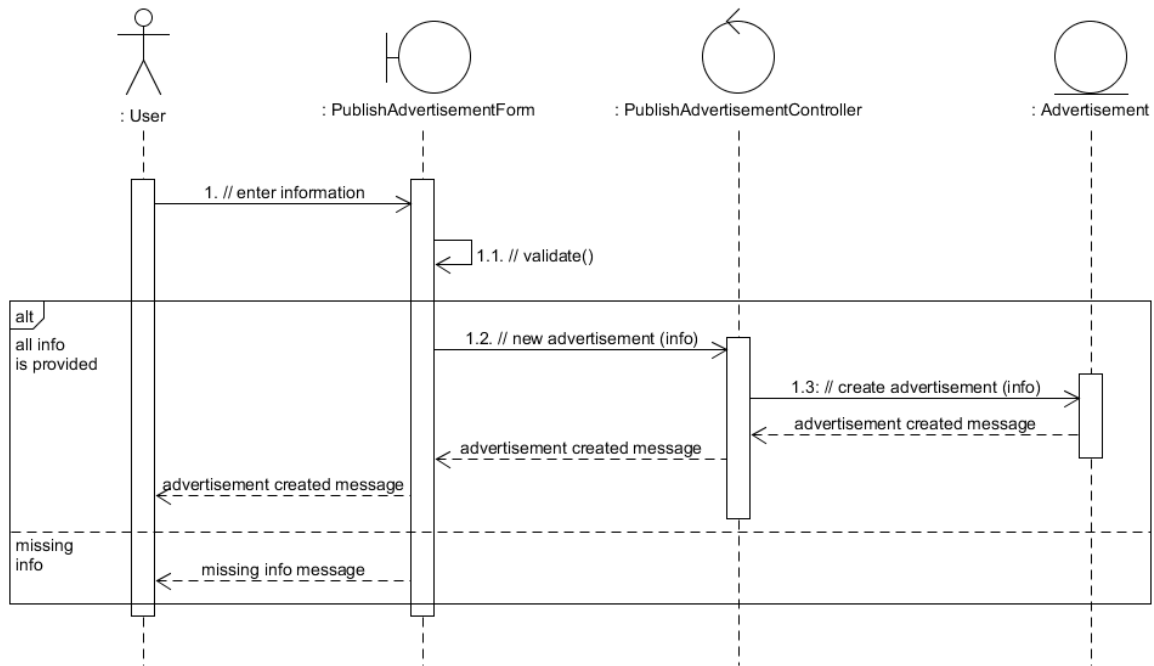


Figure 2-10. Sequence diagram for the Publish Advertisement use case

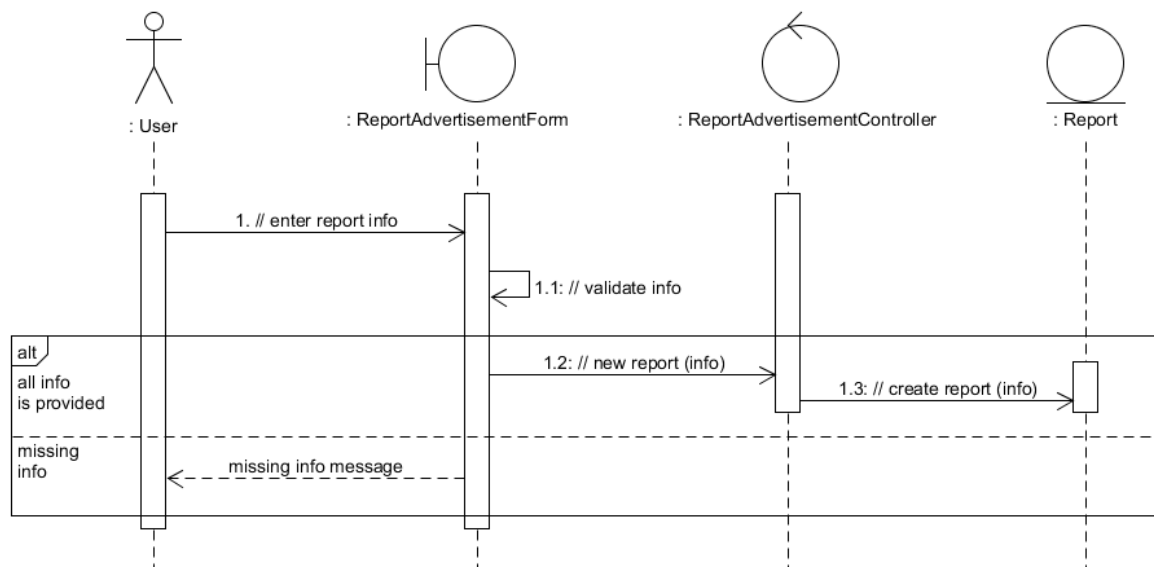


Figure 2-11. Sequence diagram for the Report Advertisement use case

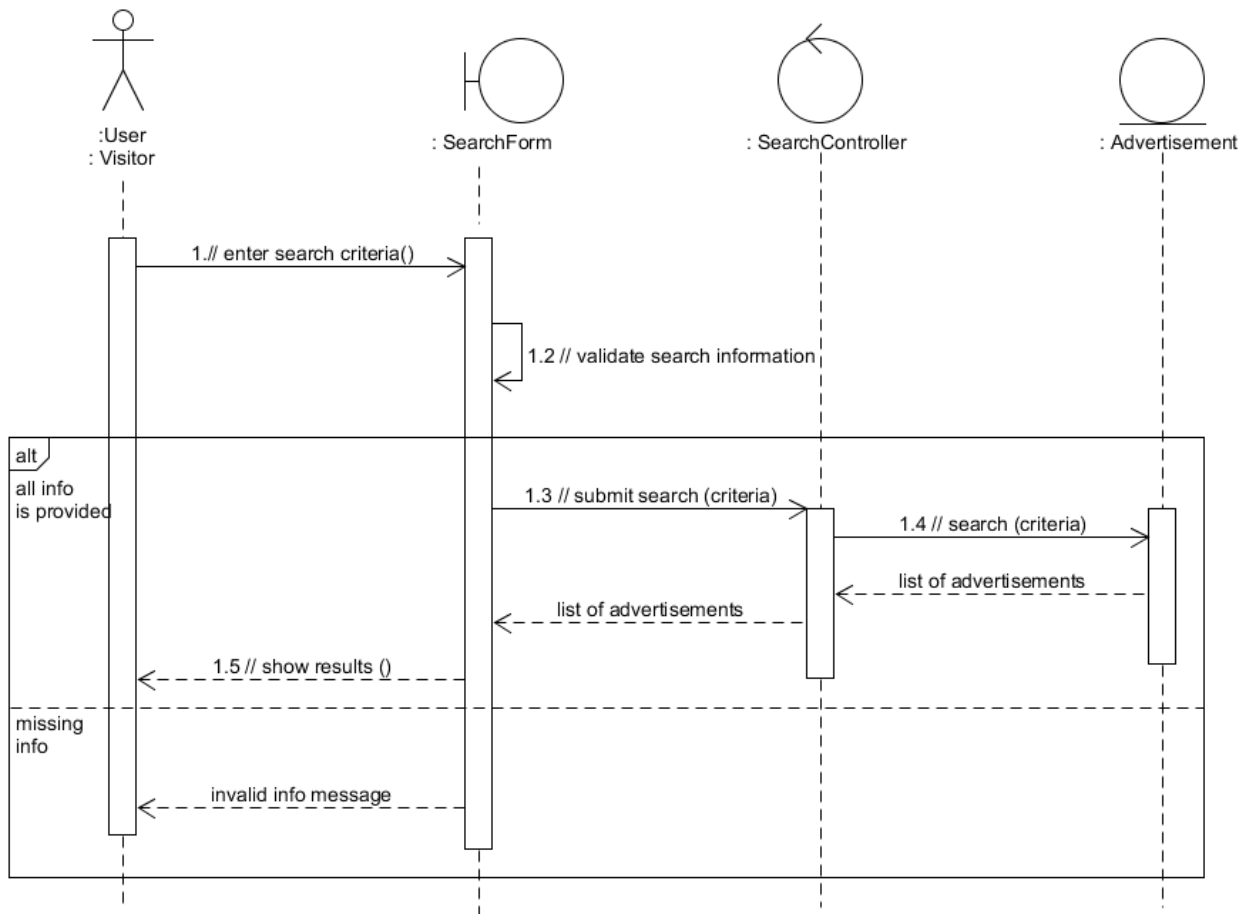


Figure 2-12. Sequence diagram for the Search For Advertisements use case

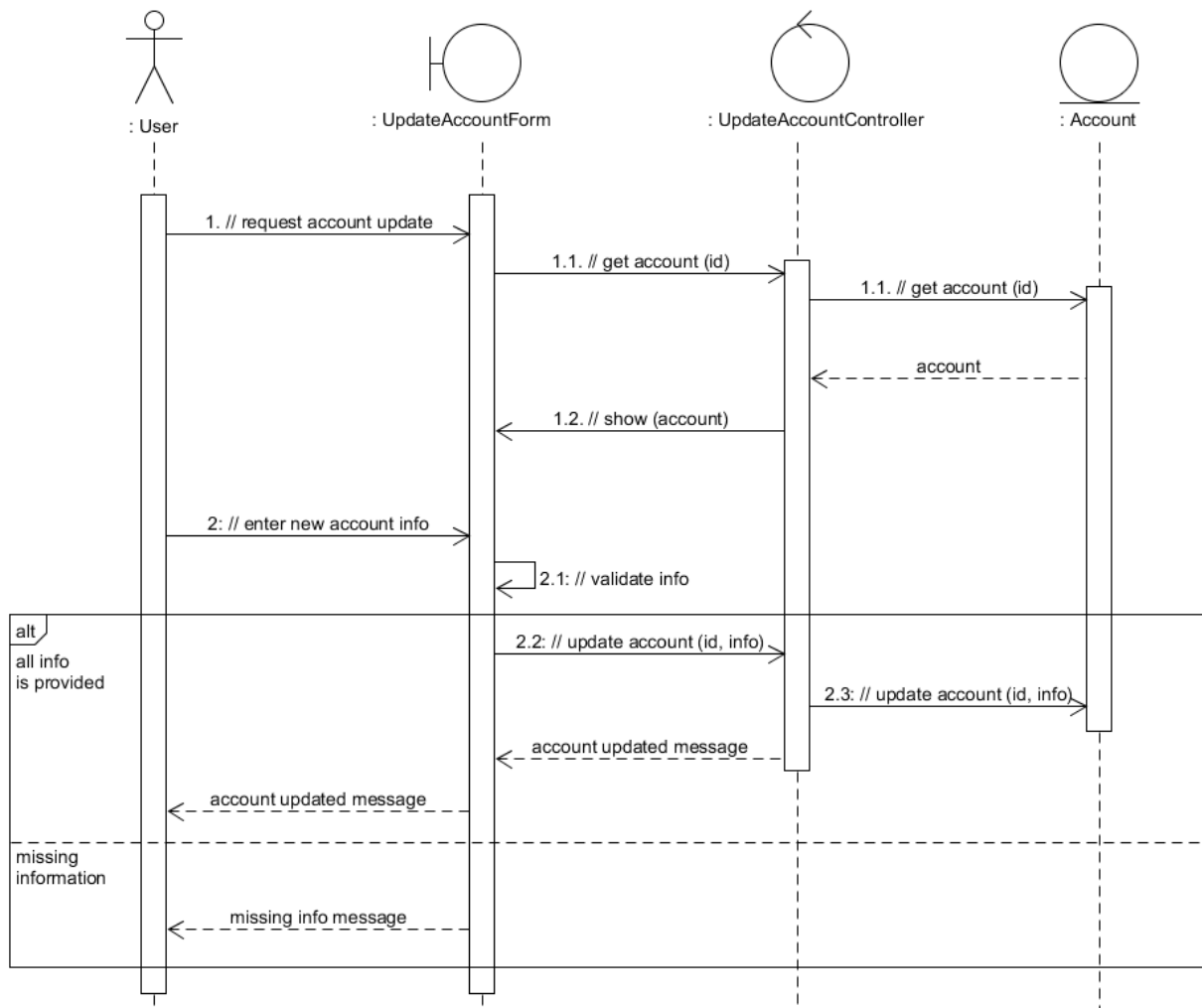


Figure 2-13. Sequence diagram for the Update Account use case

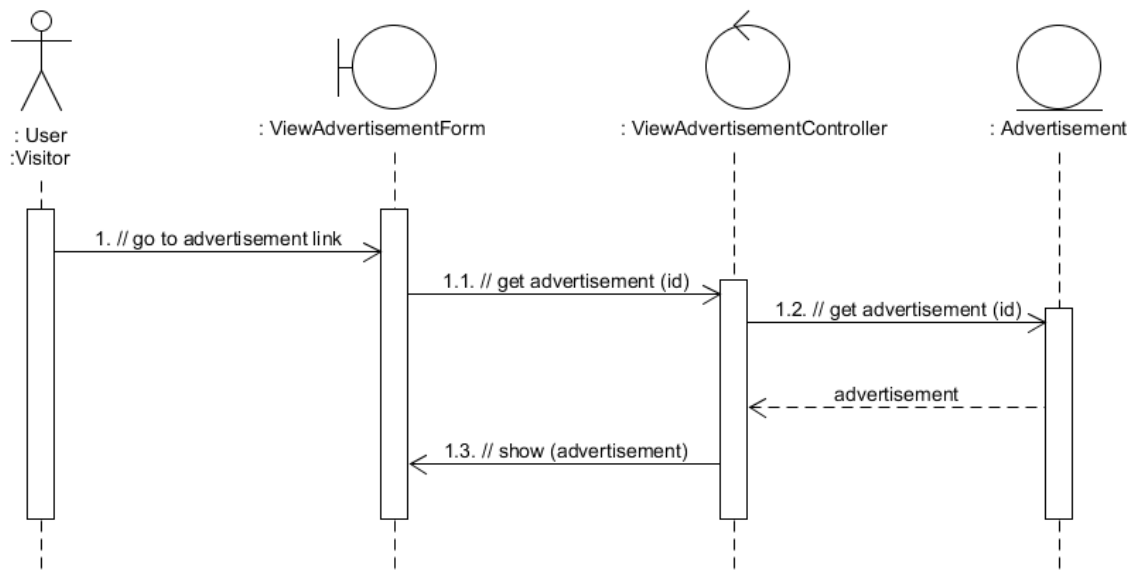


Figure 2-14. Sequence diagram for the View Advertisement use case

2.2.2 Use-case realisations: views of participating classes

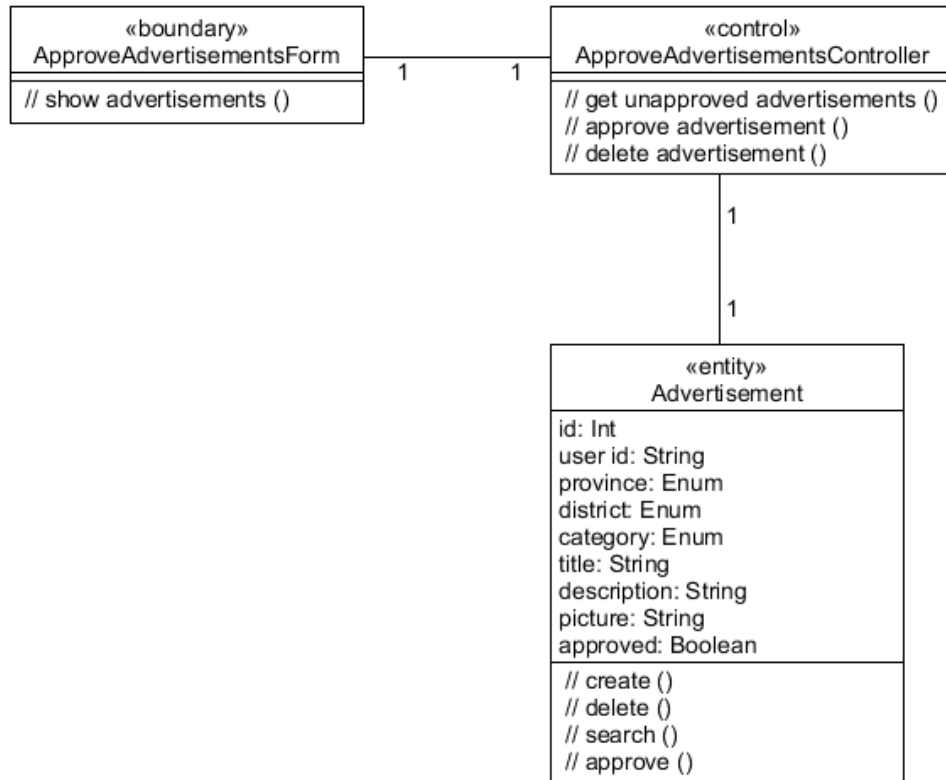


Figure 2-15. VOPC for the Approve Advertisement use case

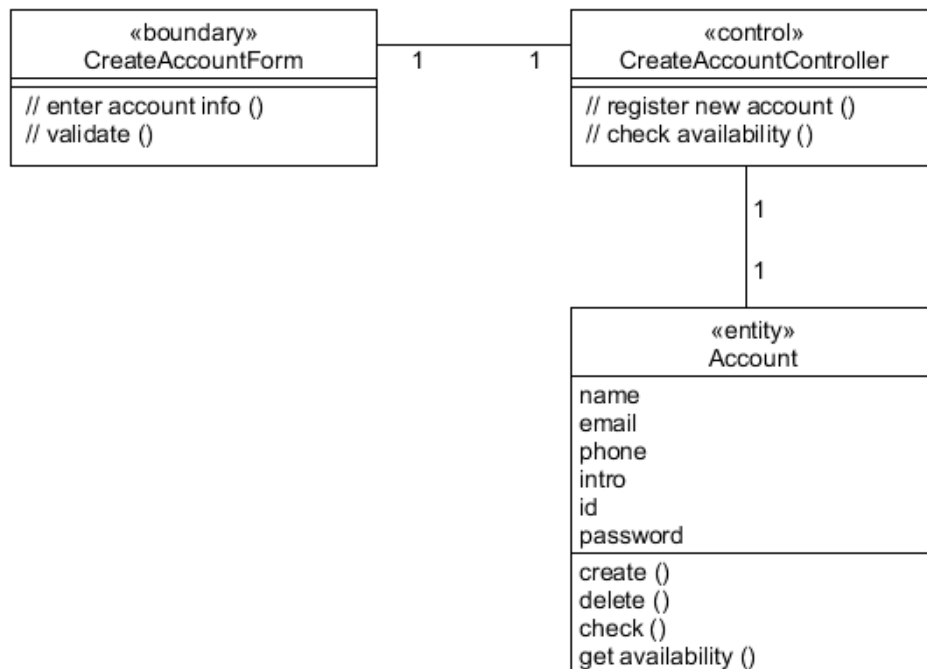


Figure 2-16. VOPC for the Create Account use case

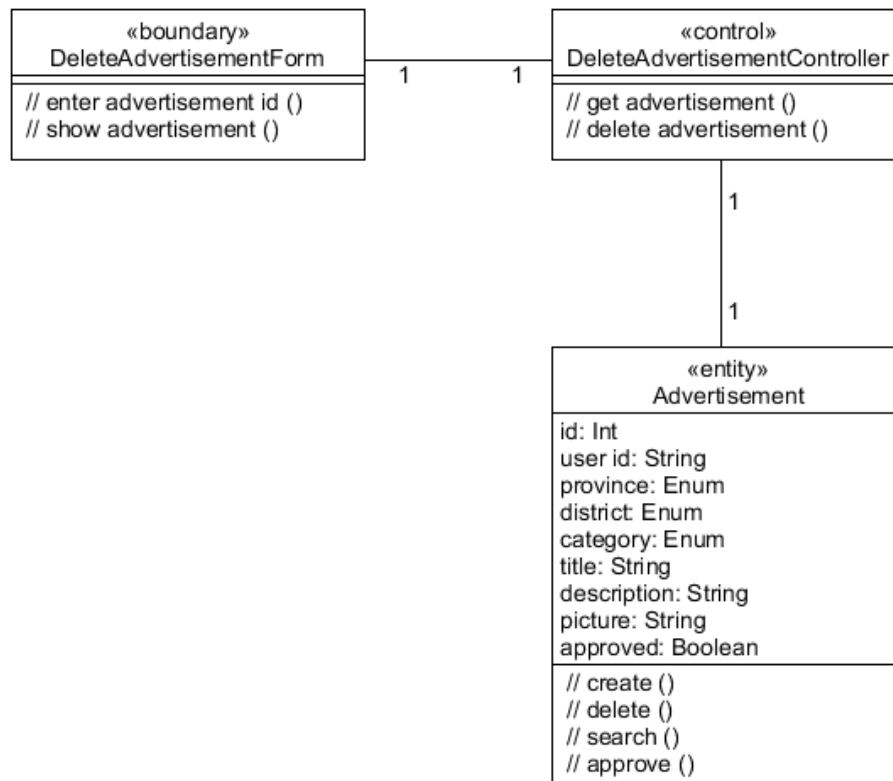


Figure 2-17. VOPC for the Delete Advertisement use case

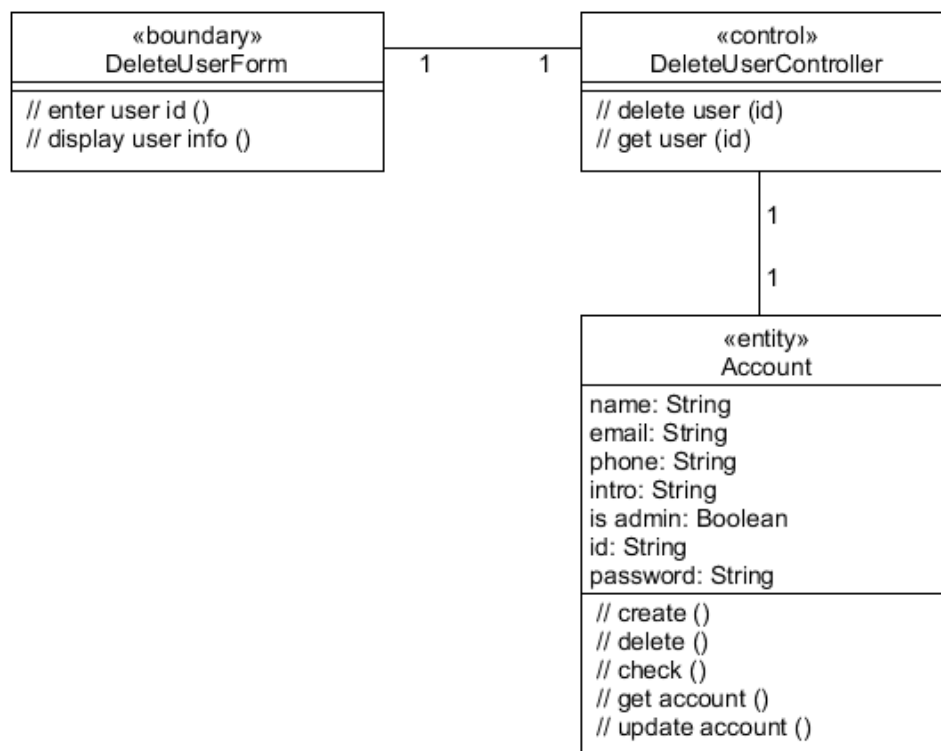


Figure 2-18. VOPC for the Delete User use case

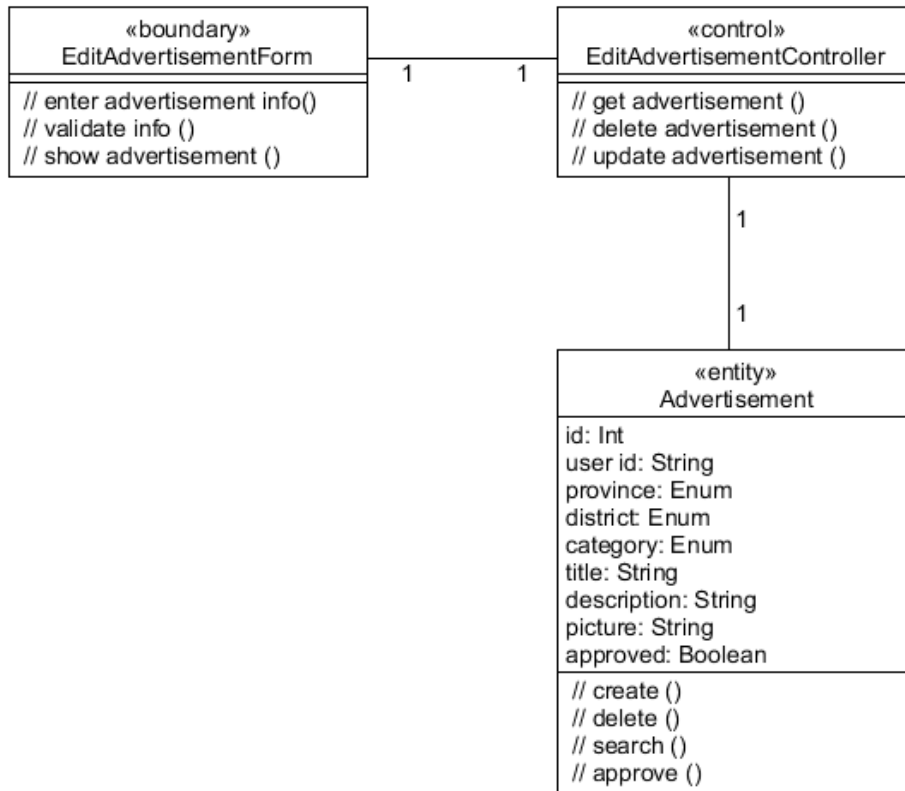


Figure 2-19. VOPC for the Edit Advertisement use case

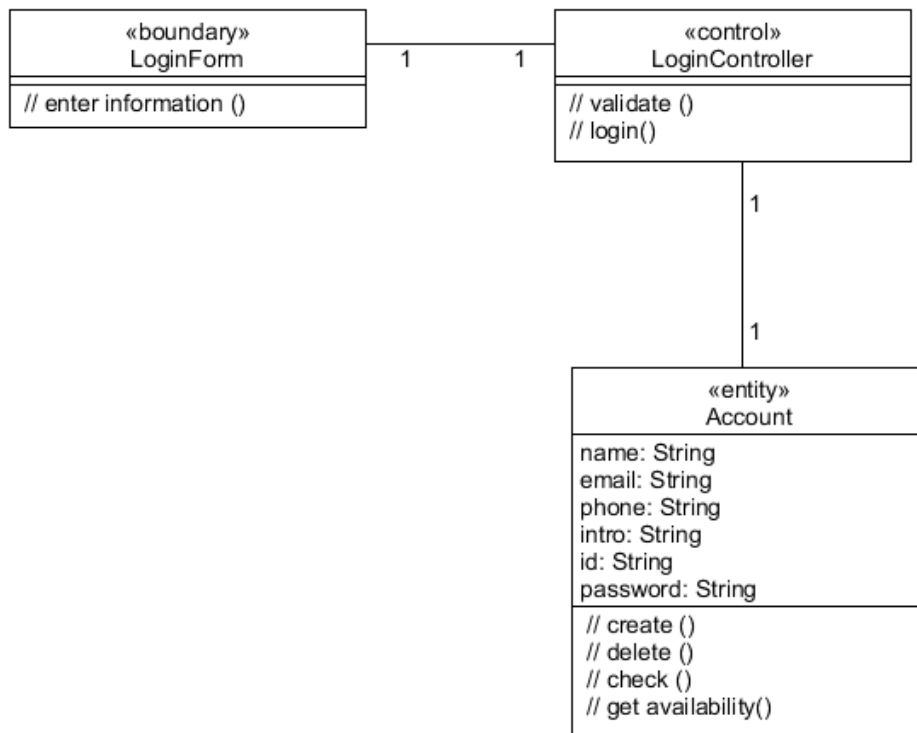


Figure 2-20. VOPC for the Log In use case

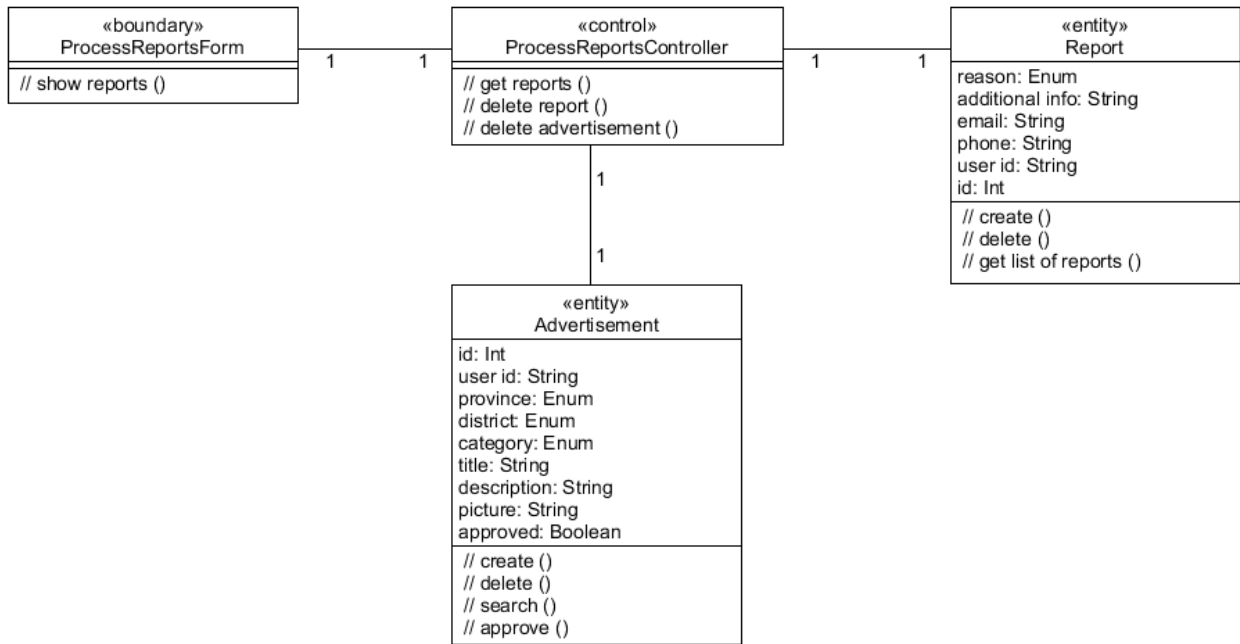


Figure 2-21. VOPC for the Process User Reports use case

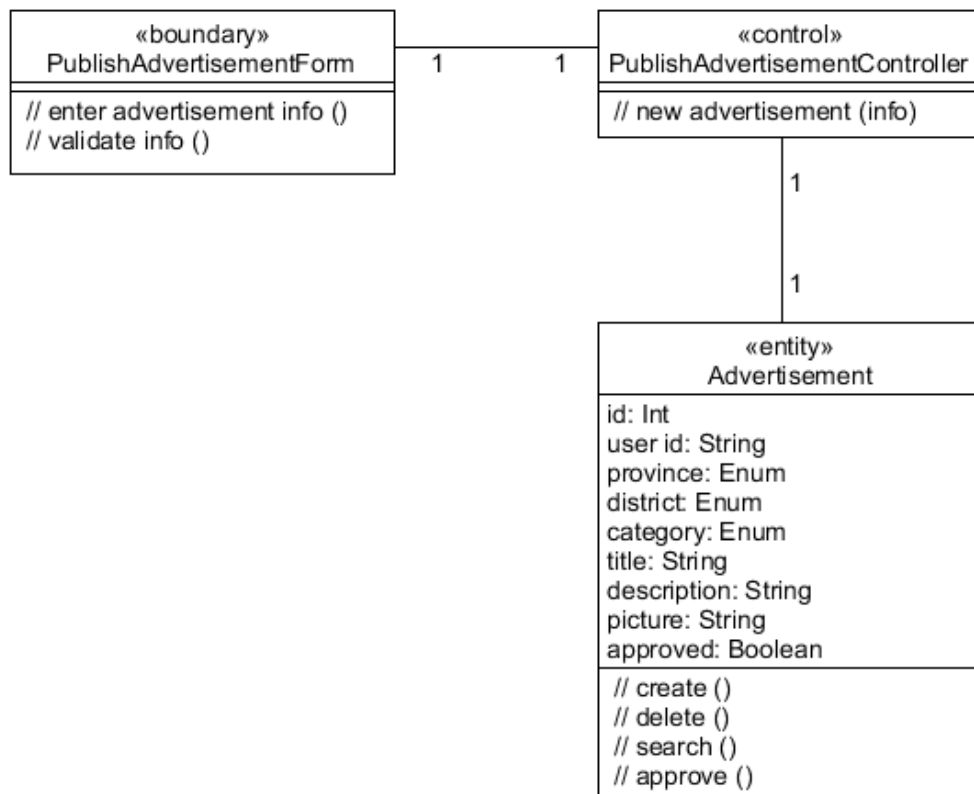


Figure 2-22. VOPC for the Publish Advertisement use case

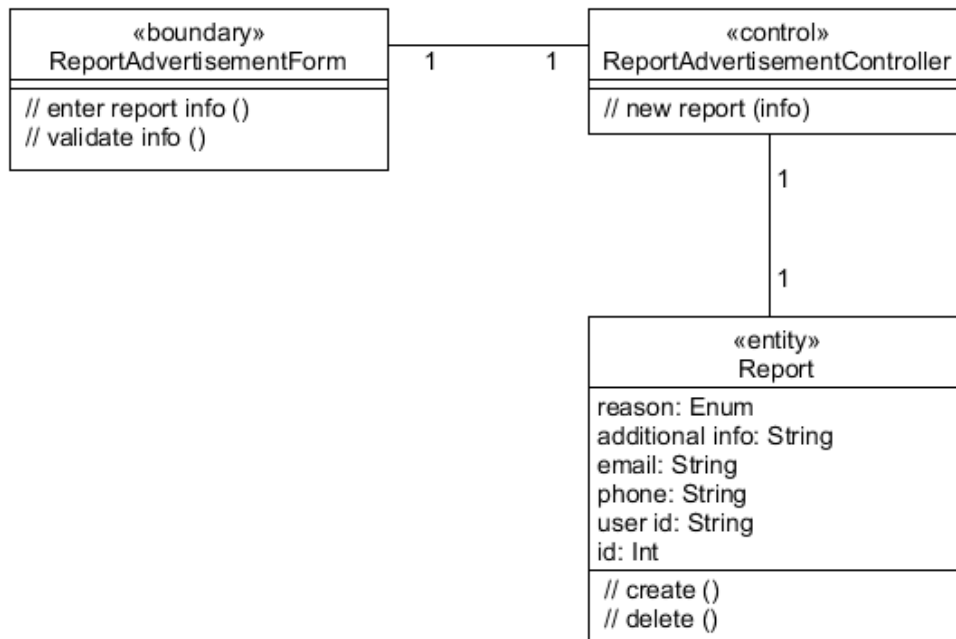


Figure 2-23. VOPC for the Report Advertisement use case

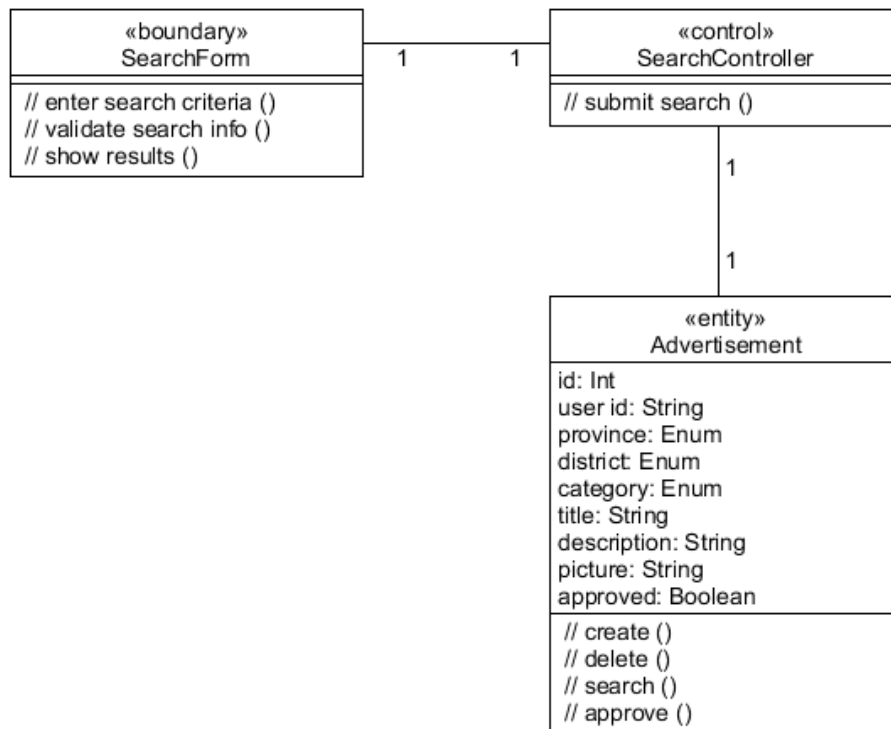


Figure 2-24. VOPC for the Search For Advertisement use case

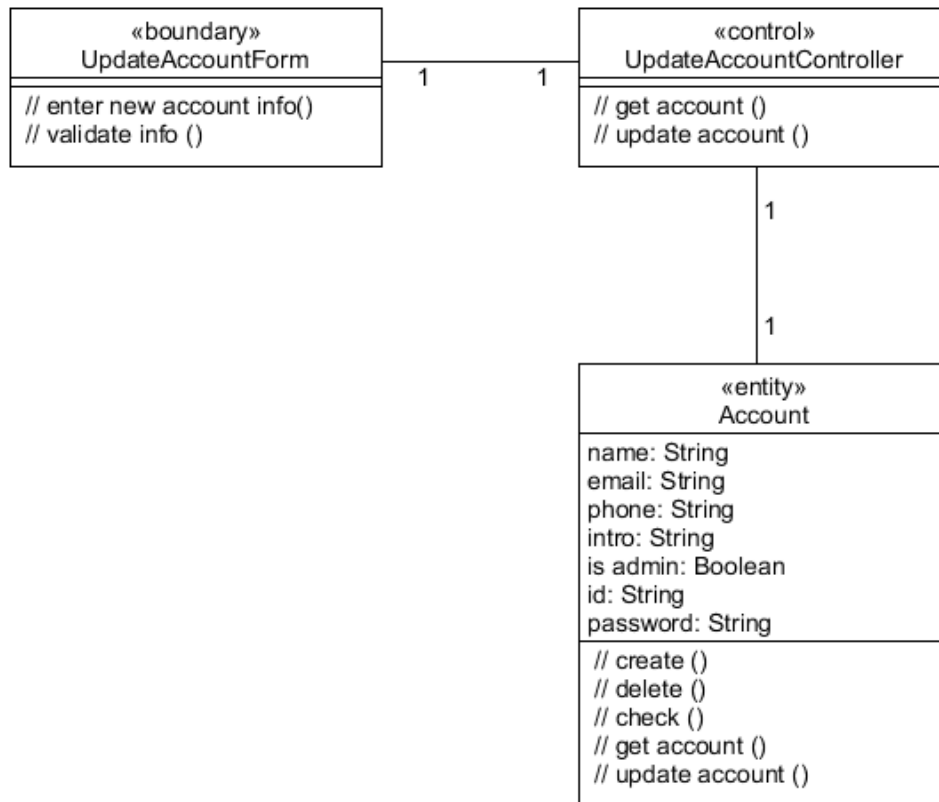


Figure 2-25. VOPC for the Update Account use case

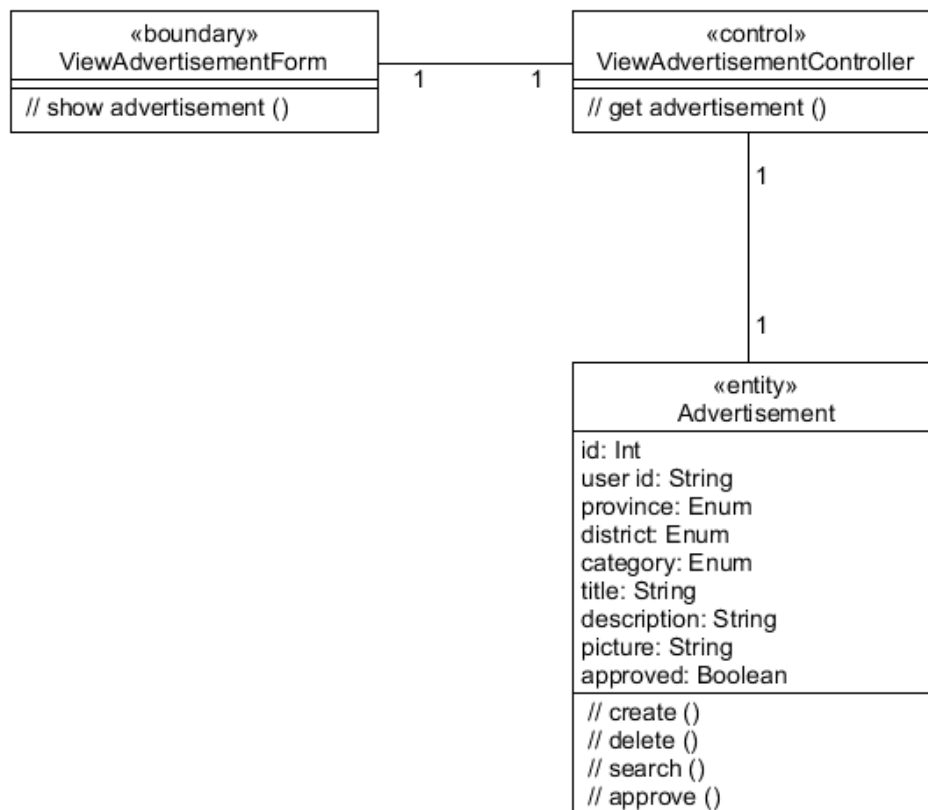


Figure 2-26. VOPC for the View Advertisement use case

2.2.3 Describe analysis mechanism

Analysis class	Analysis mechanism
Account	Persistency, Security
Advertisement	
Report	
CreateAccountController	Distribution
DeleteAdvertisementController	
DeleteUserController	
EditAdvertisementController	
LoginController	
ProcessReportsController	
PublishAdvertisementController	
ReportAdvertisementController	
SearchController	
UpdateAccountController	
ViewAdvertisementController	

Table 2-1. Analysis-Class-To-Analysis-Mechanism map

Analysis mechanism characteristics

Security

- Data granularity: attribute level
- User granularity: three roles – unregistered users, registered users and administrators
- Security rules:
 - Only registered users/administrators may log into the system.
 - Only logged in users may view and edit their own account profile.
 - Only logged in users may publish an advertisement.
 - An advertisement could only be edited by its owner.
 - Only Administrator may perform the following actions: view reports made by users, approve advertisements, delete users and delete spam advertisements.

Persistency

Class	Account	Report	Advertisement
Granularity	100 KB per product	1 to 4 KB per product	1 to 10 MB per product
Volume	Up to 100,000	Up to 1,000	Up to 500,000
Access frequency	Create: 1,000 per day Update: 500 per day Delete: 50 per day	Create: 500 per day Delete: 500 per day	Create: 2,000 per day Read: 10,000 per day Update: 5,000 per day Delete: 1,000 per day

3 Use-case design

3.1 Architectural refinement

3.1.1 Identify design elements

3.1.1.1 Identify classes

Analysis class	Design element
Account	Account, Database subsystem
Advertisement	Advertisement, Database subsystem
Report	Report, Database subsystem
CreateAccountController	Map directly to design classes
DeleteAdvertisementController	
DeleteUserController	
EditAdvertisementController	
LoginController	
ProcessReportsController	
PublishAdvertisementController	
ReportAdvertisementController	
SearchController	
UpdateAccountController	
ViewAdvertisementController	
CreateAccountForm	
DeleteAdvertisementForm	
DeleteUserForm	
EditAdvertisementForm	
LoginForm	
ProcessReportsForm	
PublishAdvertisementForm	
ReportAdvertisementForm	
SearchForm	
UpdateAccountForm	
ViewAdvertisementForm	

Table 3-1. Analysis-Class-To-Design-Element map

3.1.1.2 Identify subsystems and interfaces

The *Database* subsystem provides support for relational databases written in the SQL language. The subsystem is designed as follows:

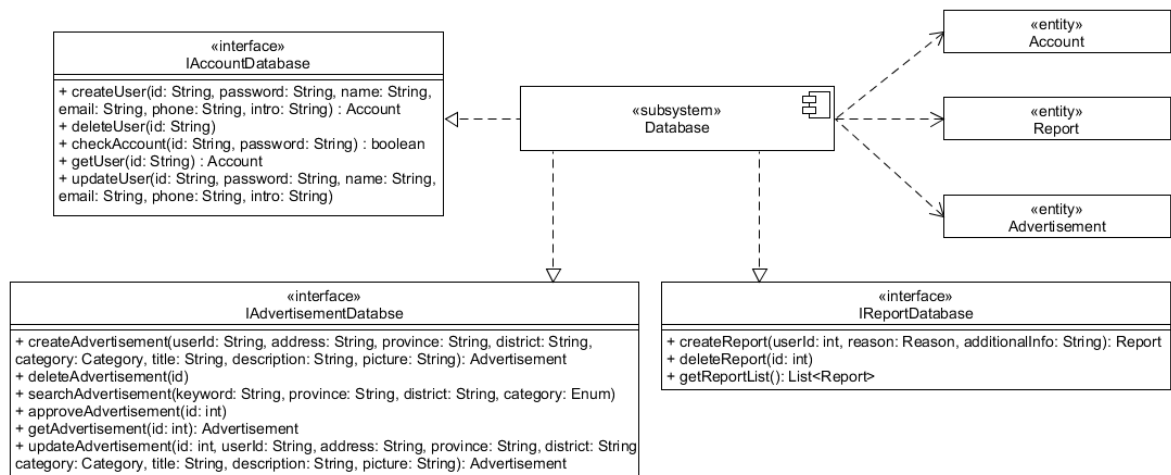


Figure 3-1. The Database subsystem and its interfaces

3.1.1.3 Identify packages

Each layer in the analysis corresponds to a high-level package in the system.

The Application package

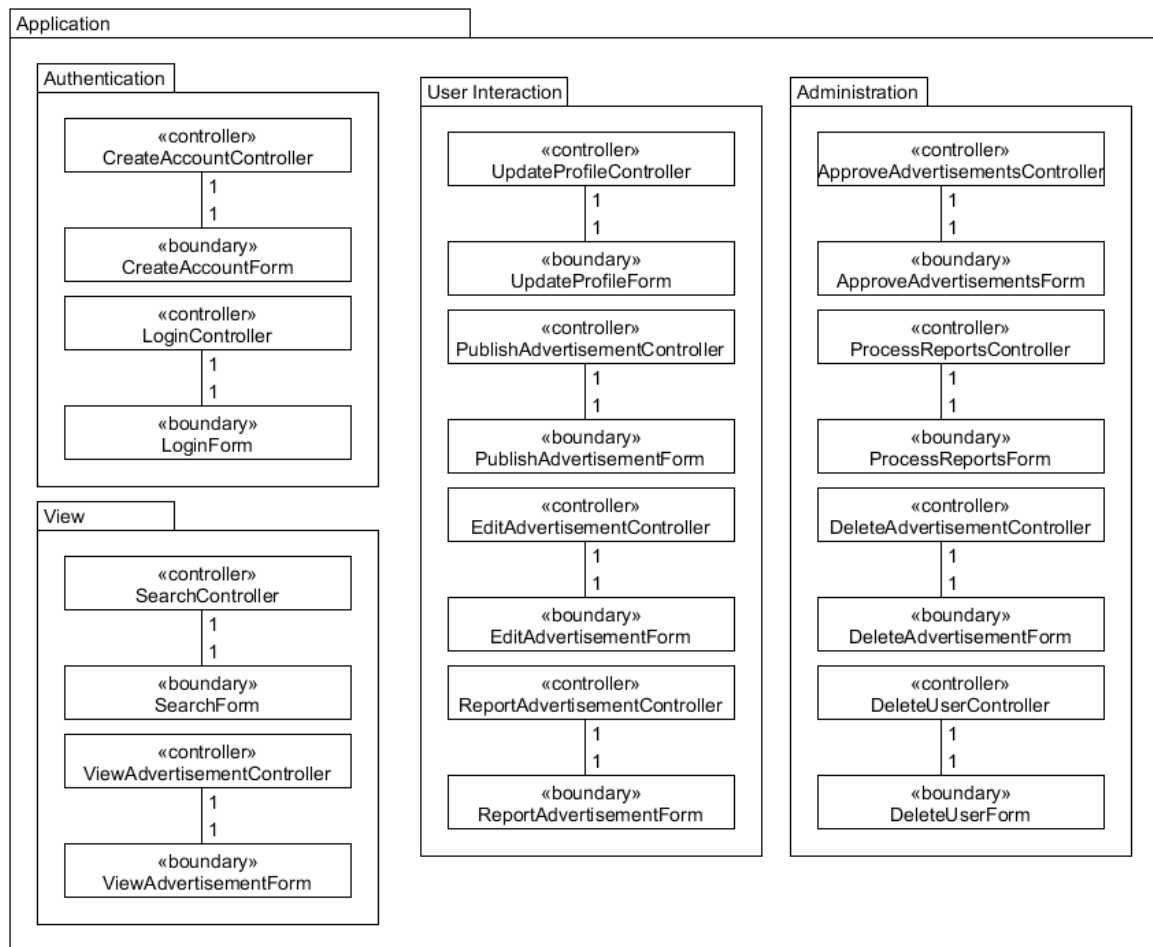


Figure 3-2. The Application package and its sub-packages

The *Application* package contains the boundary and control classes, which are present in the client application. It is further divided into four sub-packages, each responsible for a different part of the application:

- The *Authentication* sub-class handles account creation and login.
- The *View* sub-class is the realisation of two use cases related to the viewing and searching of advertisements. This part of the application can be freely accessed by anyone.
- The *User Interaction* sub-class contains classes involving actions which require the user to be logged in: updating profile and publishing/editing/reporting advertisements.
- The *Administration* sub-class contains utilities that help administrators maintain accounts, advertisements and reports. Only Administrators have access to these tools.

The *Business Services* package

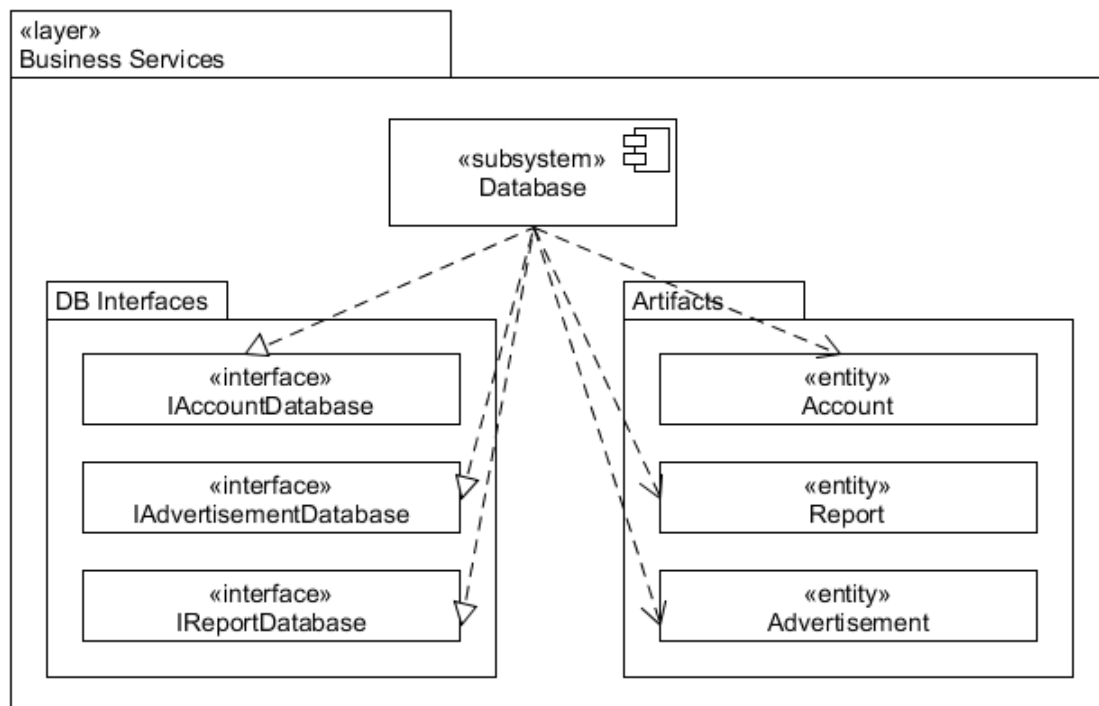


Figure 3-3. The Business Services package

The *Business Services* package contains the *Database* subsystem and its interfaces, as well as the entity classes. These elements are common to all use cases.

The *Middleware* package

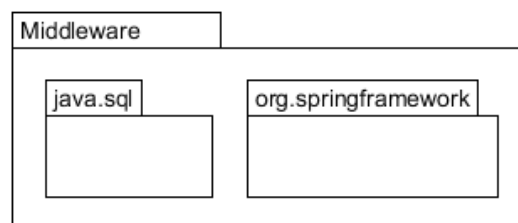


Figure 3-4. The Middleware package

The *Middleware* package includes Java’s SQL package, which provides access to databases and the Java Spring framework, which provides network services.

Packages and their dependencies

As already stated, the *Application* package depends on the *Business Services* package, which in turn depends on the *Middleware* package.

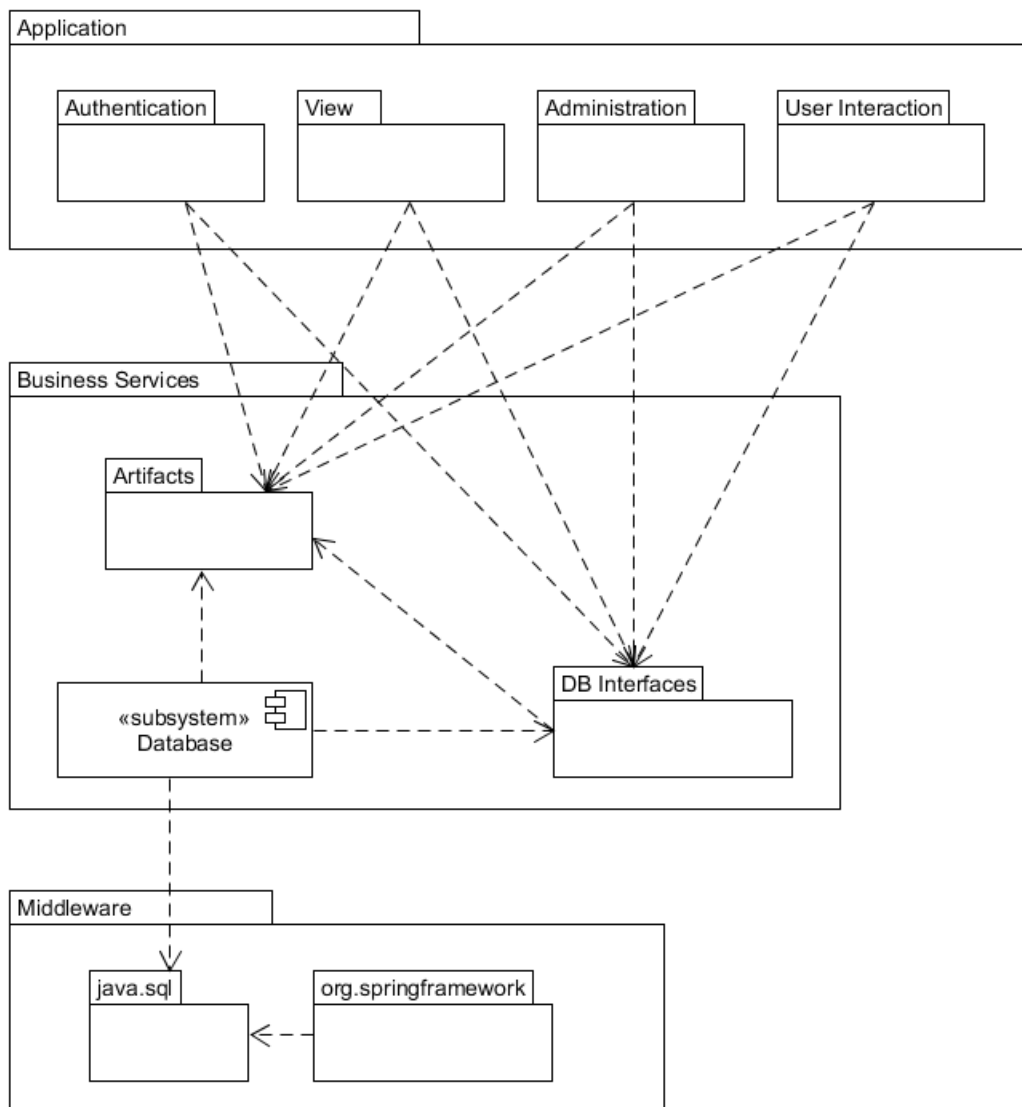


Figure 3-5. Package dependencies diagram

3.1.2 Identify design mechanisms

Analysis mechanism	Design mechanism	Implementation mechanism
Persistency	RDBMS	JDBC
Security	Web tokens	Java Spring framework
Distribution	REST API	Java Spring framework

Table 3-2. Design and implementation mechanisms

3.2 Describe the run-time architecture

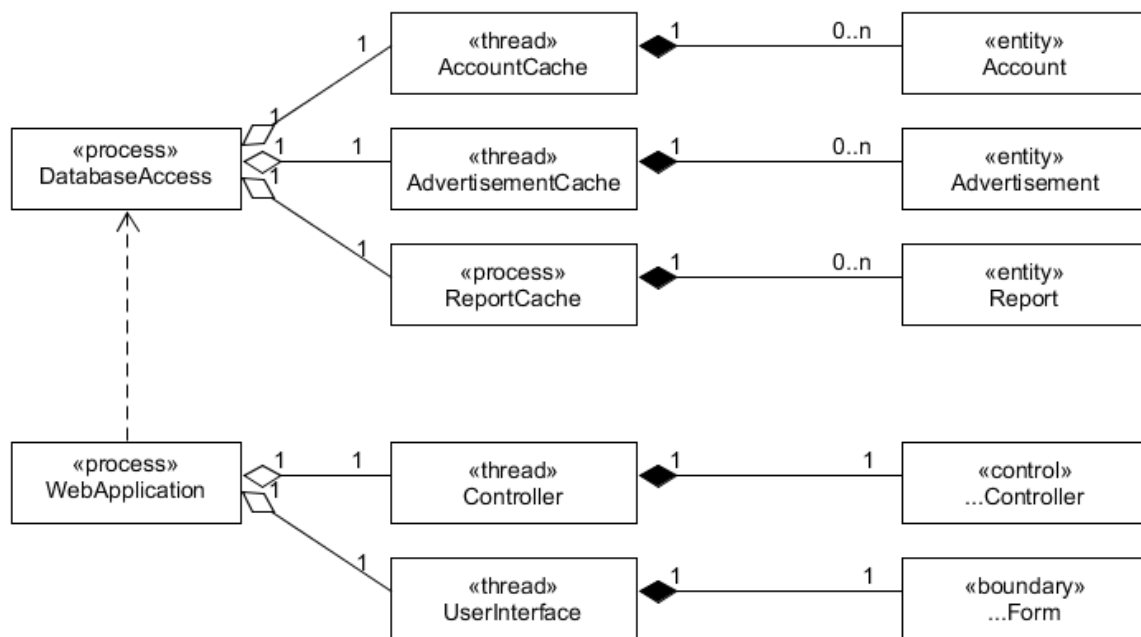


Figure 3-6. The system's process model

3.3 Describe distribution

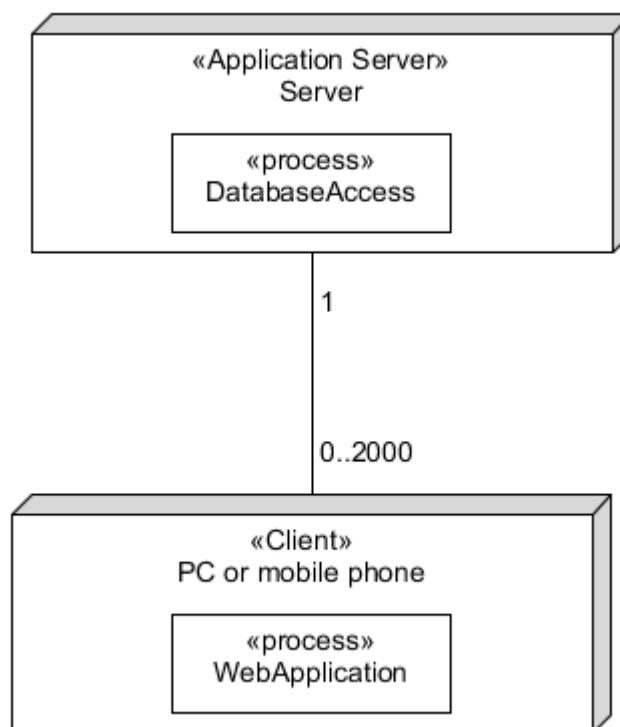


Figure 3-7. The deployment view of the system

3.4 Use-case design

3.4.1 Design sequence diagrams

After incorporating the *Database* subsystem, the model's sequence diagrams are updated as follows. Some method parameters are elided for conciseness and legibility – they are shown in full in the *Class Design* section.

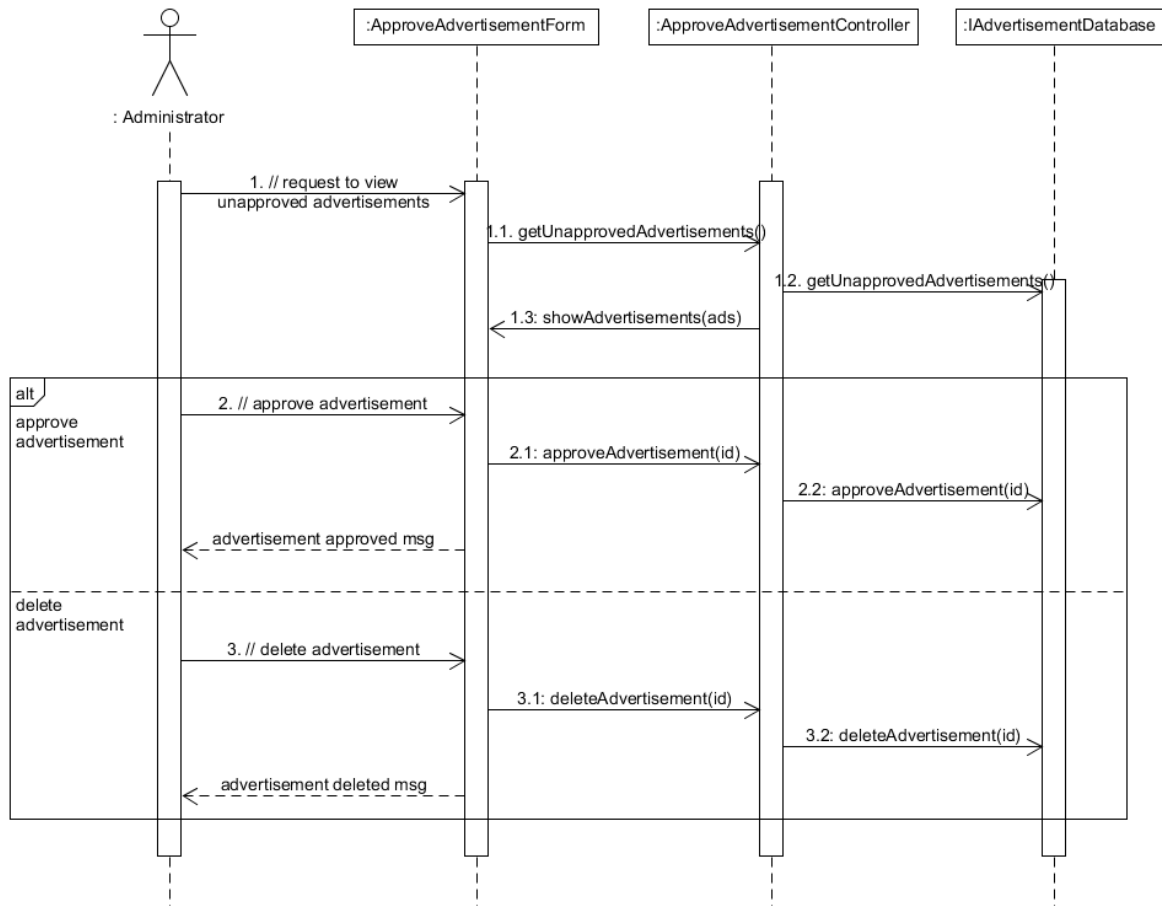


Figure 3-8. Design sequence diagram for the Approve Advertisements use case

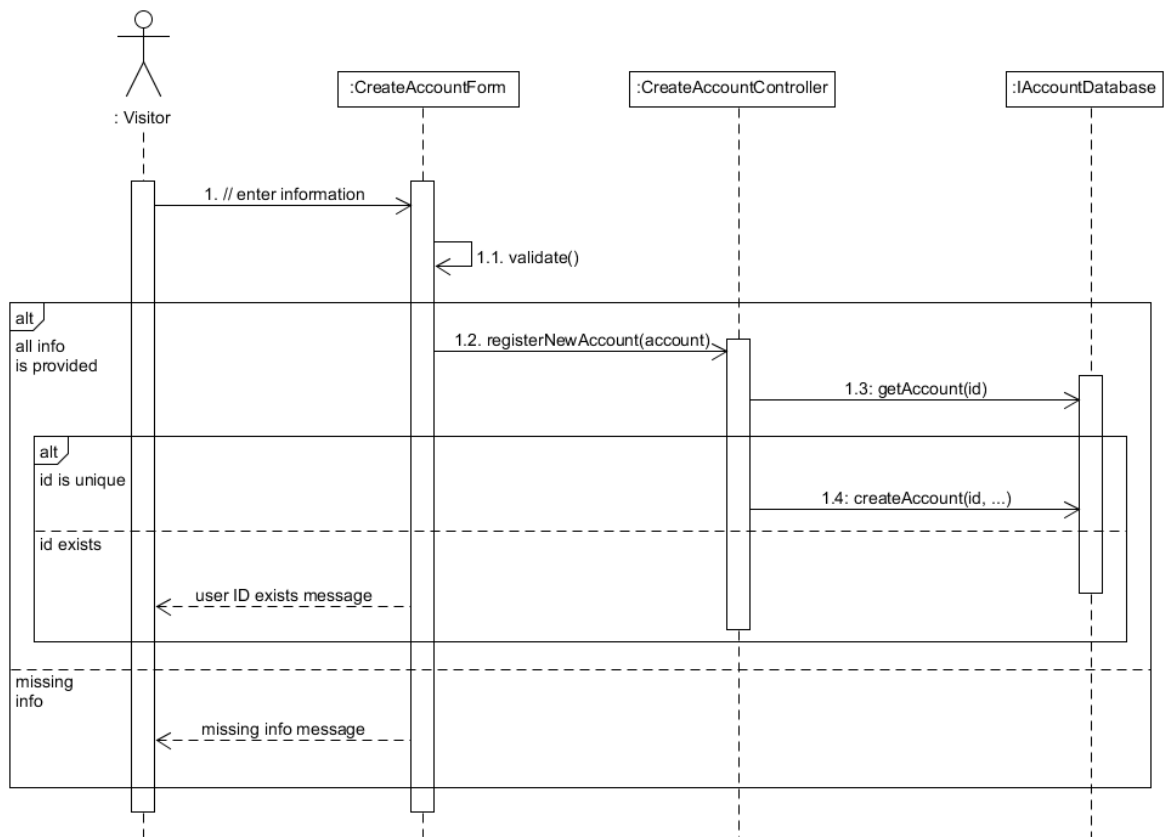


Figure 3-9. Design sequence diagram for the Create Account use case

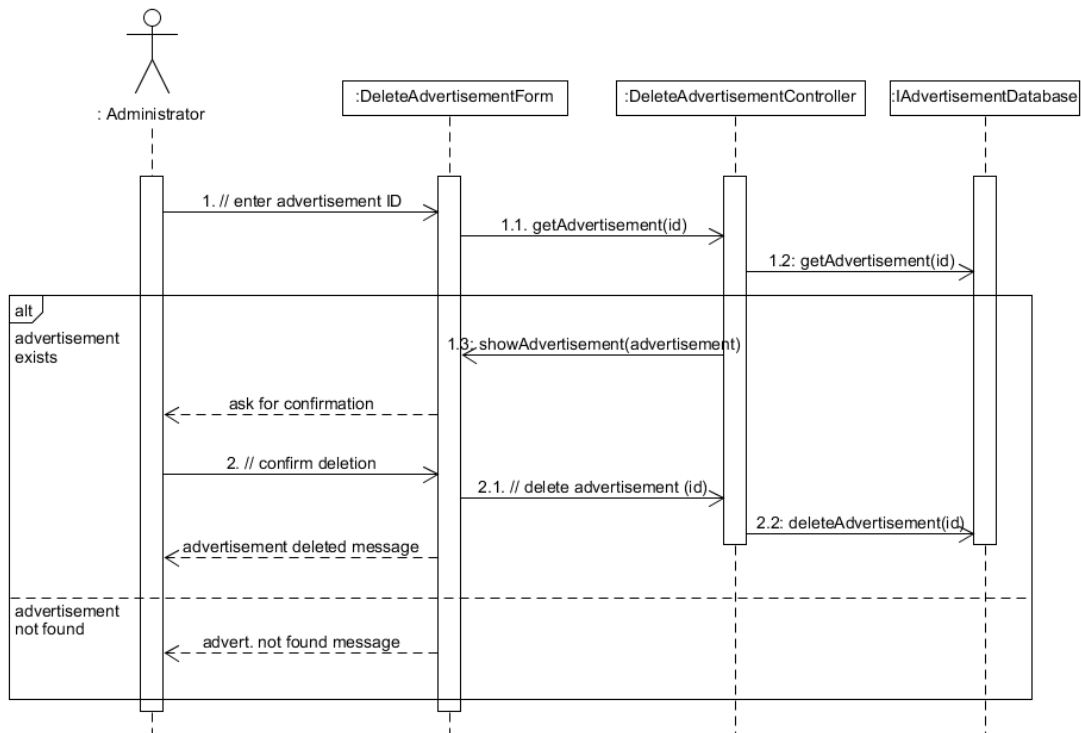


Figure 3-10. Design sequence diagram for the Delete Advertisement use case

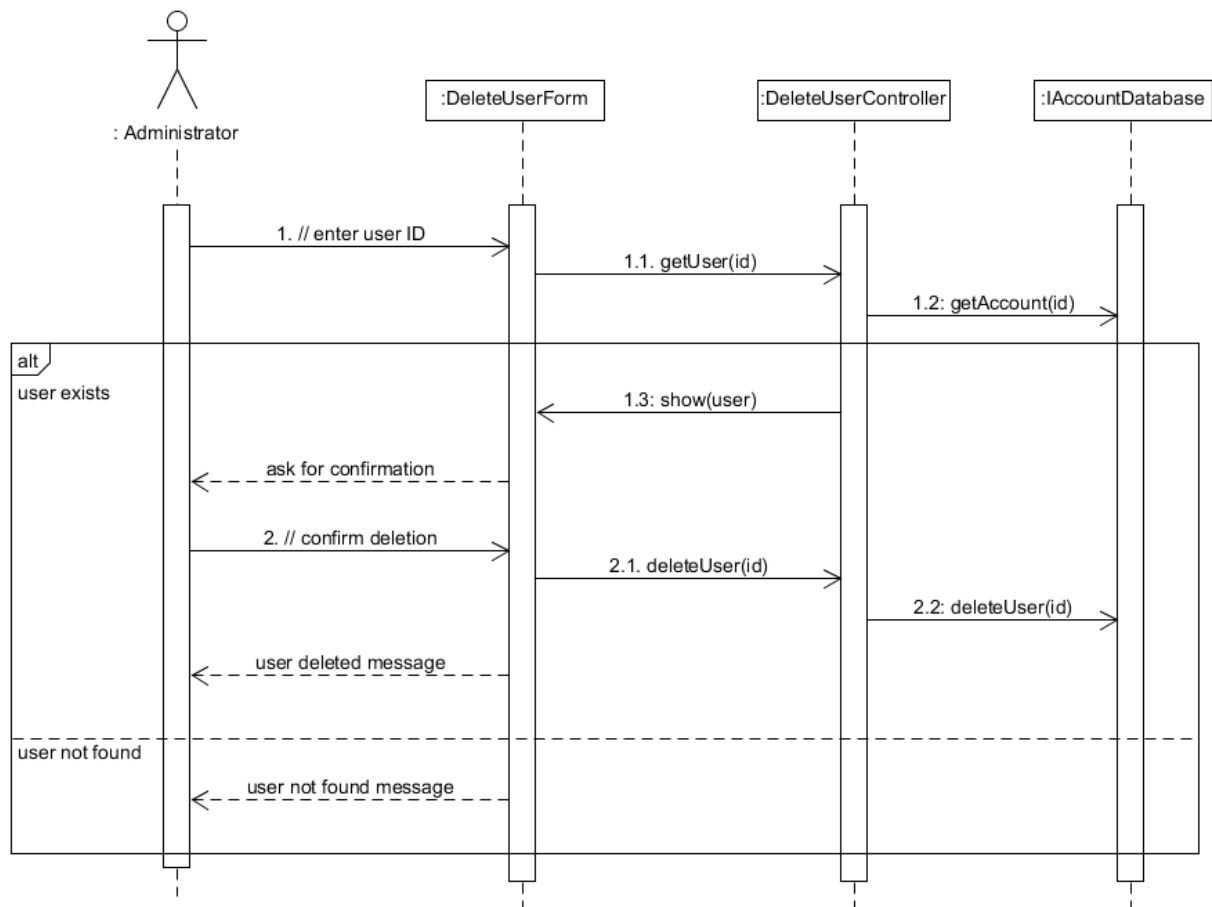


Figure 3-11. Design sequence diagram for the Delete User use case

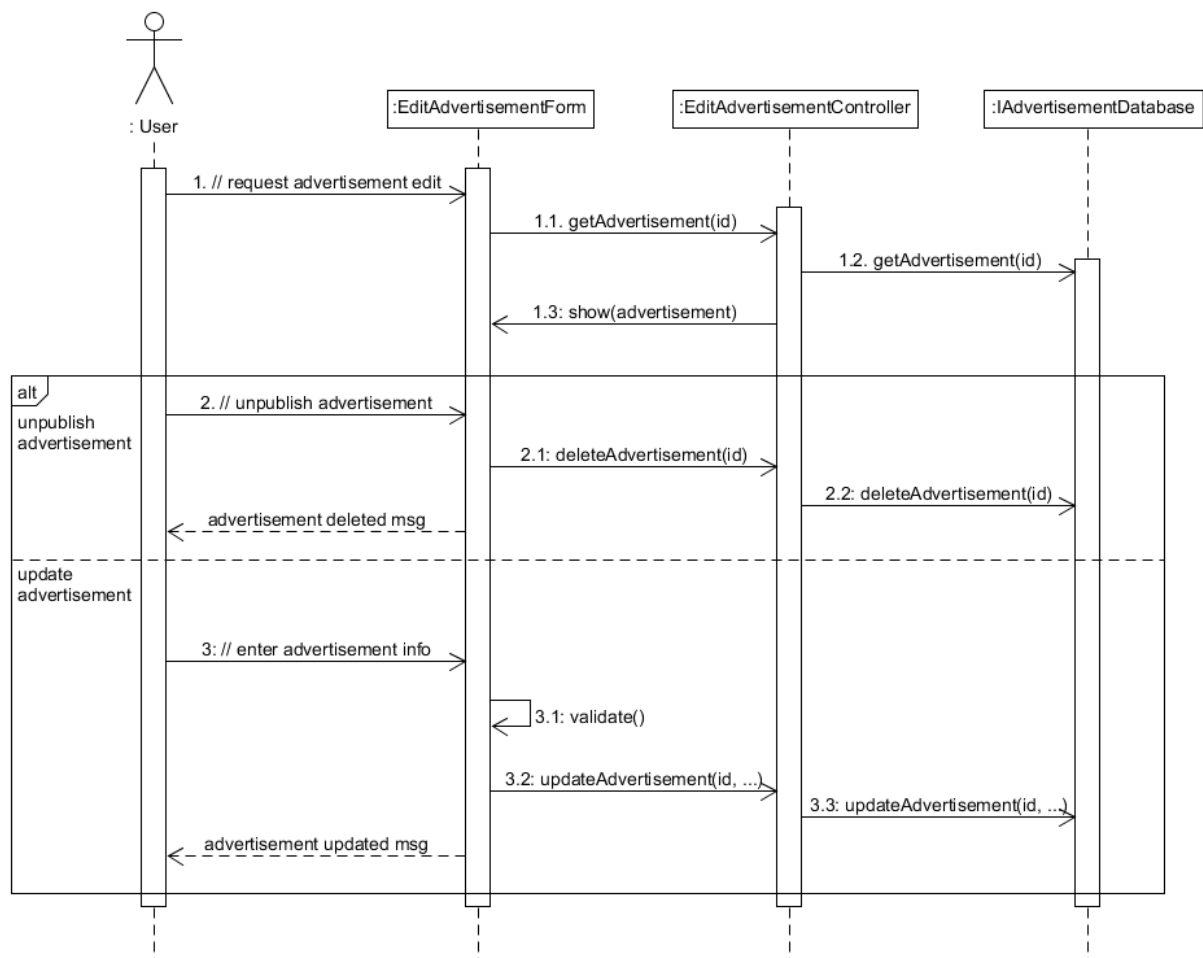


Figure 3-12. Design sequence diagram for the Edit Advertisement use case

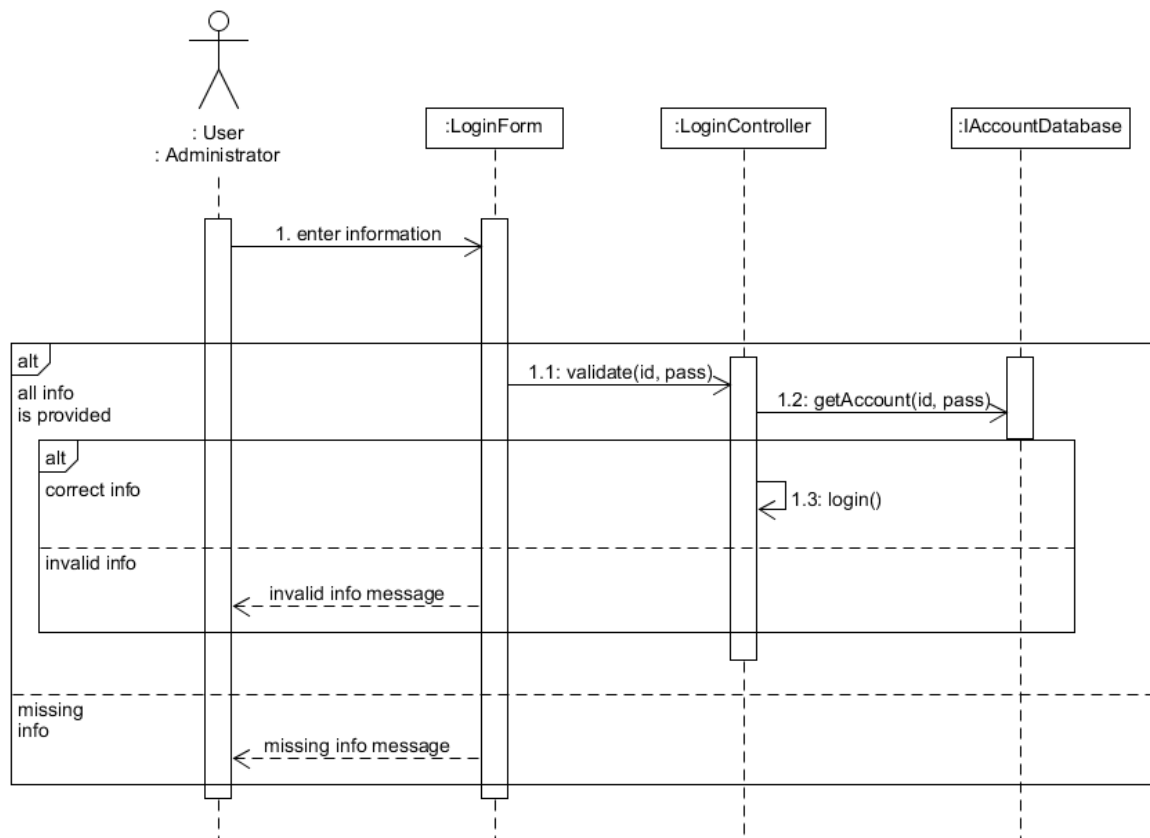


Figure 3-13. Design sequence diagram for the Log In use case

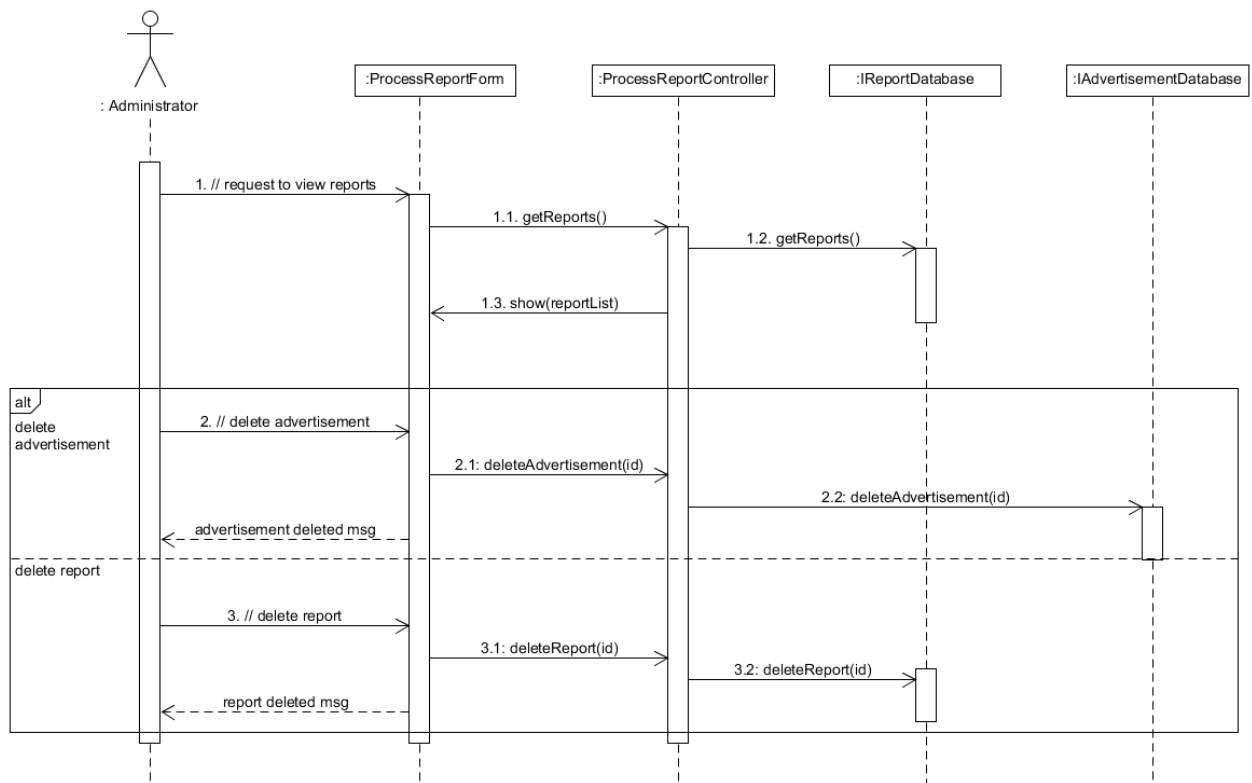


Figure 3-14. Design sequence diagram for the Process User Reports use case

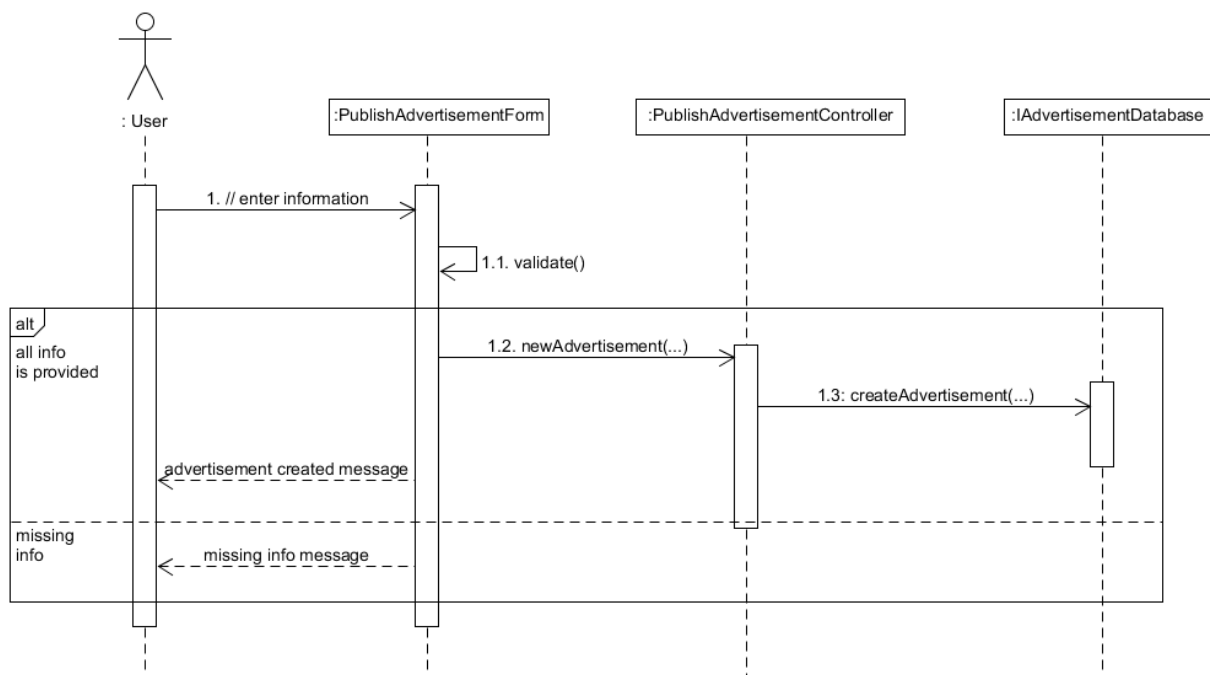


Figure 3-15. Design sequence diagram for the Publish Advertisement use case

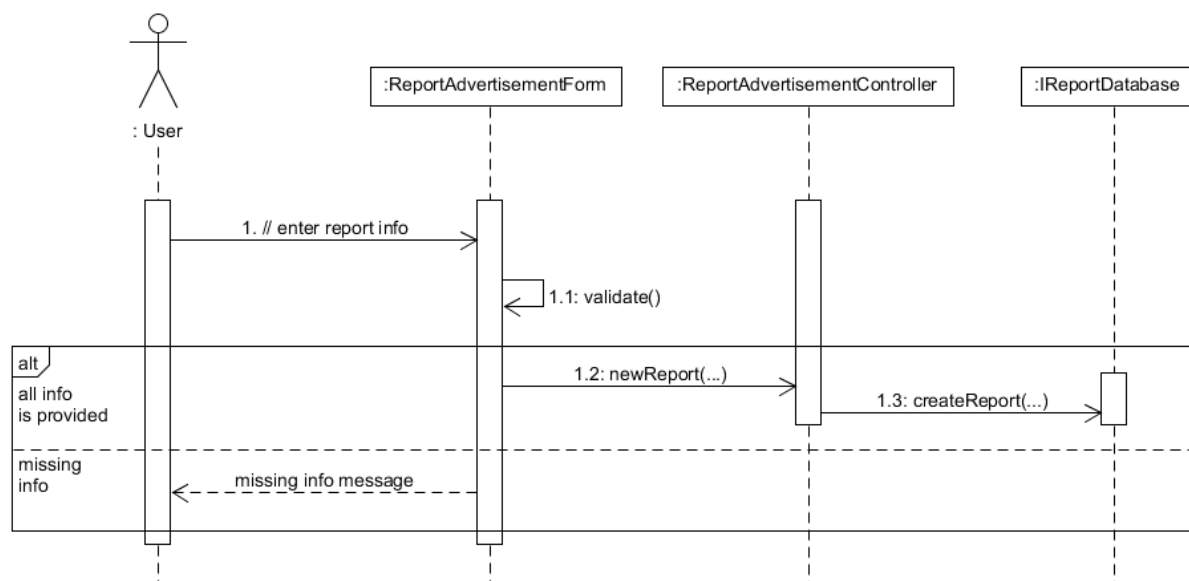


Figure 3-16. Design sequence diagram for the Report Advertisement use case

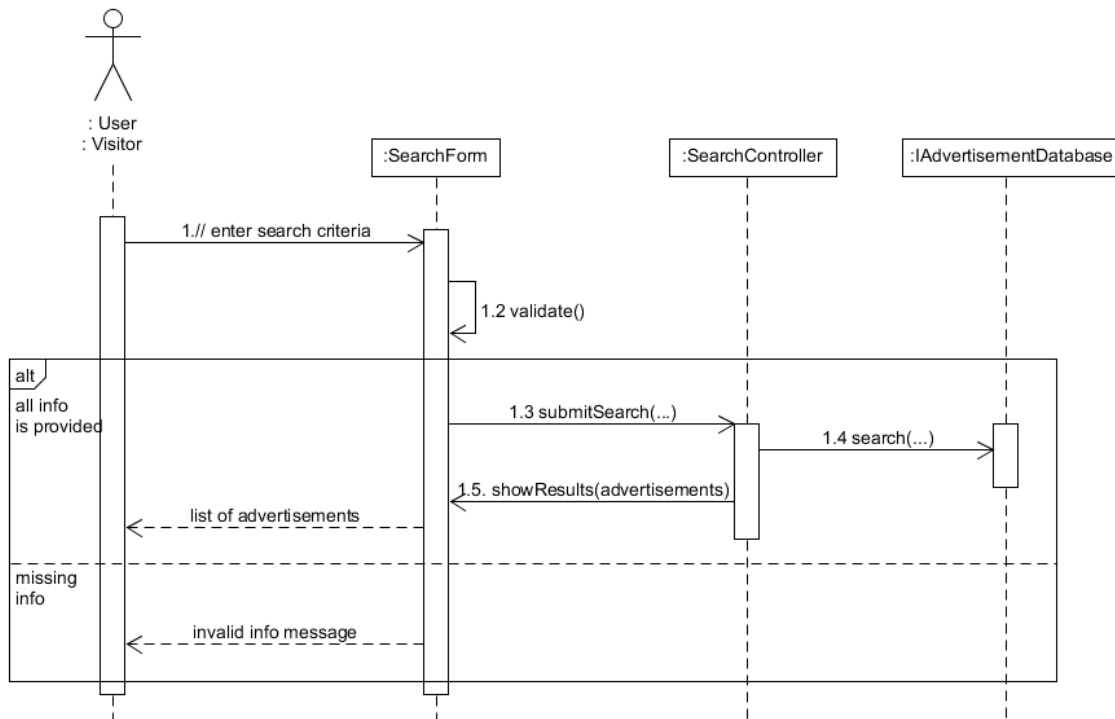


Figure 3-17. Design sequence diagram for the Search For Advertisements use case

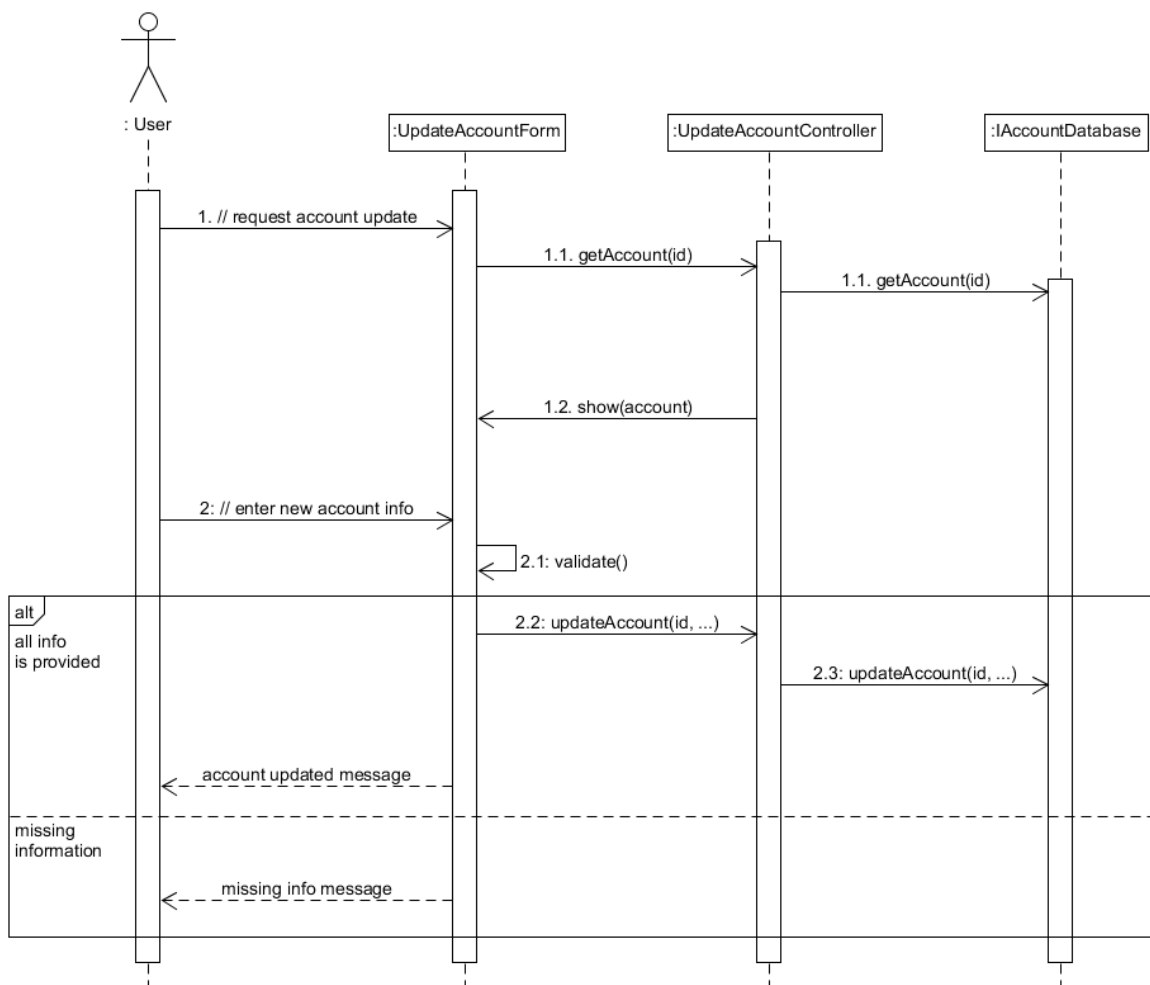


Figure 3-18. Design sequence diagram for the Update Account use case

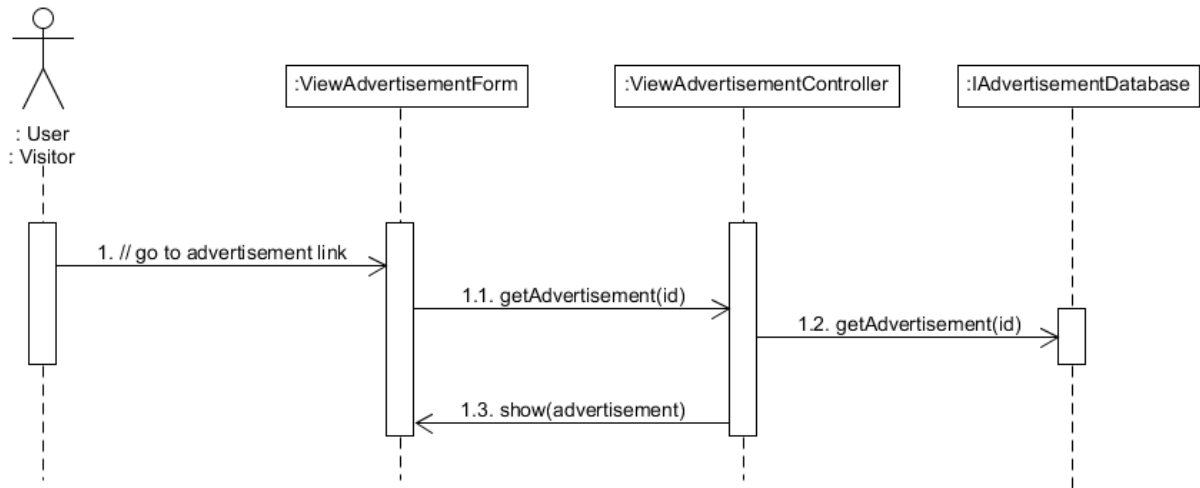


Figure 3-19. Design sequence diagram for the View Advertisement use case

3.4.2 Design views of participating classes

The updated views of participating classes for each use case are described in the *Class Design* section.

3.5 Subsystem design

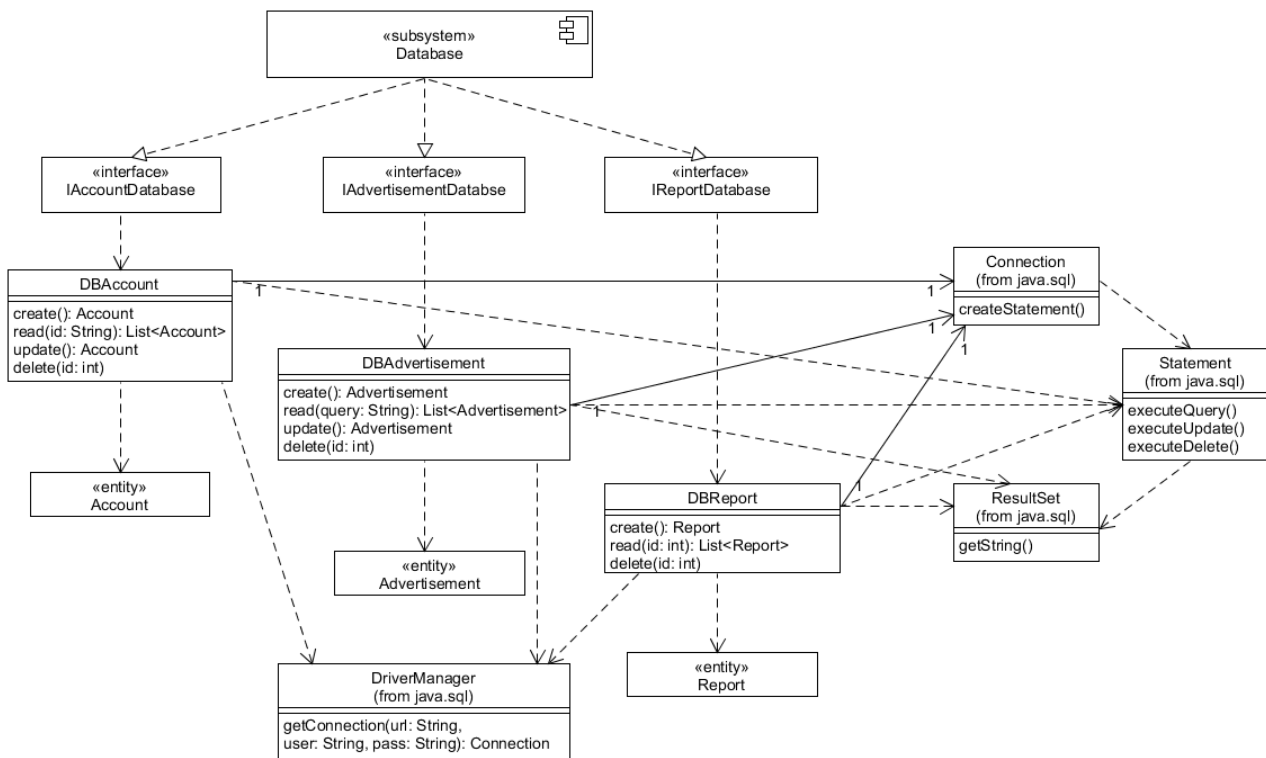


Figure 3-20. Database subsystem elements diagram

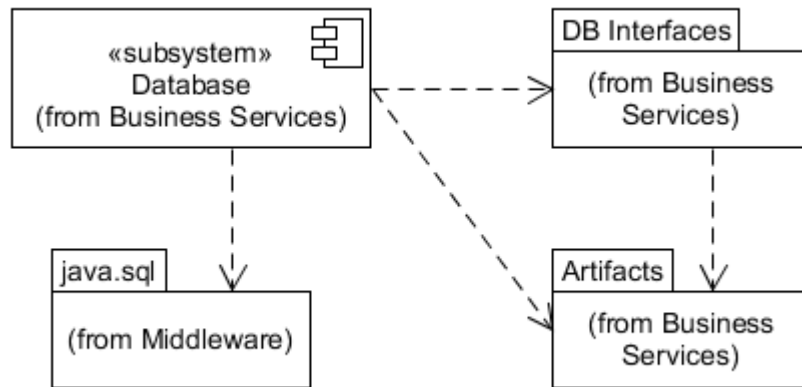


Figure 3-21. Subsystem dependencies class diagram

3.6 Class design

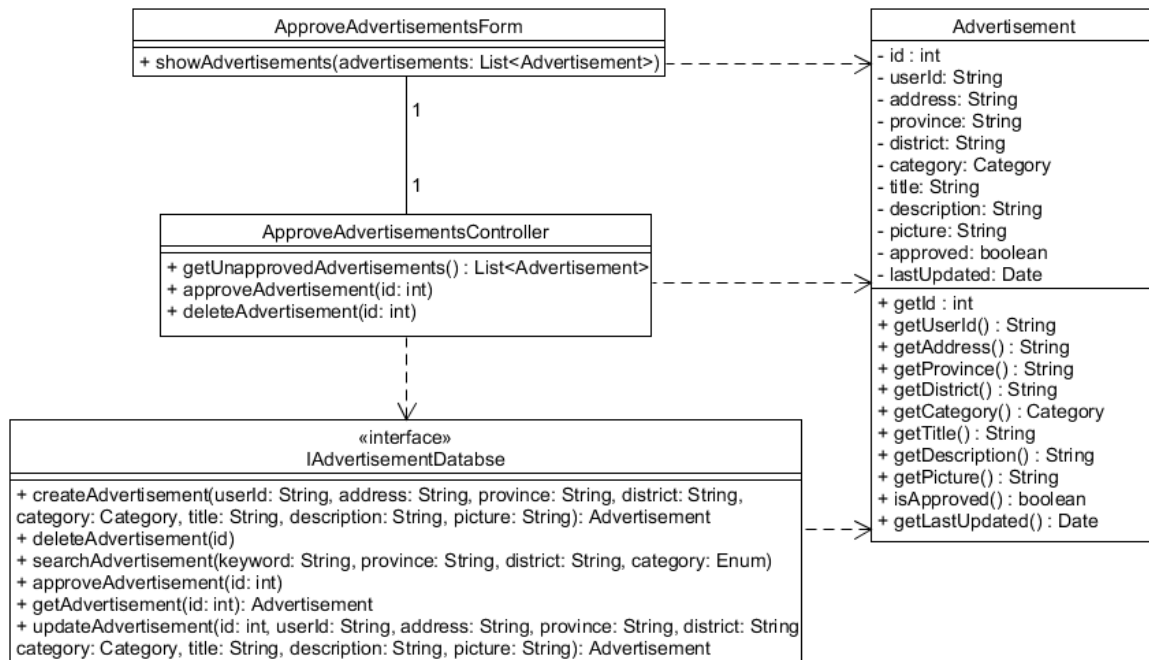


Figure 3-22. Design VOPC for the Approve Advertisements use case

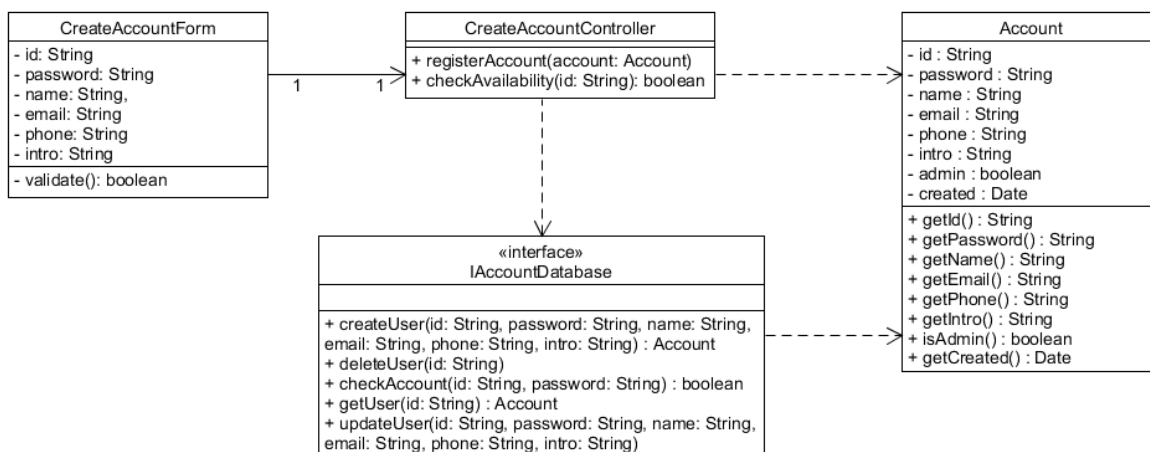


Figure 3-23. Design VOPC for the Create Account use case

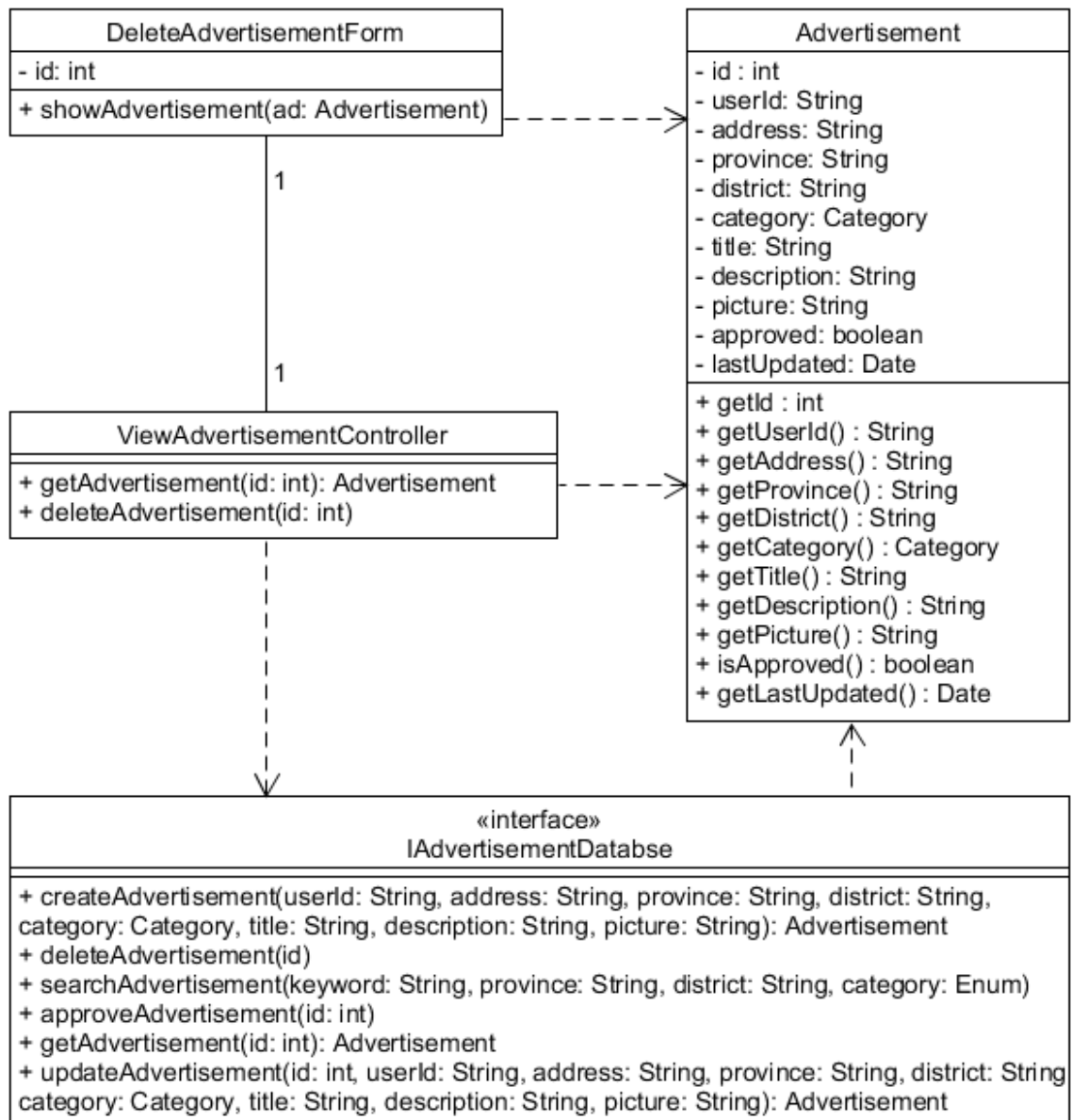


Figure 3-24. Design VOPC for the Delete Advertisement use case

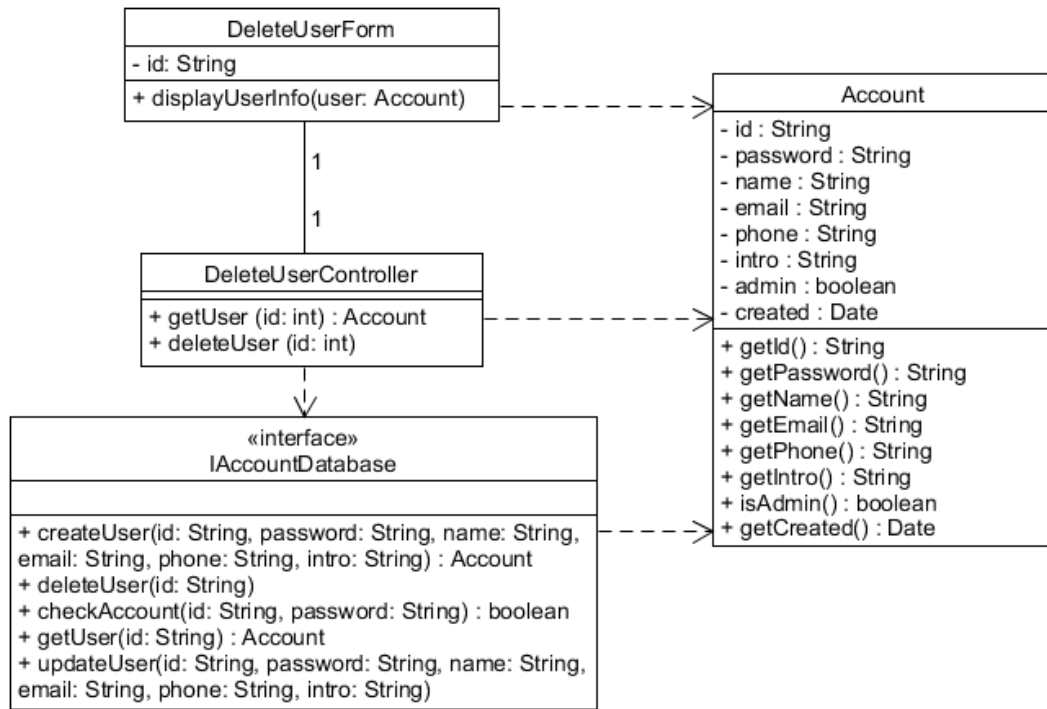


Figure 3-25. Design VOPC for the Delete User use case

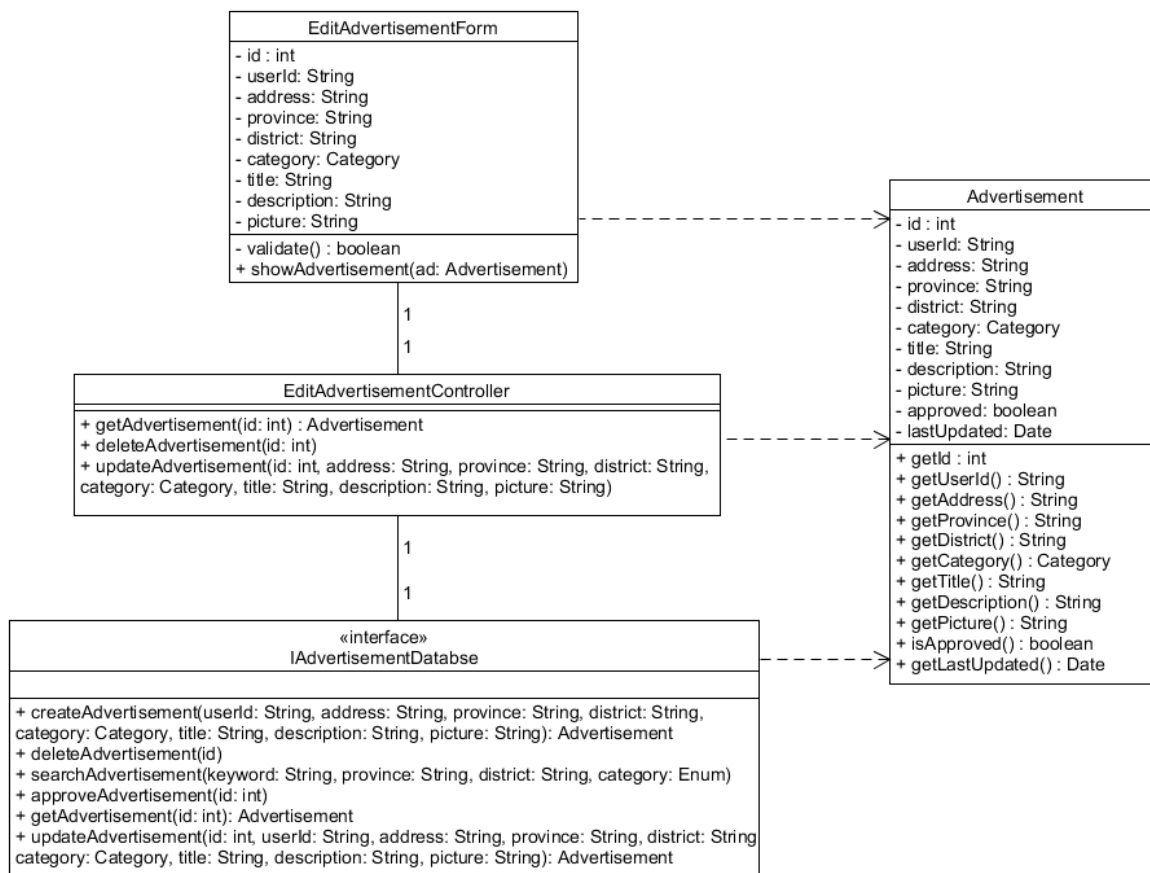


Figure 3-26. Design VOPC for the Edit Advertisement use case

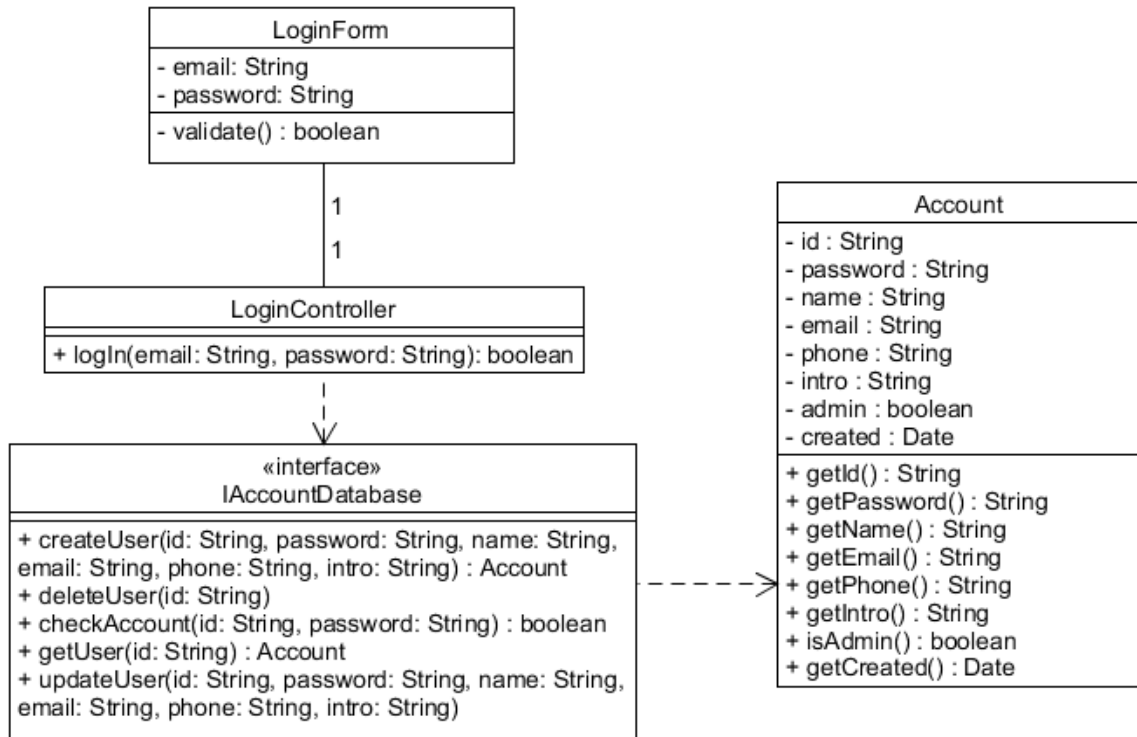


Figure 3-27. Design VOPC for the Log In use case

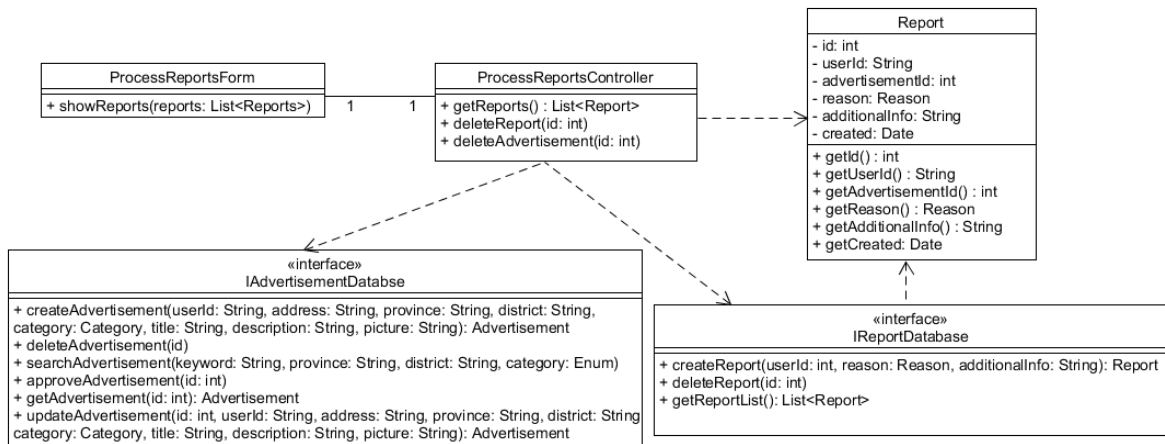


Figure 3-28. Design VOPC for the Process User Reports use case

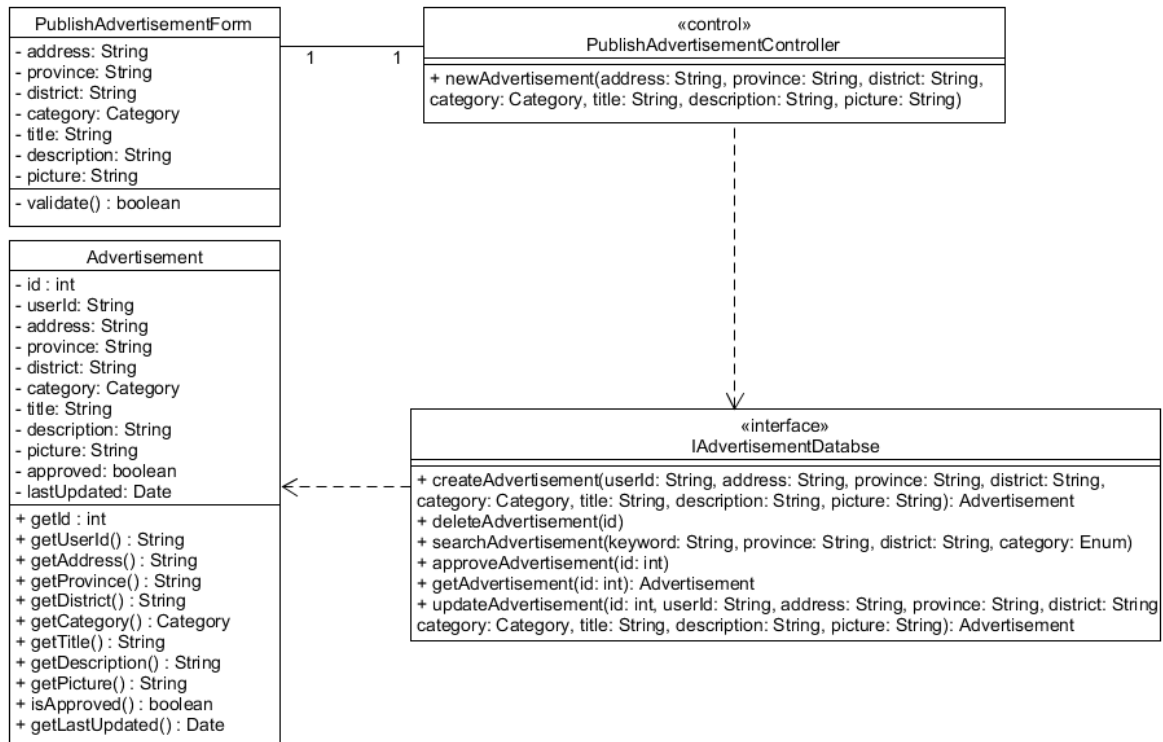


Figure 3-29. Design VOPC for the Publish Advertisement use case

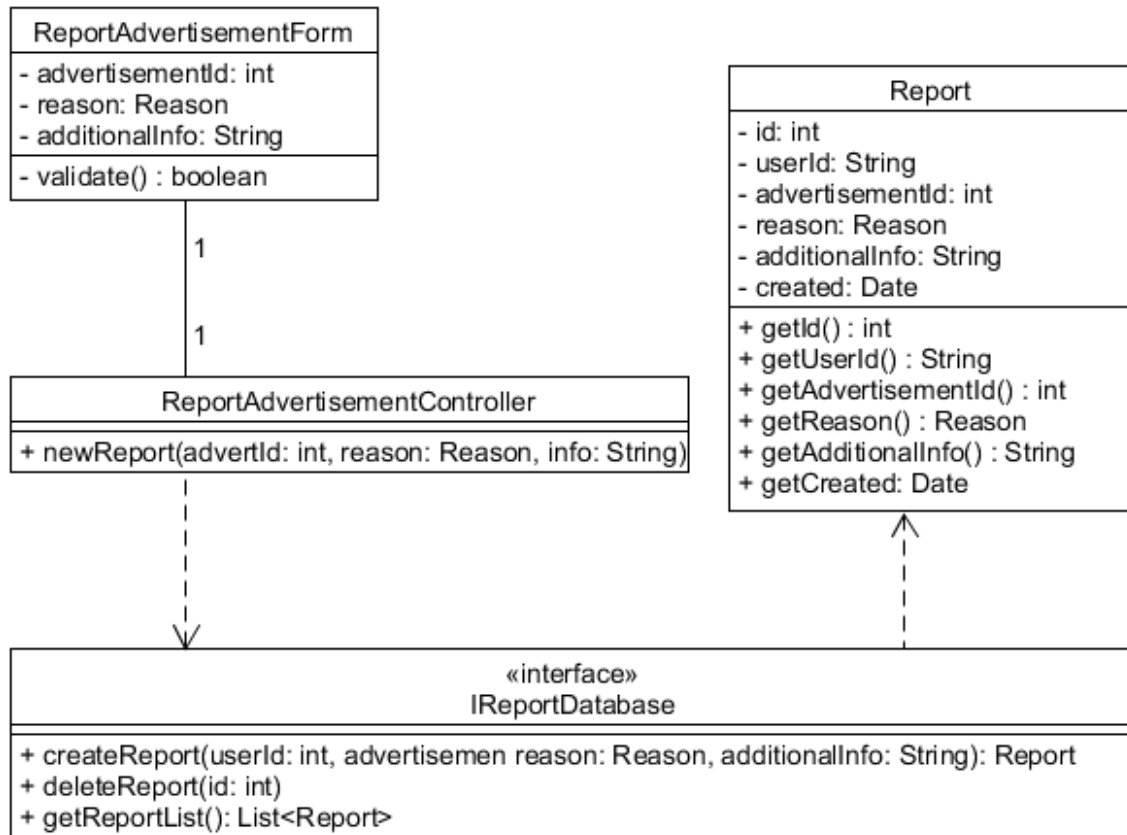


Figure 3-30. Design VOPC for the Report Advertisement use case

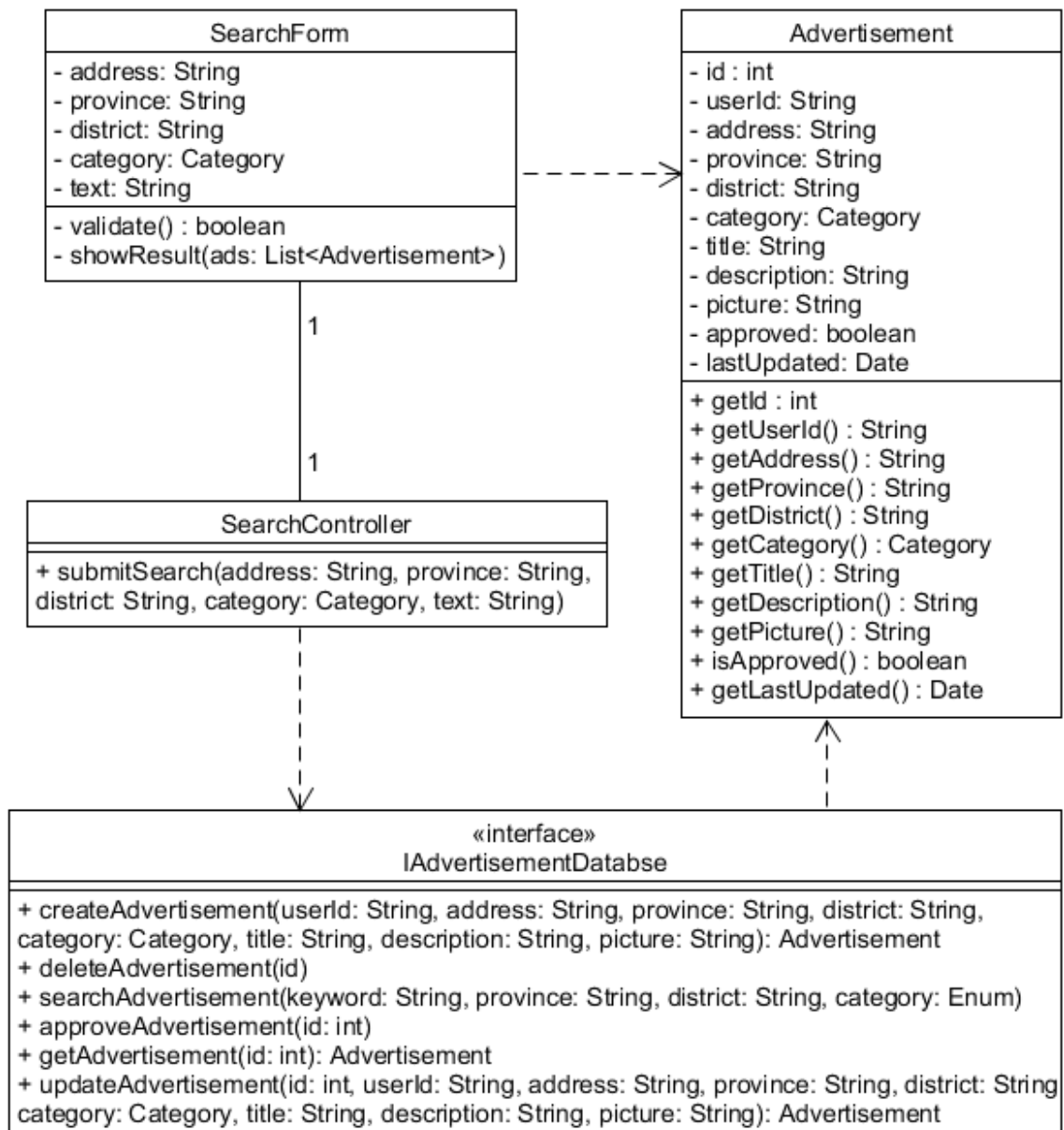


Figure 3-31. Design VOPC for the Search For Advertisements use case

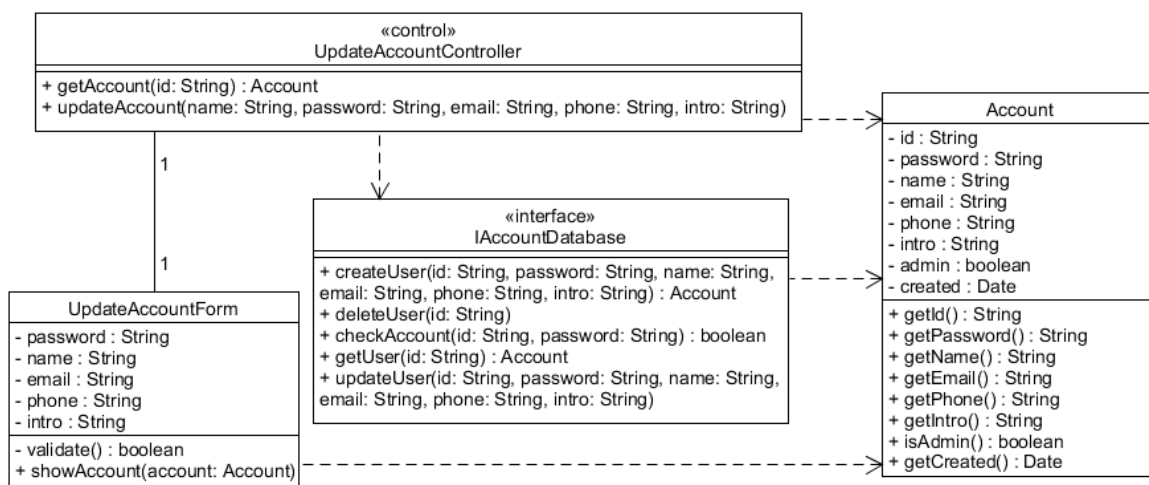


Figure 3-32. Design VOPC for the Update Account use case

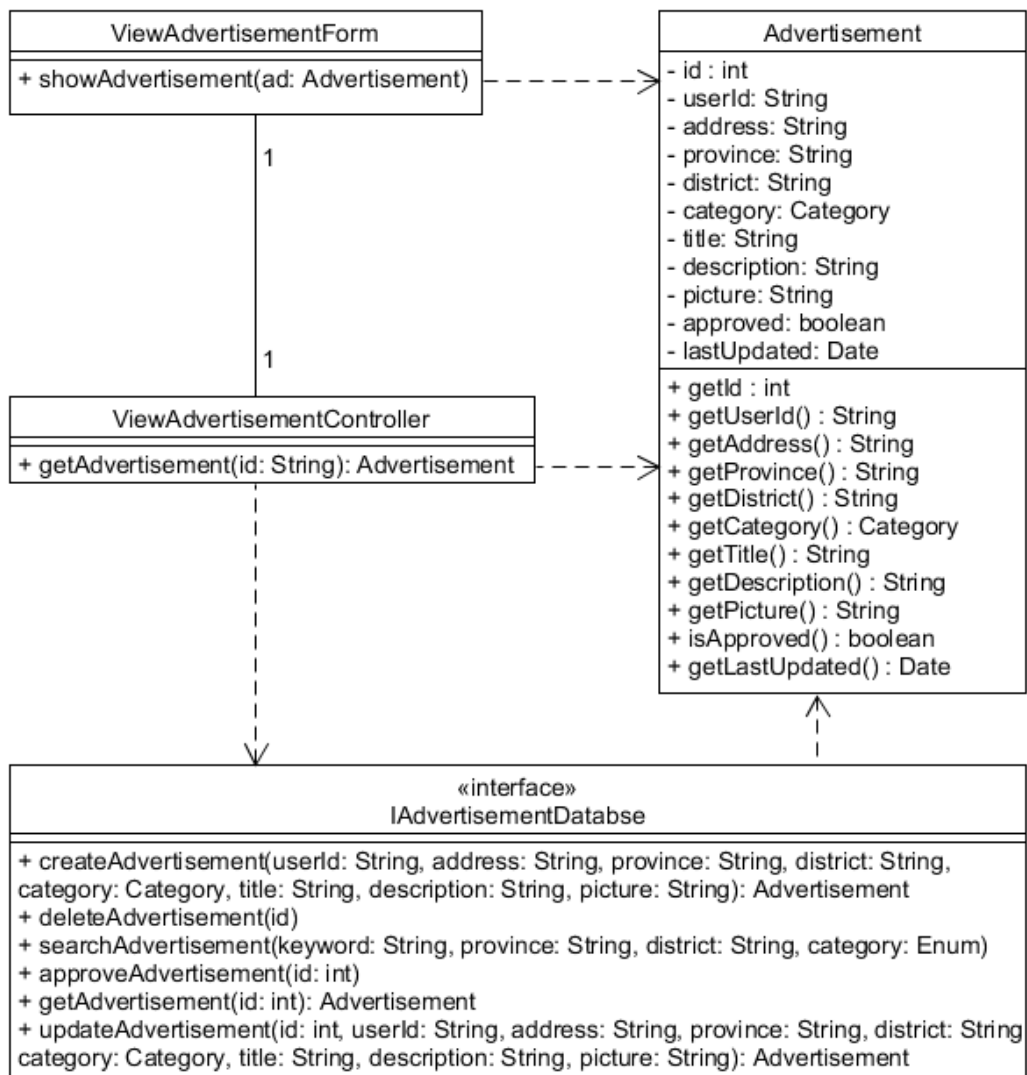


Figure 3-33. Design VOPC for the View Advertisement use case

3.7 Database design

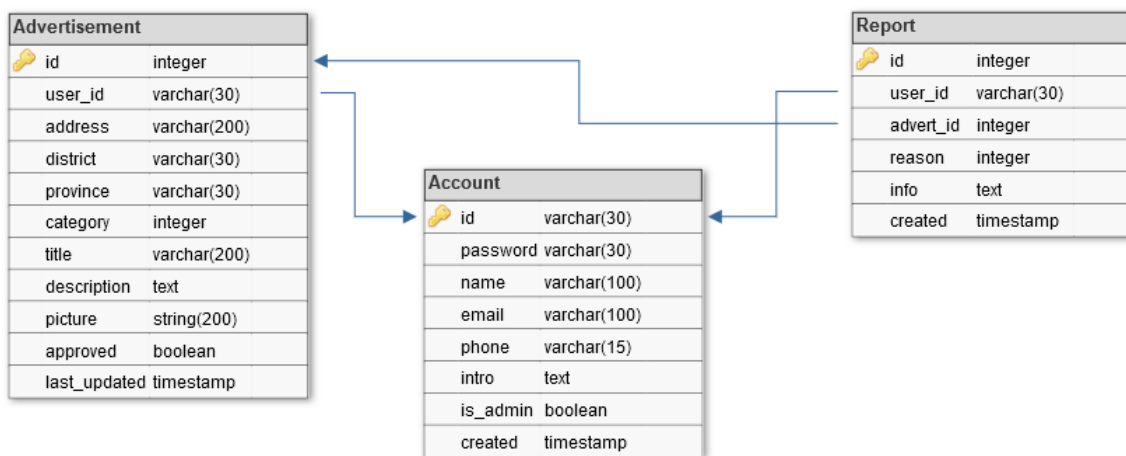


Figure 3-34. The relational data model