Journal of Software Engineering

ISSN 1819-4311

Academic Journals Inc.

www.academicjournals.com

OPEN ACCESS Journal of Software Engineering

ISSN 1819-4311
DOI: 10.3923/jse.2016.192.221

Research Article
A Systematic Approach for Reusing Web System Using UML-based Web Engineering

[1]Hamdy K. Elminir, [2]Alaa El-din M. Riad and [3]Ahmed M. El-Halawany

[1]Department of Electrical Engineering, Faculty of Engineering, Kafr Elshiekh University, Egypt
[2]Faculty of Computers and Information Sciences, Mansoura University, Egypt
[3]Sure International Technology, Cairo Branch, Egypt

Abstract
The importance of web system reuse has been increased with the revolution of the internet and building a reusable web system become a new challenge. Here, a systematic approach for building a reusable web system is presented. This approach contains two concerns; first is about web system architecture since it builds a reusable Model Driven Web Engineering (MDWE) through integrating component-based architecture approach with service oriented architecture approach. Second concern is this approach involves reusability in the early development phases regardless of the development methodology. This approach uses Unified Modeling Language (UML) and UML-based web engineering (UWE) to express MDWE artifacts in the analysis and design phases. A case tool was developed for modeling and storing analysis and design artifacts, associating these artifacts with its related proven tested implementation codes, also with its related test cases in order to maintain these reusable assets in a central library, this tool also helps to find out whether the acquired assets for building new MDWE are available or not.

Key words: Model-driven web engineering, software reuse, UML-based web engineering, web engineering, application domain

Received: November 29, 2015 Accepted: February 01, 2016 Published: March 15, 2016

Citation: Hamdy K. Elminir, Alaa El-din M. Riad and Ahmed M. El-Halawany, 2016. A systematic approach for reusing web system using UML-based web engineering. J. Software Eng., 10: 192-221.

Corresponding Author: Hamdy K. Elminir, Department of Electrical Engineering, Faculty of Engineering, Kafr Elshiekh University, Egypt

Competing Interest: The authors have declared that no competing interest exists.

Data Availability: All relevant data are within the paper and its supporting information files.

INTRODUCTION

Model Driven Development (MDD) is a software development approach that addresses a model in each development process with a high level of abstraction, it depends on the separation of concern principals and presents semi-automation methodology for generating implementation code (Moreno *et al*., 2008). Model Driven Architecture (MDA) is a software design approach that presented by Object Management Group (OMG) and has been proven success in building many application domains, it uses a platform-independent modeling language to express a model in each development process (Moreno *et al*., 2008). Model Driven Web Engineering (MDWE) is a web application model driven approach, in which MDD and MDA are adapted to model web application domain (Moreno *et al*., 2008). The MDWE models different web application concerns such as analysis, content, navigation and presentation concerns. Each one of these concerns have separated model, all of these models are integrated in order to be transformed into an implementation codes (Koch *et al*., 2008).

Software reuse is the process of using pre-defined software components to build a new software system (Das, 2012). Developing web system is a complex and time consuming task, also reusing web system is more complex due to the navigation, interactivity and presentation of web application. Web system reuse has two fold: (1) Technical reuse includes code reuse, component reuse and object or function reuse and (2) None technical reuse includes analysis documents reuse, modeling artifacts reuse, test cases reuse and management plan reuse (Sommerville, 2011). Software reuse plays an important role in the software development market due to its promising benefits, such as increasing software productivity and quality which leads to a powerful competitive advantages, besides cutting down the development costs through minimizing development staff and the time consumed in the development tasks, in addition to reducing process risk through using proven tested components (De Almeida *et al*., 2005).

There are several approaches for reusing software system (Keswani *et al*., 2014; Gerard *et al*., 2007; Mascena *et al*., 2006). Software product line approach is a software reuse approach in which a set of systems and components are integrated together to solve a specific problem domain and present the final software product (Keswani *et al*., 2014). Another approach is design patterns, which are software reusable solutions for commonly occurring problems. Code generation is a software reuse approach that uses a set of tools to generate software code depending on a set of pre-defined problem domain or entity relationship diagram. Content Management Systems (CMS) became one of the most commonly used approaches which uses open source software to build web system from pre-defined libraries and components in less time and cost (Mascena *et al*., 2006).

Most of the recent software reuse approaches are not flexible enough to meet the needs of the various industrial situations (Das, 2012). They lack the formal documentation for reusable assets which lead to an increase in maintenance and modification costs and the process of integrating with third party component became complex as a result. Keswani *et al*. (2014) considered that there are 60% additional cost for making something reusable so that, most software house companies are believe that software reuse is a separated task that consumes more time and cost to build new reusable assets, also they does not relate software reuse with the development methodology they are used (Mascena *et al*., 2006). Software reuse approaches does not integrates technical reuse with none technical reuse as they are reused technical assets like objects, code, component and functions separately from none technical assets like analysis documents, modeling artifacts, management plan and risk analysis besides reuse test cases is not investigated. Code generation approach is limited to a small number of business domains or is related to specific design pattern; it also generates unused code since generating tools depend on specific templates, also most of reuse approaches does not provide a way to find out whether the acquired components are available or not (De Almeida *et al*., 2005).

Navigation, interactivity and presentation reuse for a web system is still an emerging discipline, but recently code generation tools tries to generate navigation and presentation for small web system without formal documentation. There are many open source content management systems that help to build web systems, but they require trained personnel who know how to use it, besides learning the web development technology, also the requirements have to be adapted to reflect the functionality. It also lack of control over functionality and performance (Sommerville, 2011). The pre-defined libraries and components in the open source CMS makes it easy for attackers to hack it.

In this study, an approach was presented that helps to develop a reusable MDWE through integrating component-based architecture with service oriented architecture, the reusable MDWE was modeled using UML and

UWE, also a case tool was developed to model and store the modeling artifacts with its related implementation code and related test cases besides the ability to search for the stored reusable assets.

Several approaches were presented to tackle the challenges of software reuse range from reusing system objects and methods (technical reuse) to reuse the whole software environments (none technical reuse).

Mascena *et al*. (2006) presented an integrated reuse environment as a tool for helping organizations in achieving an efficient software development practice, this study integrate software reuse with information retrieval fields for allowing a uniform dissemination of the knowledge about available reusable objects.

Ahmed (2011) presented an approach for integrating the development environments into a case tool, this approach focused on reusing UML artifacts and the early stages artifacts. Robinson and Woo (2004) presented a simple approach to reuse software artifacts through an automated reuse case tool by helping analysts to reuse UML sequence diagram to retrieve similar and related artifacts for reuse.

Braga *et al*. (2006) presented a new approach for access and storage to domain component information through integrating a set of components into a specific application domain.

Mascena *et al*. (2006) presented medium scale example for software reuse through using an open source application server to build Content Management System (CMS) and showing that packaging software helps to improve the potential for software reuse.

Beyer *et al*. (2008) showed that a systematic reuse is beneficial than *ad hoc* reuse through applying an architecture-centric development with small team-three engineers only and the result was reduces the development and testing by more than 35% without compromising the overall quality of the system. Moreno *et al*. (2008) presented a new approach based on model driven architecture to build Rich Internet Application (RIAs), this approach includes four major development phases and it extends a UML profile for modeling web system, also develop a tool for generating code templates.

Rubin and Chechik (2012) presented an approach refactoring existing, closely related products into product line representations, this approach based on comparing and matching the related artifacts and merging them into a product line engineering adoption.

Rouille *et al*. (2013) presented a tool supported approach that integrates reuse software process line and the automation of their execution, this approach enables to automate the execution of processes containing unresolved variability, by enabling to resolve this variability during execution.

MODEL DRIVEN WEB ENGINEERING REUSE APPROACH

Model Driven Web Engineering Reuse (MDWER) approach is a web system reuse approach which uses the separation of concerns and MDD principals to tackles the different web application domain concerns (analysis, content, navigation, interactivity and presentation) during the development process. This approach integrates component-based architecture approach with service oriented architecture approach to build a reusable Model Driven Web Engineering (MDWE), this model is consists of a set of integrated components that managed by a set of integrated services. Services are integrated with each other or components through a contract layer, this contract responsible for interacting the service with its related components or other services; on the other hand, component-based contains a connectors that shows how this component interact with the related services or components.

The MDWER uses UML and UML-based Web Engineering (UWE) to express the modeling artifacts for MDWE, it also allows vertically and horizontally reuse for web application domain, vertical reuse helps to build new reusable MDWE; horizontal reuse helps to build new reusable components or services.

Vertical reuse shown in Fig. 1, application domain (A) is consists of a set of services (service A1, A2, A3 and A4) and a set of components (component A1, A2, A3 and A4) these services and components are integrated with each other's to form the application domain A. Application domain (B) and application domain (c) are consists a set of services and components integrated with each other's like application domain (A). Application domain (D) reuse a set of components and services from application domain (A) not all services or components (service A1, A2 and A3) and components (component A1 and A2); in application domain (D) reuse part of application domain (A) not the whole model as shown in Fig. 1. Application domain (E) reuse a set of services (service B1 and B3) from application domain (B) and reuse a set of components (component B1, B2, B3, B4, C1 and C3) from application domain (B) and application domain (C).

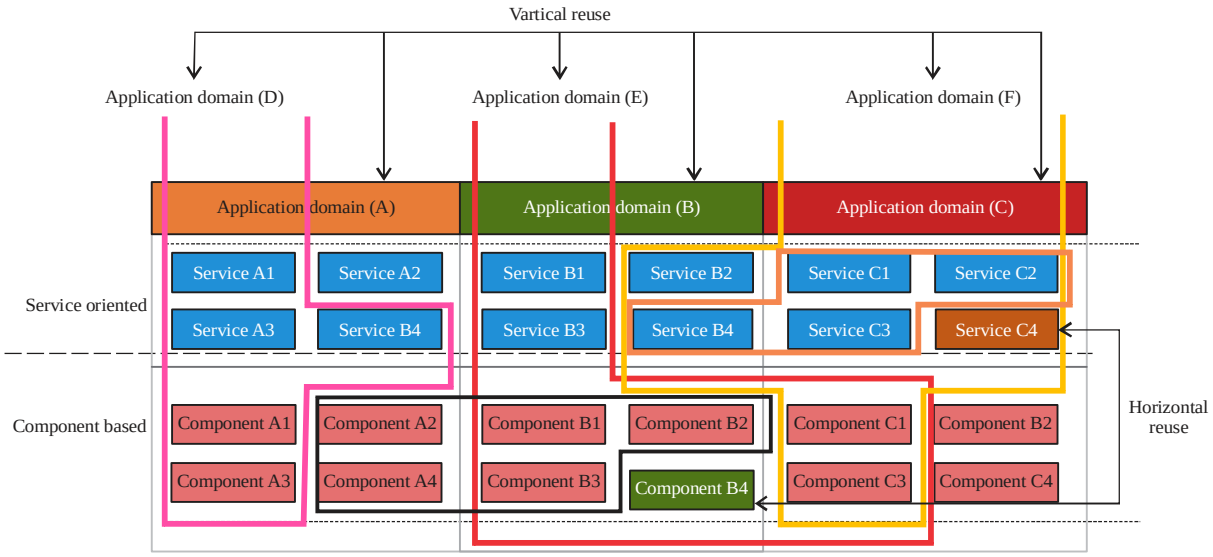

Fig. 1: Model driven web engineering reuse approach

Application domain (F) consists of a set of services (service B2, B4, C1, C2, C3 and C4) from application domain (B) and application domain (C) and a set of components (component C1 and C3) from application domain (F). Horizontal reuse shown in Fig. 1 as follows; Component "Component B4" is reused from components (component A2, A4, B1, B2 and B3); also service "Service C4" is reused from services (service B4, C1, C2 and C1).

This approach does not support auto-generation of implementation code to avoid two problems of code generation approach, they are the generation for unused code and using a specific design patterns. The MDWER enables developer to build their own specific web application domains and reuse it as needed through allowing developers to convert the MDWE modeling artifacts into implementation codes depending on their code standards and using their own development methodology.

A case tool was developed which represent a software repository or central library, this case tool does not generating code; it only stores analysis and design documents with related modeling artifacts, related proven tested implementation codes and related test case (reusable assets); Furthermore, it allows participators to: (1) Search in the existing web application domains, (2) Extract service or component from stored web application domain, (3) Document any changes or submission to the repository, (4) Evaluate these assets and (5) Extend the existing web application domain to build new domain so that this tool helps to reuse the reused assets and presents a series generation of reusable assets (Mascena *et al*., 2006).

WEB SYSTEM DEVELOPMENT PHASES

Practice software reuse to build a web system must be start from the early development phases so that MDWER adds an early phase in the web system development process called "Initiation phase" thus MDWE development phases become five phases: The initiation phase in addition to the four commons (analysis, design, implementation and testing). The MDWER uses Standard UML model diagram to model the initiation phase, whereas, a modified version of UML-based web engineering framework is used to model the analysis and design phases with difference concerns (analysis, content, navigation and presentation) (Elminir *et al*., 2011).

The MDWER does not focus only on reusing web system analysis and design artifacts, but also relates design with implementation codes and test cases via a platform and programming language independent case tool.

The MDWER breaks web system into a set of domains as shown in Fig. 2, each domain contains a set of modules and each module contains one or more small independent business requirement. A module is a version of UML and UWE artifacts that model specific business requirements exists in a specific web application domain, it also uses package diagram to model business requirements at the initiation, analysis and design phases. A domain is a container for a group of modules

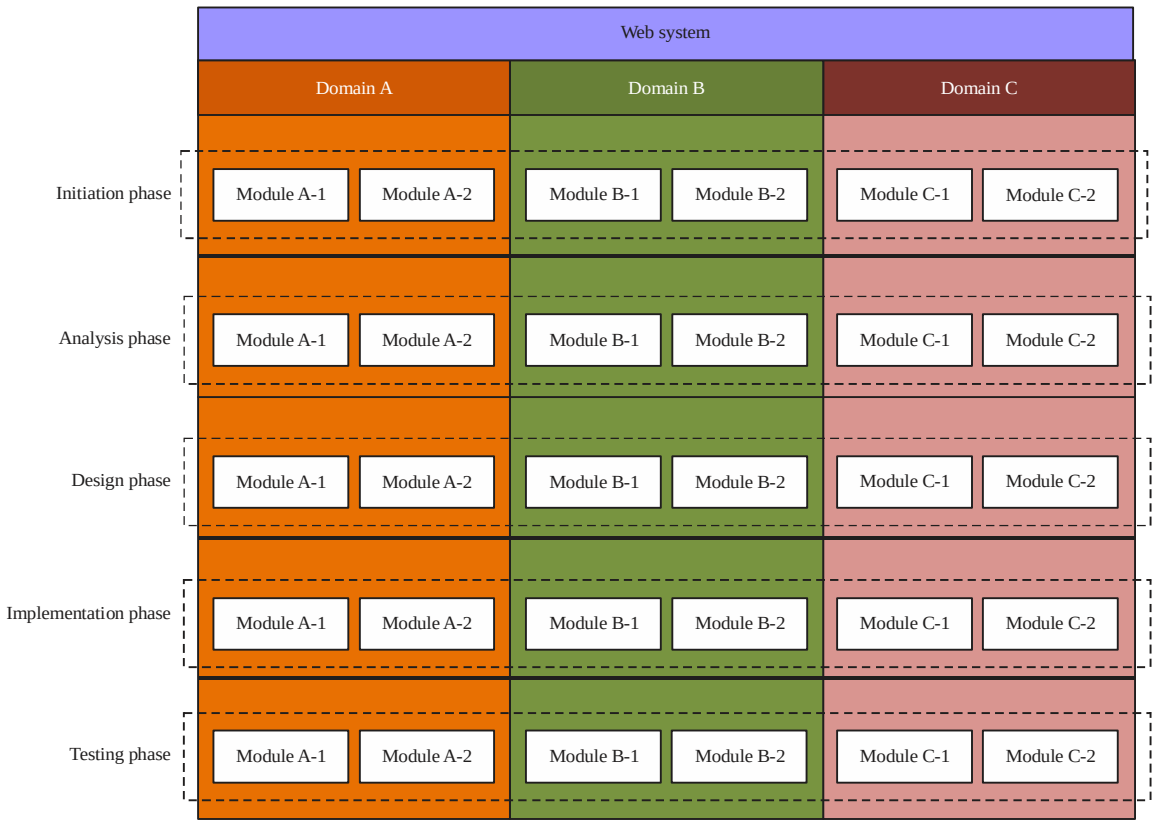

Fig. 2: Web system development process using MDWER

that defines a specific web application domain; a domain also helps to maintain the relations among the initiation, analysis, design and testing phases.

This study breaks web system vertically to help in application domain reuse and horizontally to help in component reuse. So, any application domain can be developed by reusing all the phases (initiation, analysis, design, implementation and testing) previously developed or by modifying the existing assets to match new web system requirements and if a domain or a module is related to another one, then both of their assets must be reused.

Initiation phase: The MDWER involves reusability in the early development phases through starting the development process with the initiation phase which responsible for determining which assets will be developed for reuse, which will develop with reuse and which will not be reused any more. In this phase, according to MDWER, web system divided into a set of business domain with their related modules using standard UML model diagram. Domains and modules are categorized into three types:

- Reusable assets which were not developed yet, it will be developed to be used in the current web system and to be reused again in similar systems as needed
- Existing assets which already exists and are used and tested before. It will be developed depending on a previously existing domain or module or the modular will modify it to match new business requirements needs
- Non reusable assets which is a highly customizable domain or module that is related to specific business needs, have not developed yet and will not be developed for reuse in the future

A list of steps was determined to evaluate software reuse: (1) State all domains for web system, (2) List modules related to each domain, (3) Determine the type for all domains and modules stated in the previous steps, (4) Describe the relation among modules and domains and (5) Check if a web system will use existing module or domain as it is or will modify it to make a new version.

Analysis phase: Knowledge reusing is considered one of the major factors in software assets reusing methodology as it represents the business analysis. In the analysis phase, MDWER uses UML-based web engineering framework (Elminir *et al*., 2011) to express the domains and modules which were determined in the previous initiation phase, this framework uses use case diagram to provide a rough description of web system functionality by defining three types of use cases: User process, browse and server process use case, also it uses a standard UML use case documents to provide more details description, these documents helps to extract test cases scenarios needed in the test phase. According to UML-based web engineering framework, activity diagram is used to provide more detailed description of the related use cases using four types of activities: Browse, client side, server side and outer side. The output of this phase is a set of UML-based web engineering use cases and activity diagrams (Elminir *et al*., 2011).

Design phase: The MDWER uses UML-based web engineering framework to model domains and modules which were analyzed in the analysis phase; the result is a set of artifacts that will be stored in the presenting case tool, this framework models three different web system concerns. They are: Conceptual model, navigation model and presentation model (Elminir *et al*., 2011). Conceptual model expressed in standard UML class diagram; navigation model expressed in a set of navigational classes, direct navigability, index, guided tour, query and menu stereotypes [27]; presentation model expressed in a set of stereotypes (presentation page, presentation group, presentation class and user interface elements) (Elminir *et al*., 2011).

Implementation phase: In this phase, according to MDWER, developers are translates the MDWE artifacts which were modeled using UML and UML-based web engineering in the previous phases into implementation codes using an object oriented programming language, their code standards and their suitable design pattern. Depicts to MDWER, the implementation code is stored in the developing case tool after verifying and fixing bugs related to this code.

Testing phase: Test cases are used to verify software business scenarios against implemented codes. They are extracted from use cases and their related activities and are organized in the same domains and modules of the use cases from which they were extracted. A single use case may be related to one test case or more, these test cases are reused when their related use cases or activities are used to develop with reuse.

We does not reuse all analysis and design artifacts, but we defines which assets in each phase can be reused as follows:

- Use case and activity diagram in the analysis phase
- Conceptual model, navigation space model and static presentation in the design phase
- Test cases in the testing phase
- Implemented codes in the implementation phase

WEB SYSTEM REUSE CASE STUDIES

The MDWER approach was applied on three different case studies with different business domains to show how this approach helps to increase productivity and quality through applying reusability as long as we builds a new web system.

News portals: This is a portal web site that displays categorized news (politic, sports, economic and technology). This portal has three actors, visitor can view all news lists and read news details, registered users can browse all the news and news comments and portal admin can manage news and news category. A daily email that contains the latest news is sent to registered users based on their favorite news category.

According to MDWER, the development process starts with the initiation phase. It breaks web portal into four domains (news, security, mail list and email service). The news domain contains news, comments, news comments categories and news category modules. Initiation phase determines which domain or module will develop for reuse, which will develop with reuse and which will not reused any more. In this case study, we do not have any modules or domains that were modeled since we begin to apply this study, so the all domains and modules will develop for reuse. For each domain or module that will develop for reuse start to model its analysis phase using UWE as shown in Fig. 3. Activity diagrams are modeled as needed to describe a specific use case; here four activities are shown in Fig. 4-7.

In the design phase, the conceptual model contains entity object model, extended object model, data provider model and rendering model; they are modeled using standard UML class diagram as shown in Fig. 8-11.

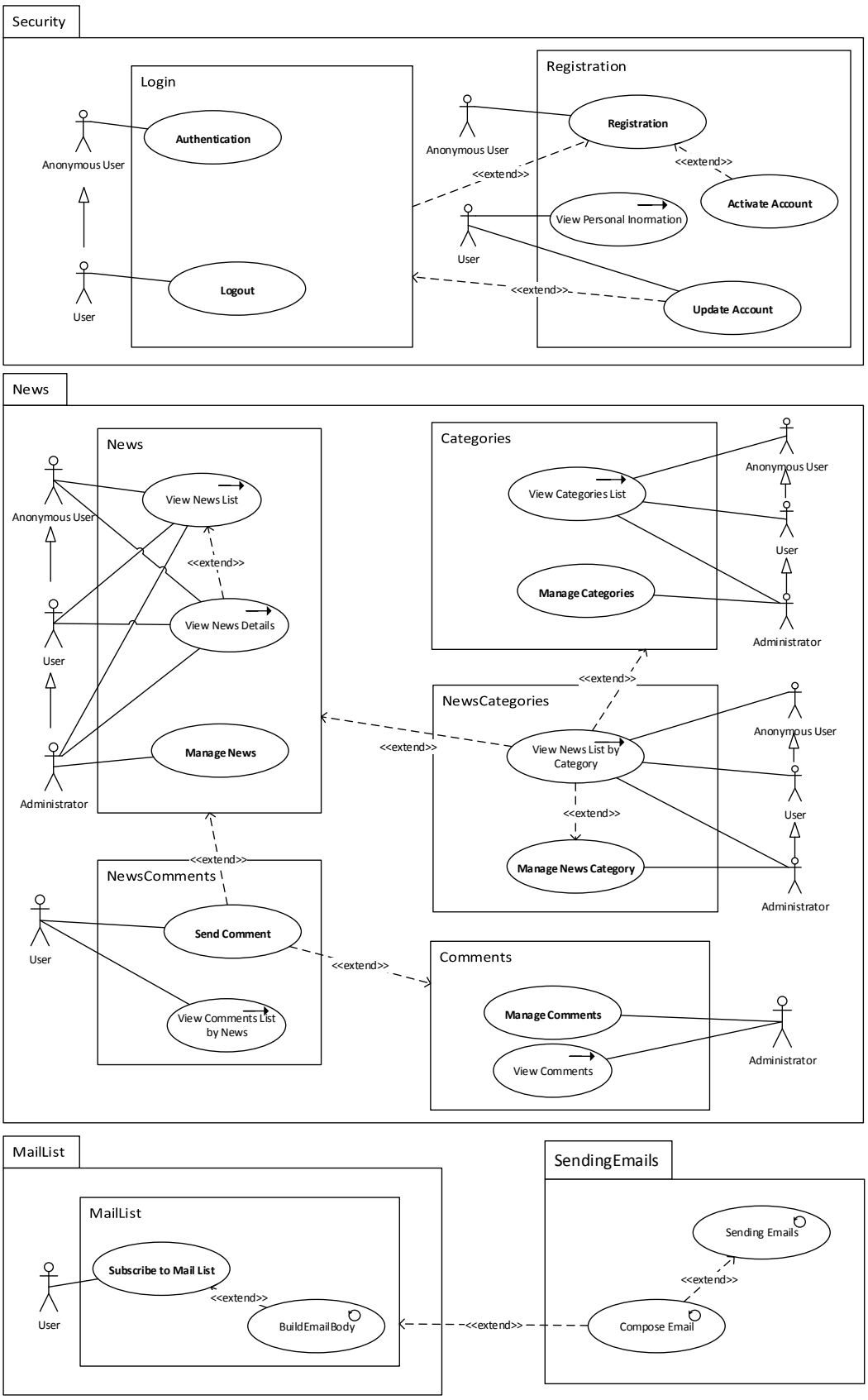

Fig. 3: Use case diagrams for news portal web site

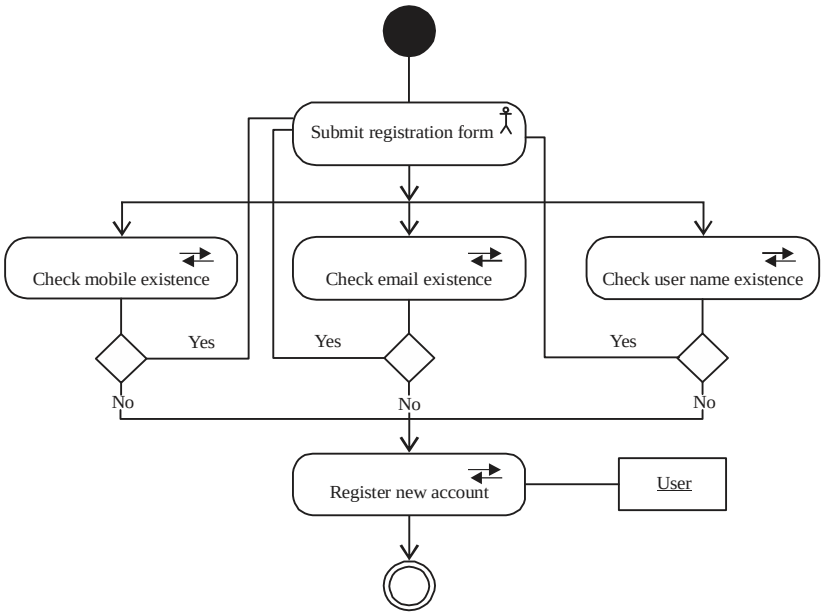

Fig. 4: Registration activity diagram for news portal web site

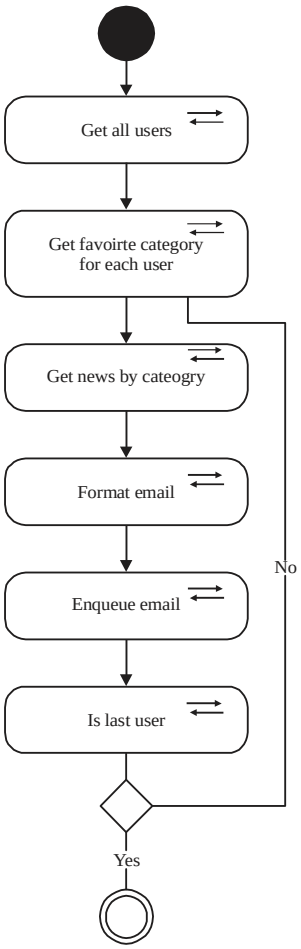

Fig. 5: Compose email activity diagram for news portal web site

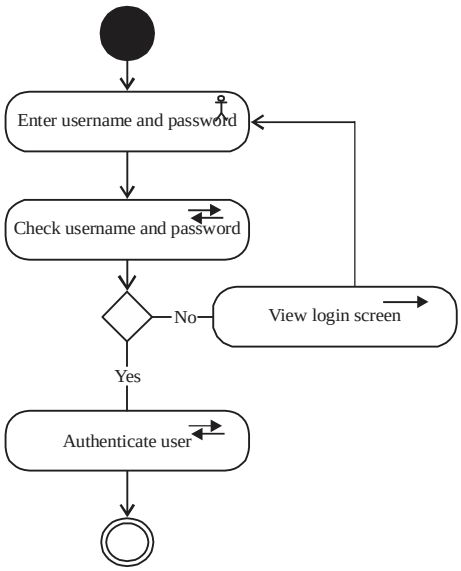

Fig. 6: Authentication Activity diagram for news portal web site

Navigation space model shows which news portal objects can be visited from three actors (anonymous, user and administrator) perspective and determines which property and methods can be accessed as shown in Fig. 12-14. Static presentation for news portal web site is modeled as shown in Fig. 15.

E-trading web based application: This is a trading system in which customer can login in two steps. The first will be using a username and a password for authentication then using

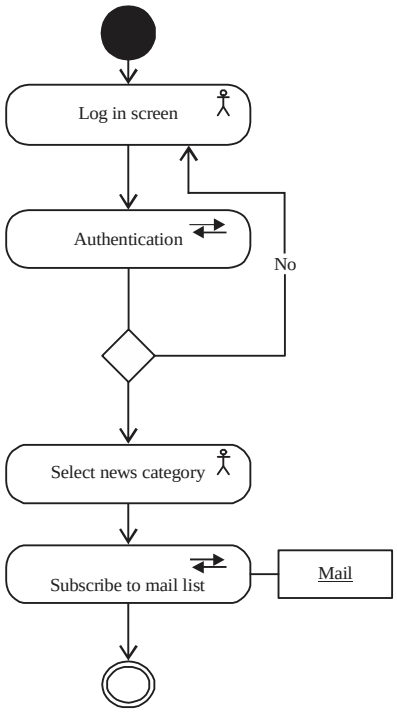

Fig. 7: Subscribe to mail list activity diagram for news portal web site

One Time Password (OTP) generated with a token to verify the authenticated user. After login, customers can submit orders to buy or sale stocks or browse news about current stocks. When buying stocks the E-trading web based application validates customer balance via third party system to check if the customer balance allows him to buy or not. Customers can browse their orders status. System administrator adds new customers and news about stocks to the system.

Initiation phase breaks E-trading web system into five domains; security, stocks, news, customers and stock news. Security domain contains three modules login OTP and registration. Login module is developed with reuse from news portal web site case study without any modifications as well as the news module. Registration module in security domain is developed with reuse from news portal web site but with some modification, other modules will develop for reuse.

In the analysis phase, use case diagram for login and news modules are modeled the same way as the news portal web site as they develop with reuse. Registration module has some modifications from the news portal web site as shown in Fig. 16. Other modules develop for reuse. Activity diagrams related to login and news modules are modeled in the same way as news portal web site, the authentication activity diagram in Fig. 6 and the registration activity in Fig. 4 are used here without any modifications. A new activity diagram for validation token and buy stock use cases are modeled in Fig. 17 and 18.

In the design phase for the E-trading web based application, there is no need to model extend object model, login and news modules are developed with reuse from news portal web site without any modifications. Figure 19-21 show design phase for E-trading web based application where registration module is developed with reuse and modified by adding and deleting methods or property to classes.

Navigation space model shows which E-trading web based objects can be visited from three actors (anonymous, customer and administrator) perspective and determines which property and methods can be accessed as shown in Fig. 22-24. Login and news module are modeled the same in news portal web site for three actors as they are developed with reuse.

Static presentation for E-trading web based application shown in Fig. 25. Login and news module are modeled the same way as in the news portal web site since they are developed with reuse. Registration module has modification from the original that exists in news portal web site.

Training center management system: This system manages training center instructors, trainee, training halls, courses and training halls schedules. System admin has the ability to add or edit training halls, announce new courses, browse training halls schedules, add or accept requests for halls reservation. Instructors can view their schedule, add or cancel schedule. Trainee can browse training center web site, view their schedules, reserve in new course and payment online.

The initiation phase breaks training center management system into five domains; security, halls, news, courses and schedules. Security domain contains three modules login, registration and user roles. Login and registration modules are developed with reuse from news portal web site case study without any modifications as well as news module. Other domains and modules are developed for reuse.

Figure 26 show use case diagram for training course management system where login, registration and news modules are developed with reuse as they are modeled the same way as in news portal web site without any modifications besides their related activities which were modeled.

Other domains and modules are developed for reuse. Conceptual model shown in Fig. 27-29. Login, news and registration modules are modeled the same as in news portal web site since they are developed with reuse. Navigation

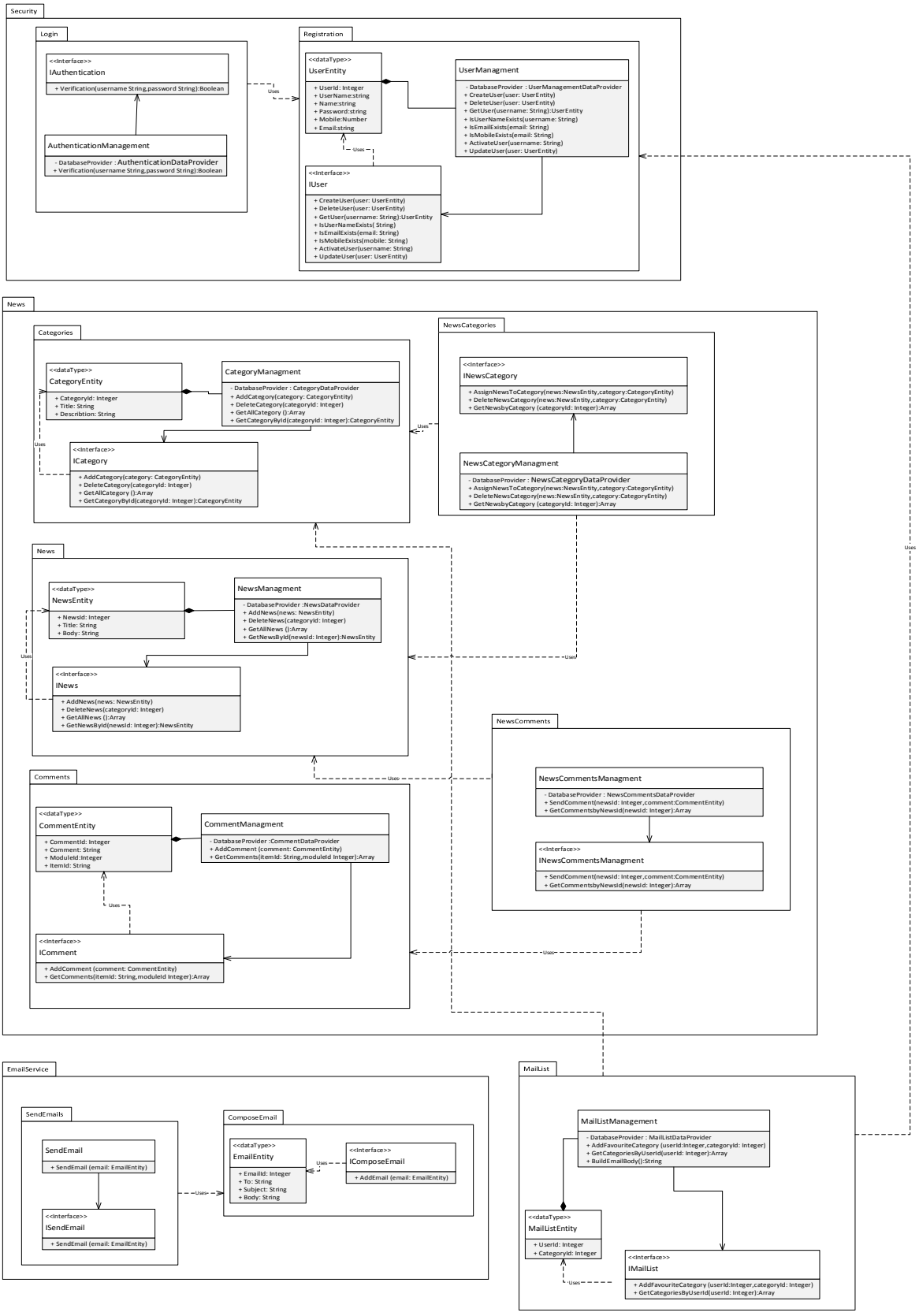

Fig. 8: Entity object model diagram for news portal web site

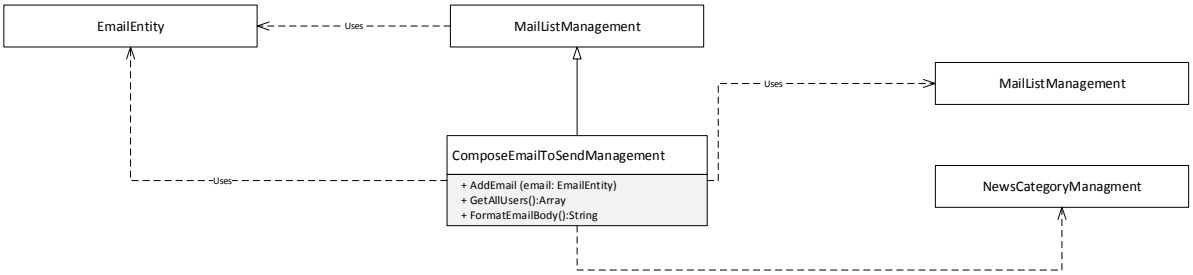

Fig. 9: Extended object model diagram for news portal web site

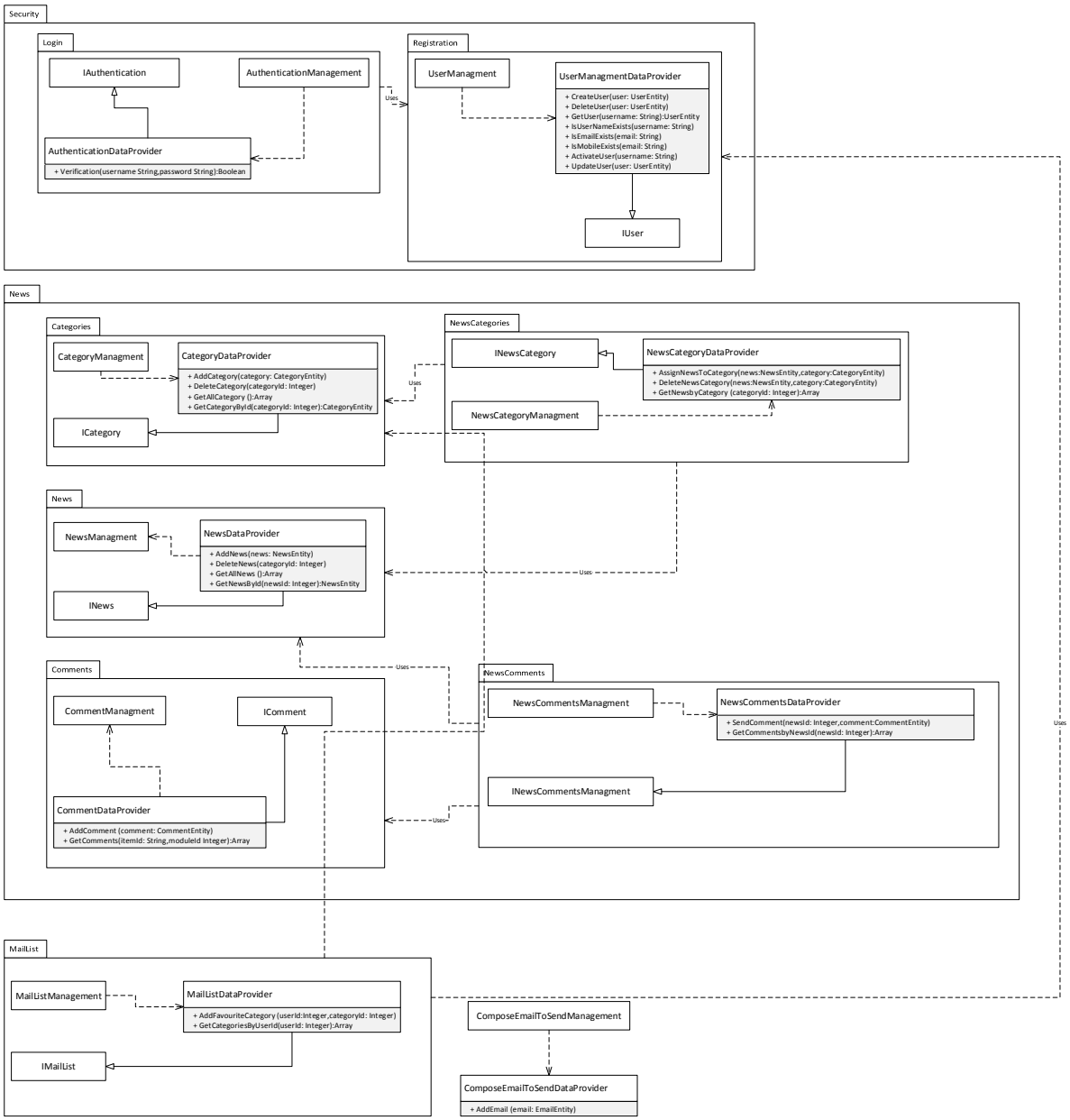

Fig. 10: Data provider model diagram for news portal web site

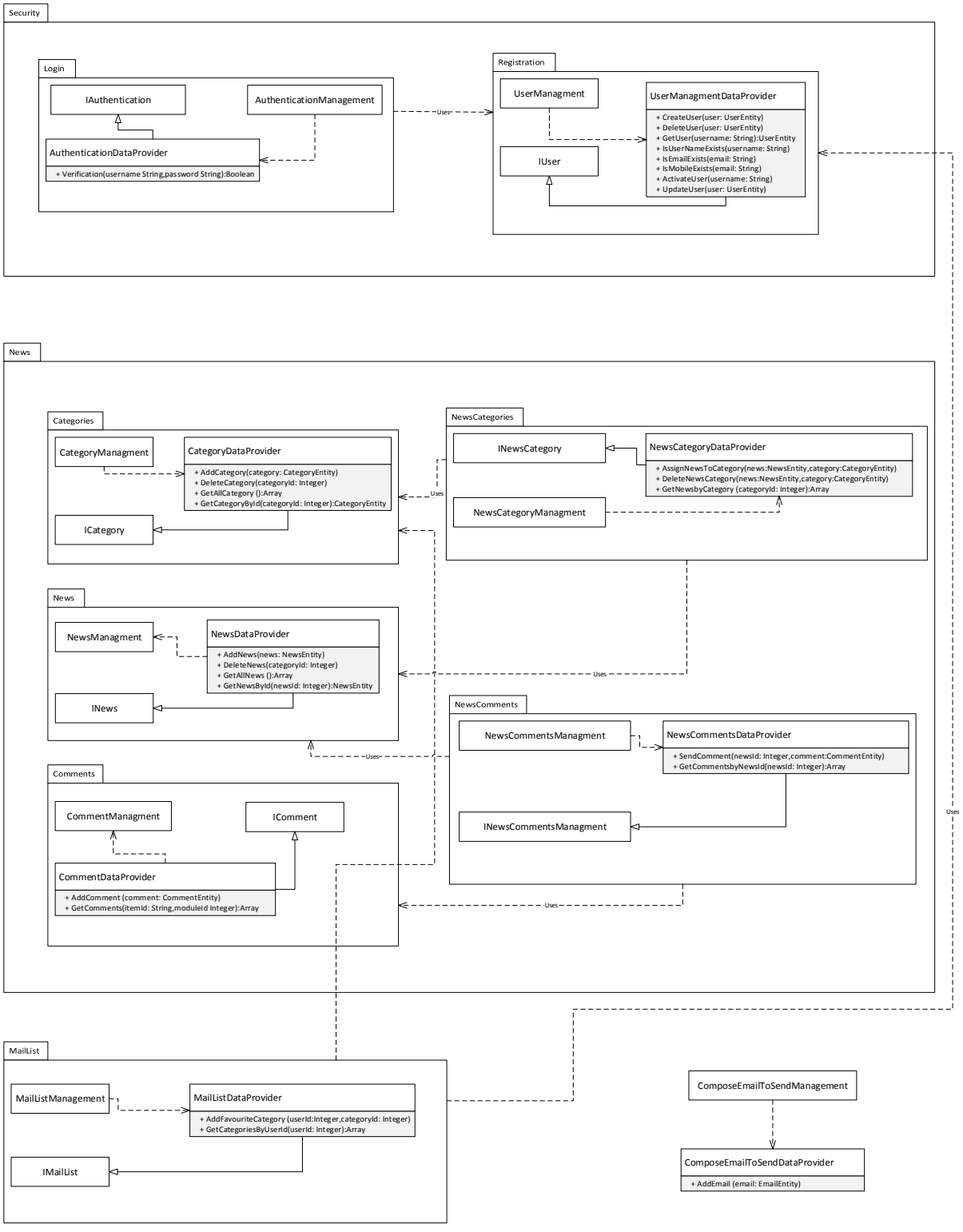

Fig. 11: Rendering model diagram for news portal web site

space model have four perspectives: Anonymous user, administrator, instructor and trainee shown in Fig. 30-33. Static presentation for training center management system is shown in Fig. 34.

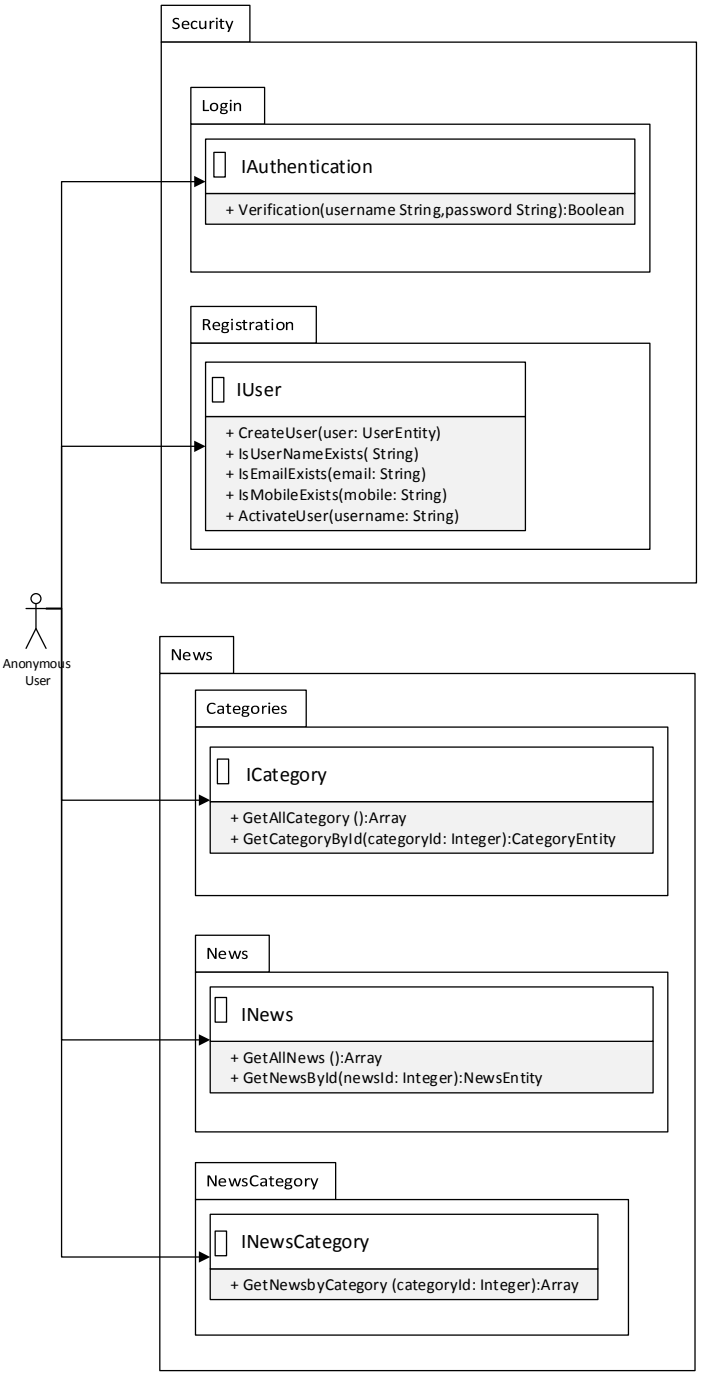

Fig. 12: Navigation space model for news portal web site describing the navigation from anonymous user perspective

DISCUSSION AND OPEN ISSUES

The effects of applying MDWER approach were measured on the web system development life cycle. Table 1 shows a comparison between the previous three case studies showing the number of reusable assets that were developed with reuse. Note that the number of reusable assets that was developed with reuse increases as long as we apply this approach. The first time after application of this approach it consumes the usual time to develop such web system from scratch, but as soon as this approach was applied for long term, the time consumed in the development process decreases.

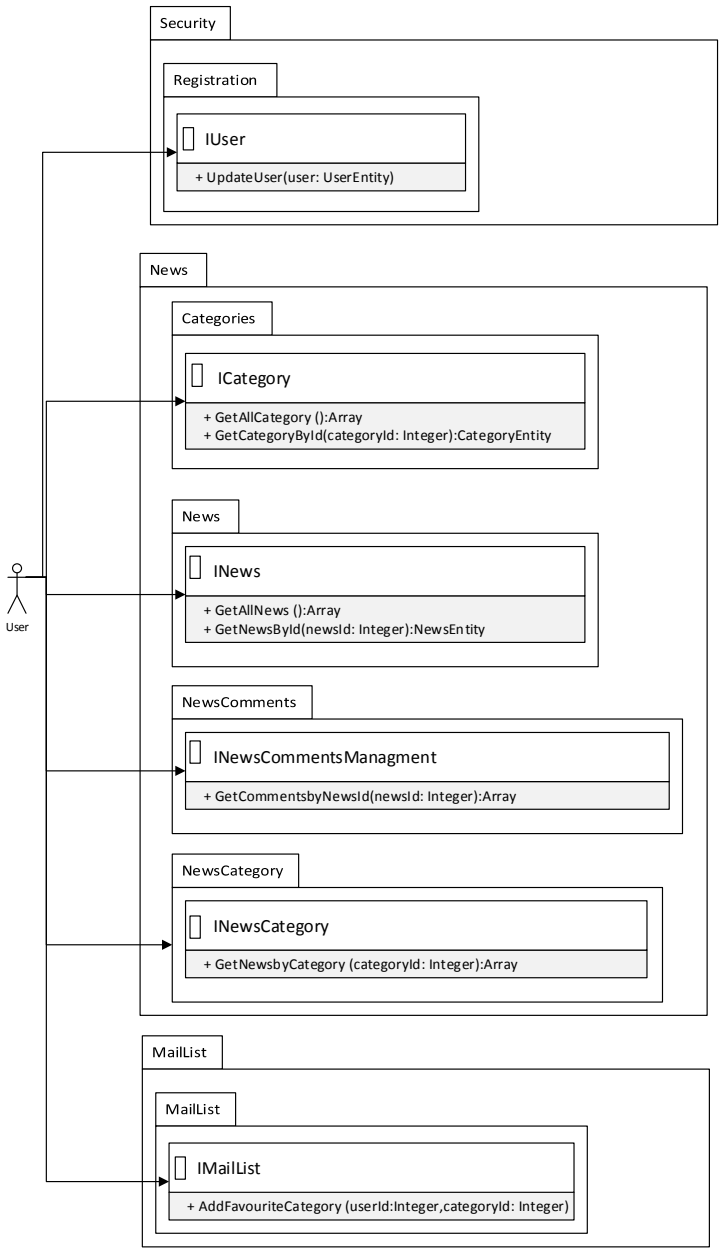

Fig. 13: Navigation space model for news portal web site describing the navigation from users perspective

Table 1: Number of reusable assets that developed with reuse for the three case studies

Reusable assets	Case study A	Case study B	Case study C
Modules	0	3	3
Use case	0	8	9
Activity diagram	0	2	2
Entity object model (Classes)	0	8	8
Data provider model (Classes)	0	3	3
Rendering model (Classes)	0	3	3
Navigation space model (Navigational class)	0	3	3
Static presentation model (Presentation page)	0	6	6

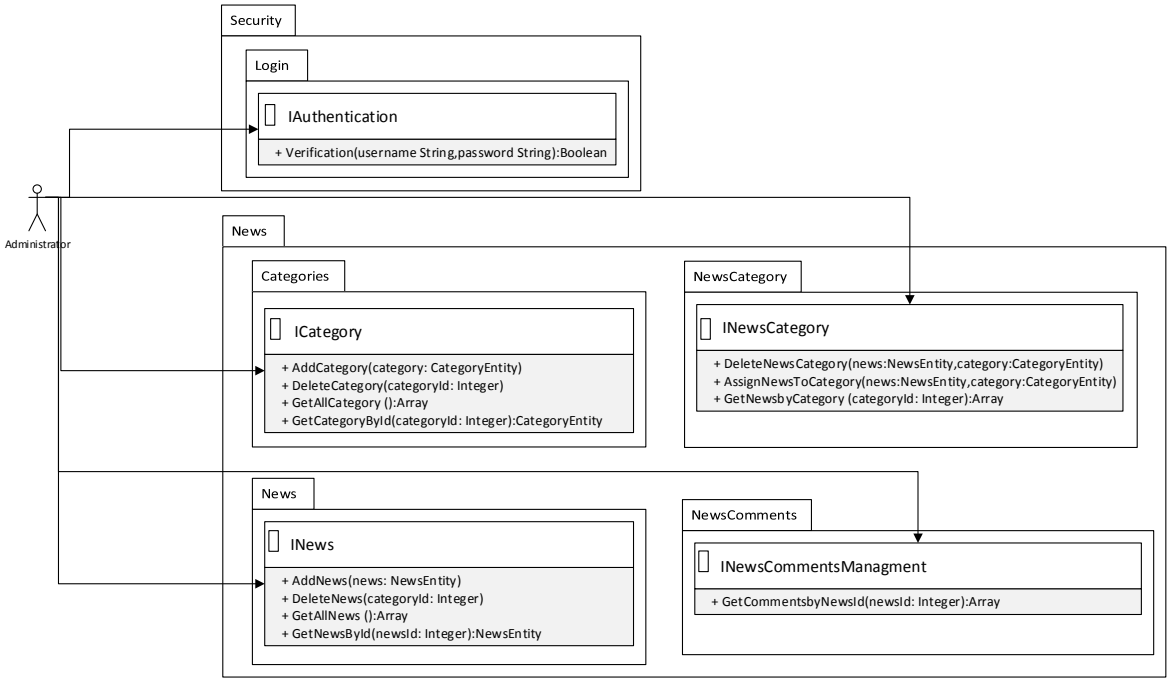

Fig. 14: Navigation space model for news portal web site describing the navigation from administrator prespective

Table 2: Estimated time with working hours for the development process of the previous three case studies with and without applying WER

	Case study A		Case study B		Case study C	
Phases	Without WER	With WER	Without WER	With WER	Without WER	With WER
Initiation	0	10	0	15	0	20
Analysis	30	30	50	30	60	39
Design	60	60	100	75	120	78
Implementation	160	160	200	150	240	156
Testing	40	40	50	35	60	39
Total hours	290	300	400	305	480	332

Table 2 shows the estimated time consumed to develop the previous three case studies with and without applying WER approach. The total estimated time to develop case study A using WER approach is more than the estimated time to develop without using this approach because we still do not have any reusable assets yet. Developing case study B with WER approach is less time consuming than developing without it as well as case study C.

The main contributions for MDWER are: (1) Using UML and UWE to provide a formal documentation about MDWE and relates these artifacts with implementation codes and test cases which lead to decrease time and cost for modification and maintenance, (2) Software house companies view software reuse as a separated task that needs analysis, design, implementation and testing phases in addition to dedicated team to build these reusable assets that is why they believed that software reuse costs more money and efforts (Sommerville, 2011), MDWER planned software reuse as part of an organization and involves reusability with development process since they can apply it without changes in their development methodology since this approach adds the initiation phase to the four commons phases (analysis, design, implementation and testing) in any development model they are using, (3) This approach can be considered by software house companies as a software mega store that contains not only implementation codes but also the related business analysis, design artifacts and test cases along with the ability to modify the existing web system assets and store it as a new version to form a new MDWE, (4) MDWER uses UWE framework to model difference concerns for MDWE like content, navigation, interactivity and presentation, it also helps to reuse these artifacts to build new MDWE and

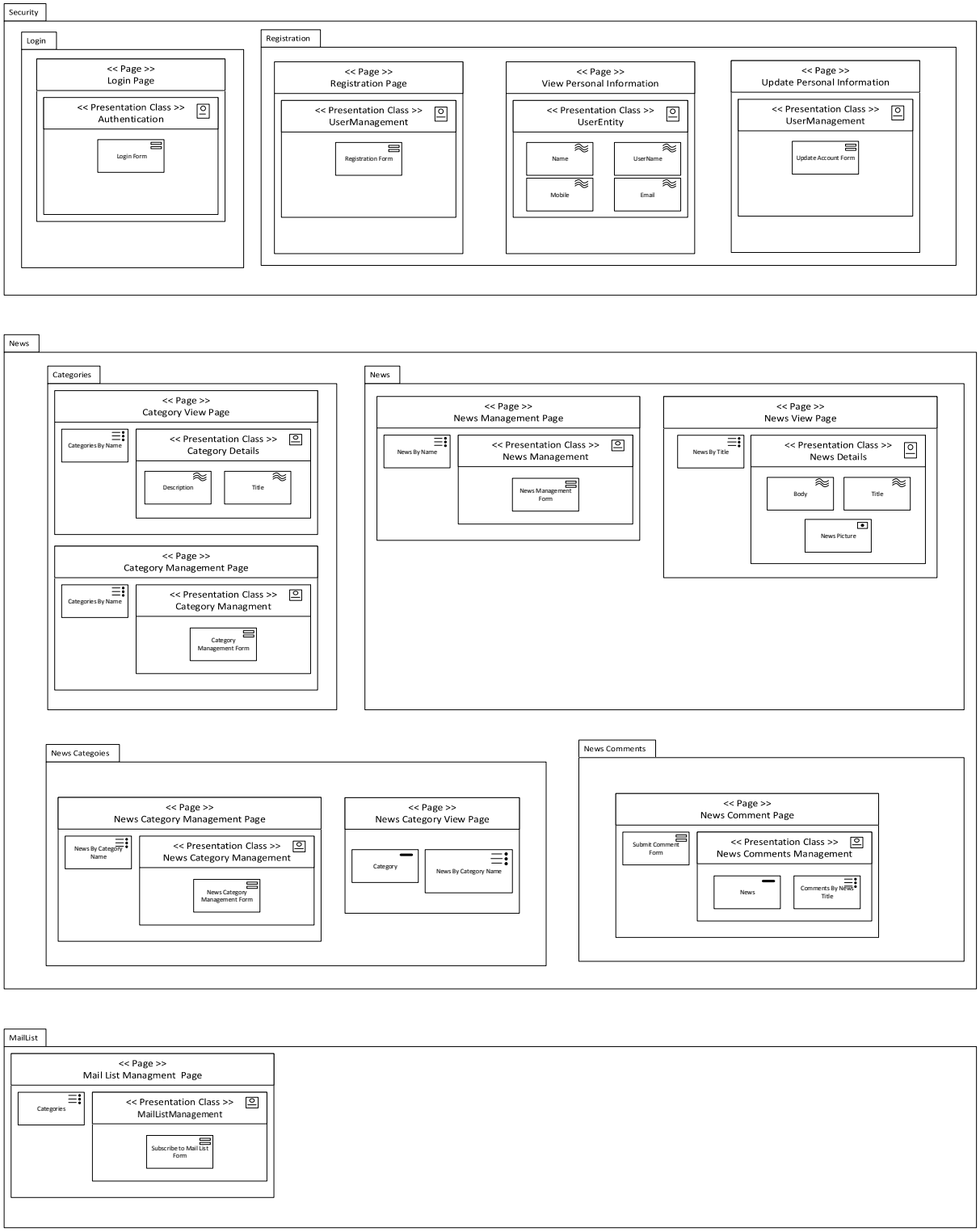

Fig. 15: Static presentation diagram for news portal web site

(5) This approach does not support auto-generation for codes to avoid generation of unused code or using a specific design pattern, it allow developers to implement their web application domain using their preferred design pattern and code standard which leads to reuse implementation codes in easily way and decrease maintenance costs.

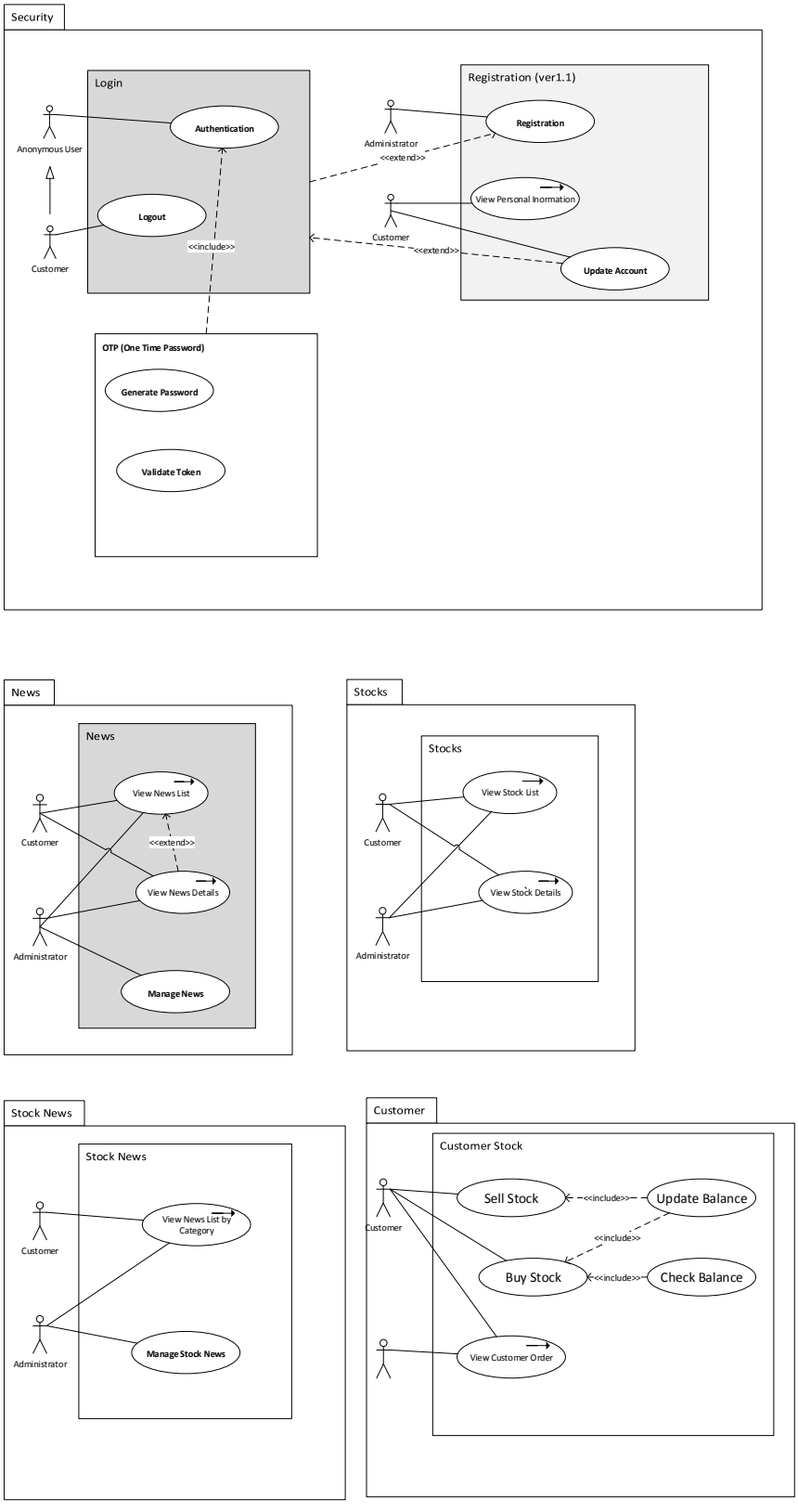

Fig. 16: Use case daigrams for E-trading web based application

Table 3: Comparison between web system development with and without applying MDWER

Factors	Development with MDWER approach	Development without MDWER approach
Time consuming for developing web system on short term	More time consuming since the initiation phase takes place in the early development process	Less time consuming since there is no initiation phase only the four common phases
Time consuming for developing web system on long term	Less time consuming for the development process since the reusable assets are increased which lead to increase the number of assets developed with reuse	More time consuming since no reusable assets are available and the development process starts from scratch with analysis, design, implementation and testing phases without reuse
Productivity	Increase in productivity as long as we apply this approach in the development process	Poor productivity as the development process for pre-developed web system is duplicated
Risk process	Low risk process as we build new web system from predefined and tested assets	High risk process as the development process and testing phase start from scratch
Number of stakeholders	A huge library of reusable assets will be available after a period of time applying this approach, so software companies can decrease the number of stakeholders	As every new web system needs stockholders for the development process which start from scratch, so software companies cannot decrease the number of stockholders
Potential to create competitive advantages	High	Low
Cost	Low cost for the long term as we reuse a proven tested artifacts, implementation codes or test cases in addition to reduce maintenance cost	High cost for long term as we duplicate development process, codes or test cases in addition to the maintenance cost

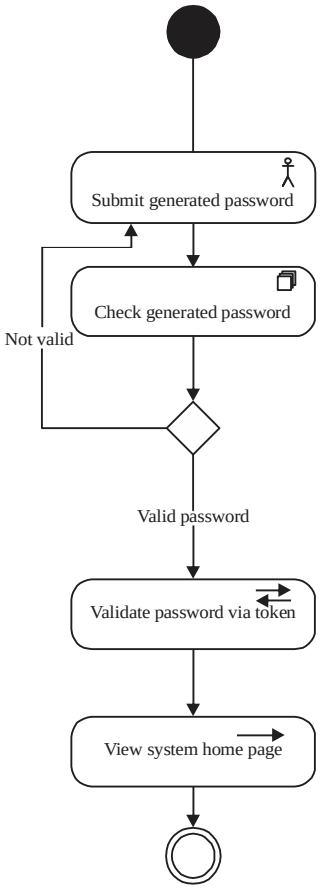

Fig. 17: Validate token activity diagram

As a part of this study, a set of factors important for software houses was identified when they develop a new web system, this study compare between web system development with and without applying MDWER referring to these factors, this comparison shown in in Table 3.

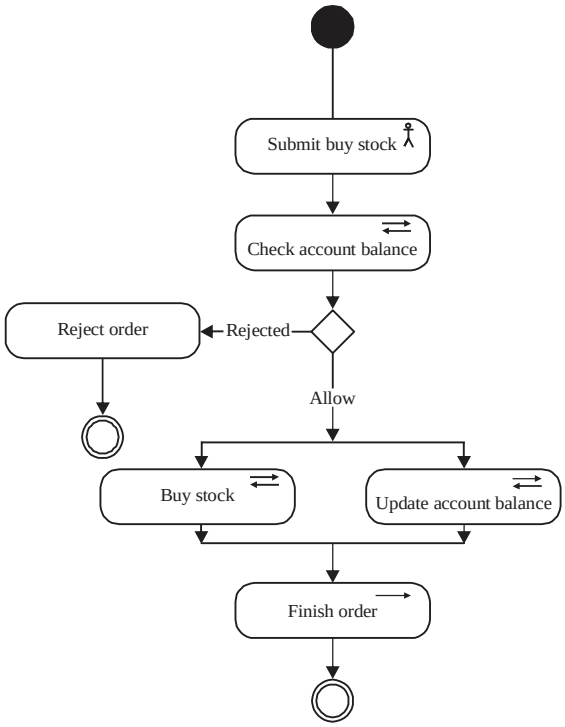

Fig. 18: Buy stock activity diagram

In the first period of time applying MDWER to build model driven web engineering (application domain), the development cost will increased as it takes additional time in the initiation phase to determine the reusable assets types; but as long as we apply this approach, the number of reusable assets stored in the developing case tool will increase which leads to decrease the time required for build a new MDWE which in turn lead to decrease the development cost and time.

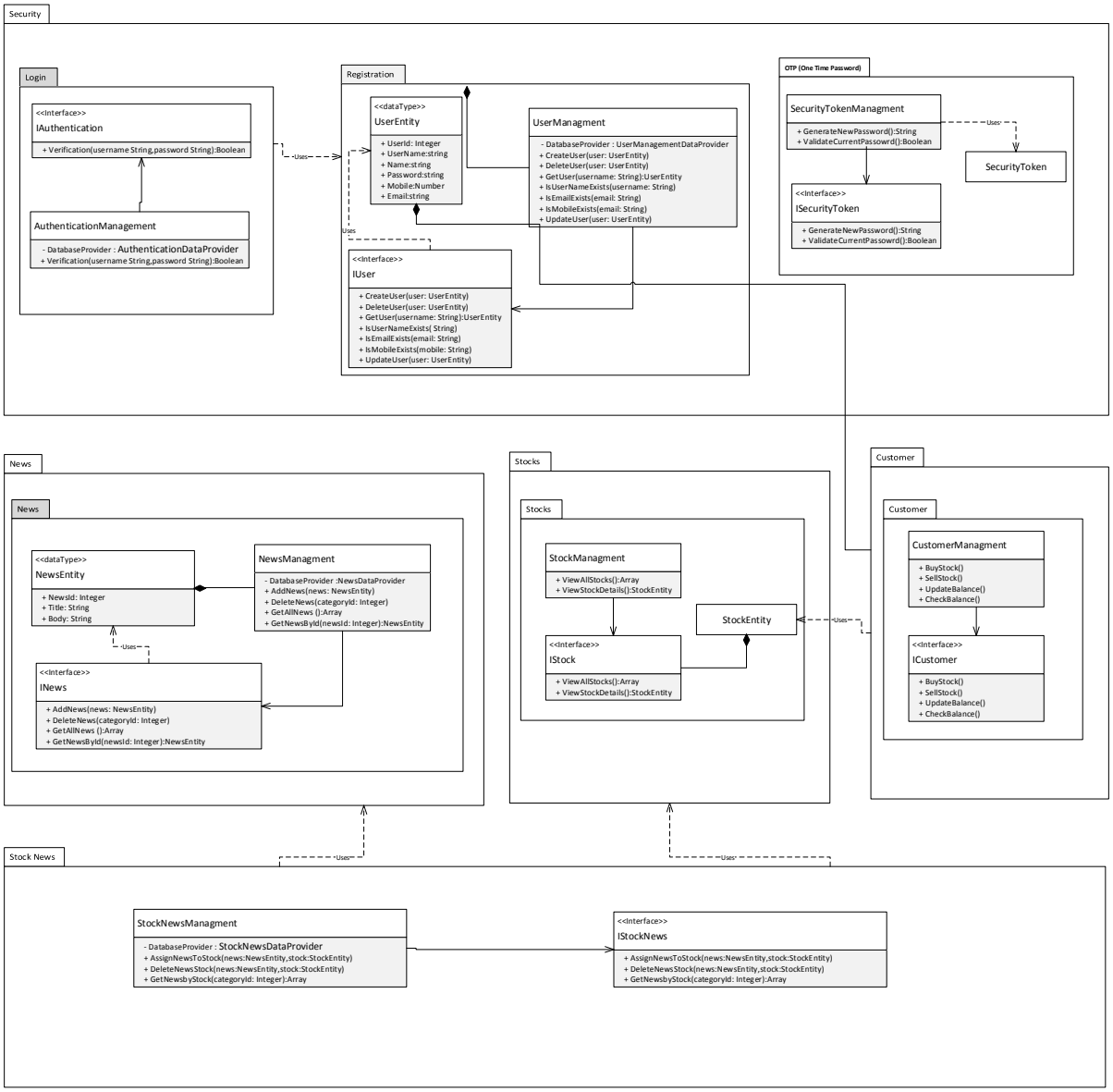

Fig. 19: Entity object model diagram for E-trading web based application

This approach combines the strength of different reuse approaches and allows stakeholders to build their own domain specific, reuse it and modify it as needed. The MDWER takes the advantages of application product line approach, where generalized common models can be adapted for different customers; code generation approach, where implementation code is related to different web system development phases and modeling artifacts; component-based development approach, where conceptual model is used to develop reusable component and finally the CMS approach where web application domains and modules are ready to be used.

There are a set of open issues and challenges associated with software and web system reuse, we summarize it as follows:

- One of the most important problems in software reuse is assets indexing, storing and maintenance (De Almeida *et al*., 2005)
- Search about reusable assets is not investigated since no similarity metrics to search for reusable assets (Keswani *et al*., 2014)
- Reusability measurements for web system is not realized

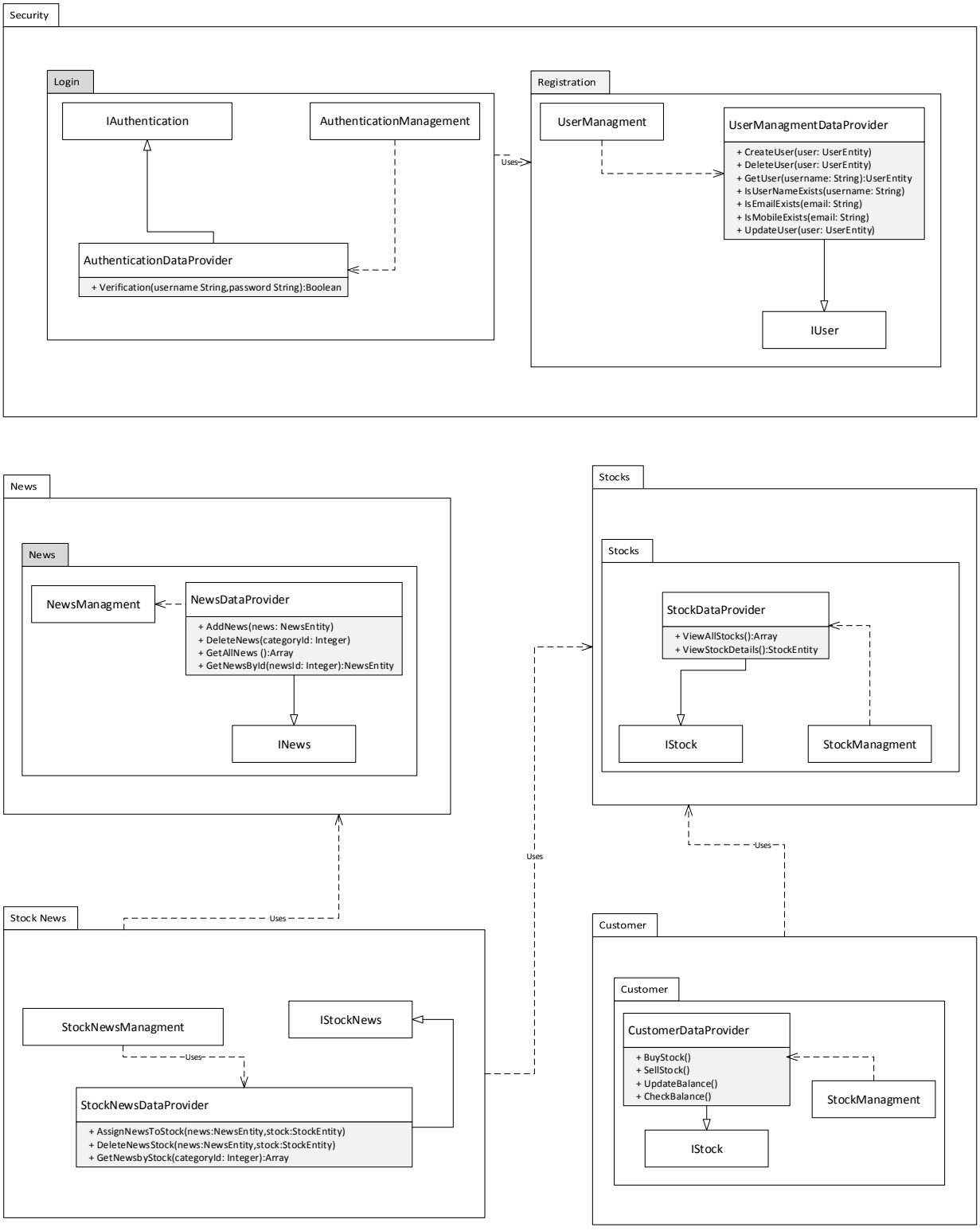

Fig. 20: Data provider model diagram for E-trading web based application

- Developing web system needs many participators with different technology background, web system reuse approaches does not provide multi-view for reusable assets of web system

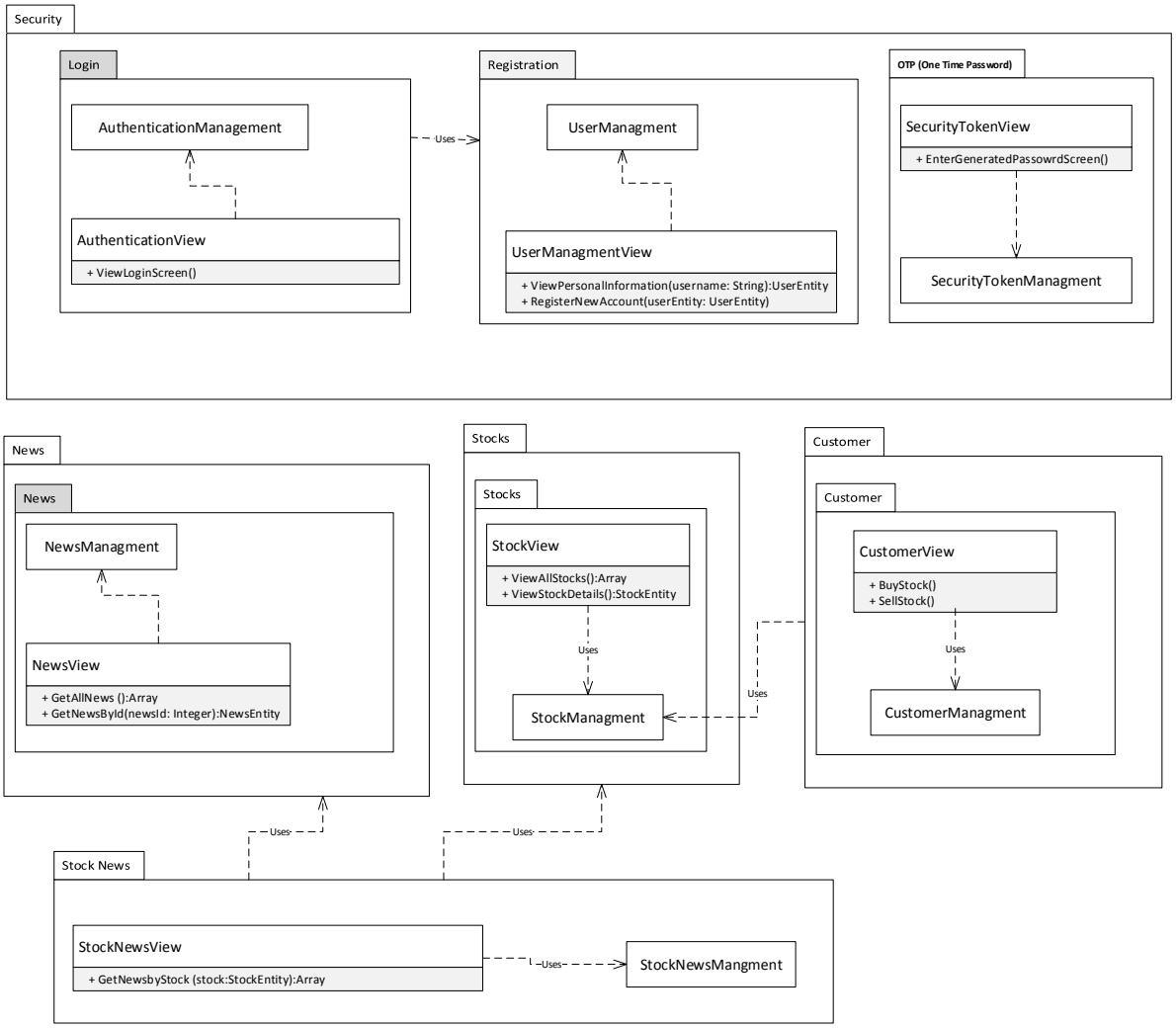

Fig. 21: Rendering model diagram for E-trading web based application

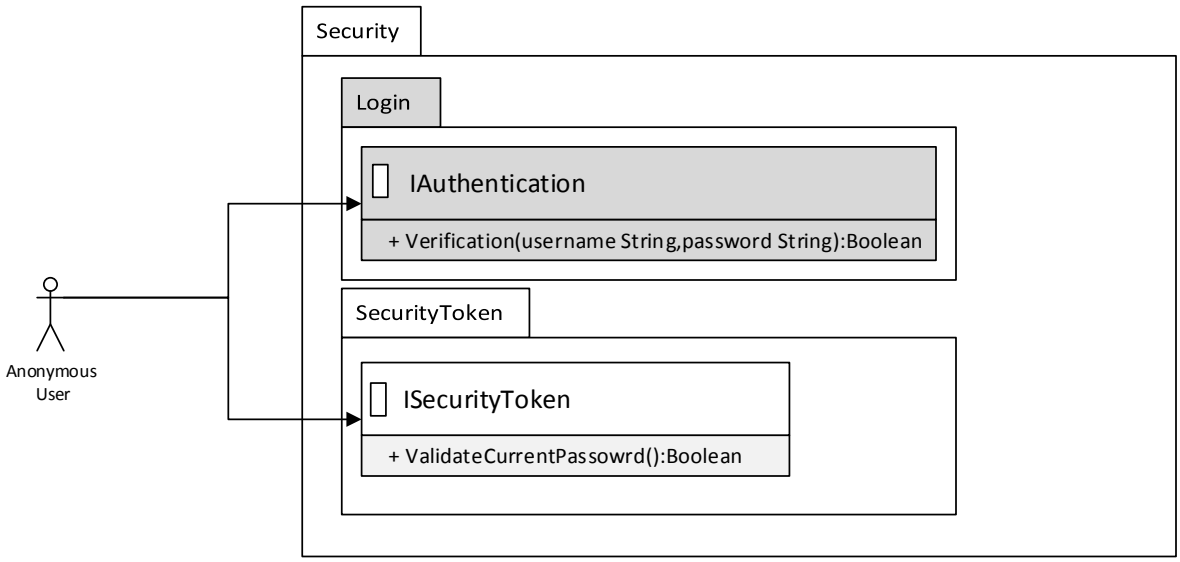

Fig. 22: Navigation space model for E-trading web based application from anyonymous user prespective

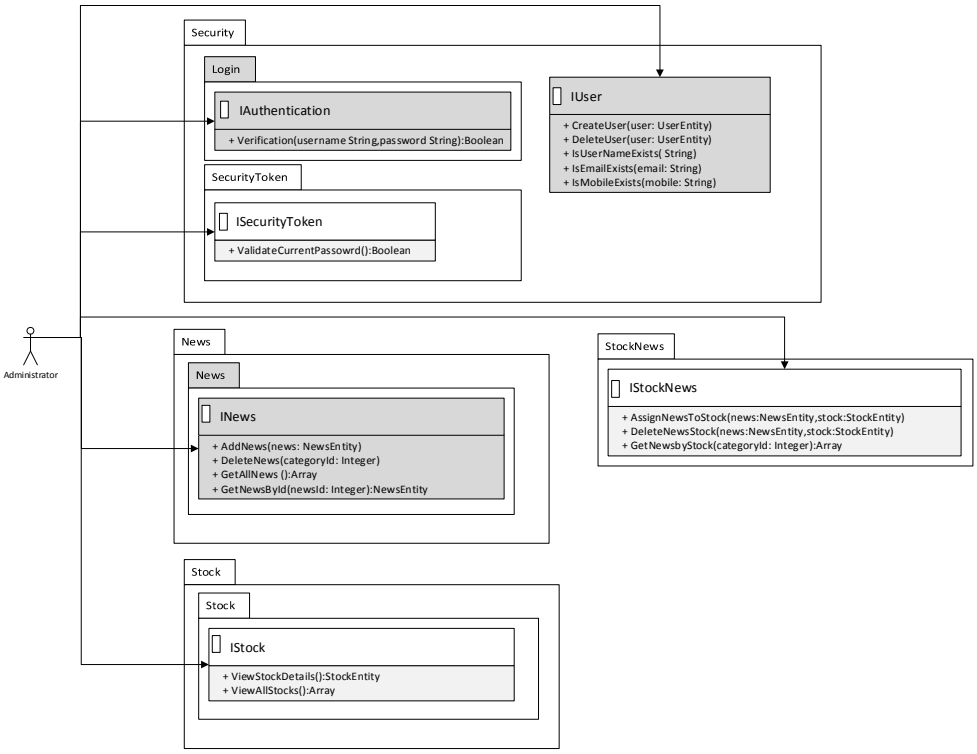

Fig. 23: Navigation space model for E-trading web based application from administrator prespective

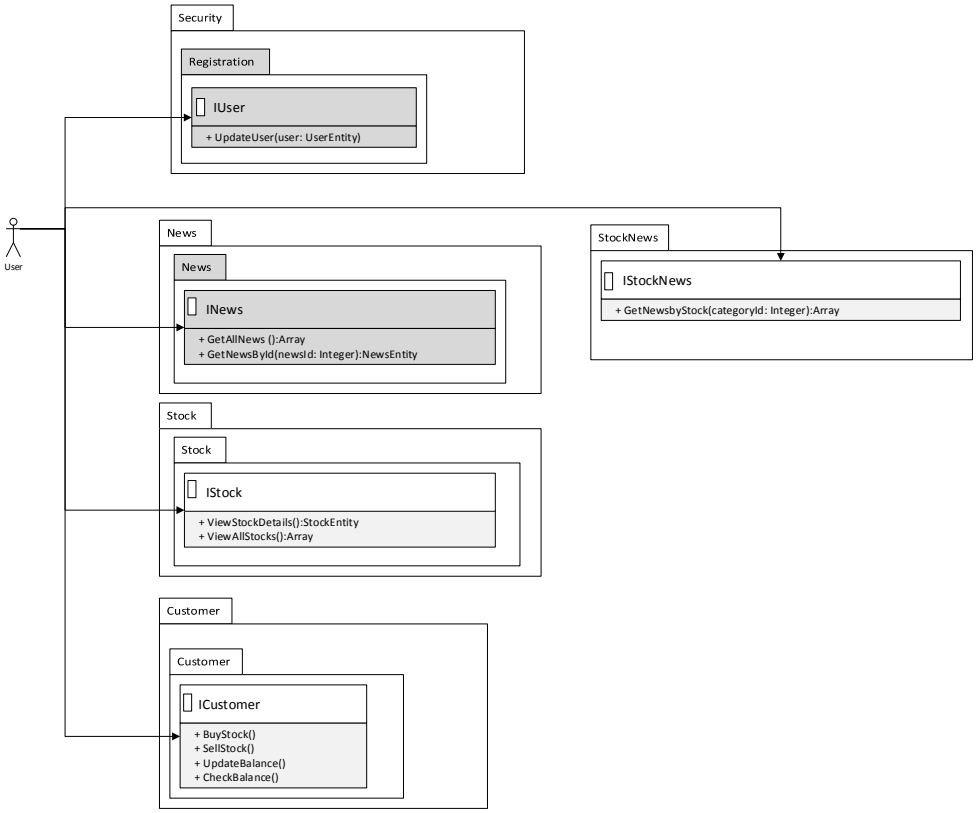

Fig. 24: Navigation space model for E-trading web based application from users prespective

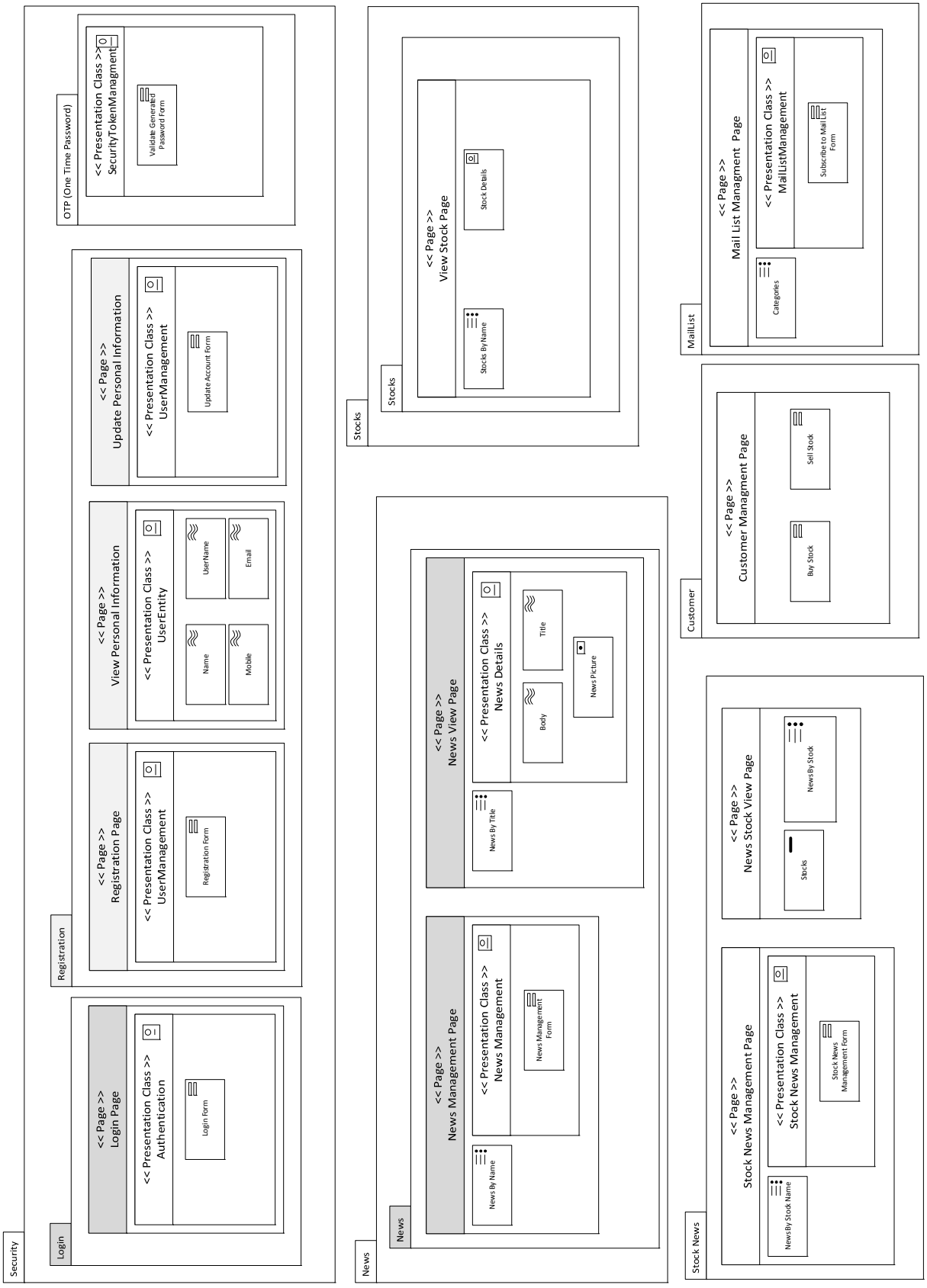

Fig. 25: Static presentation for E-trading web based application

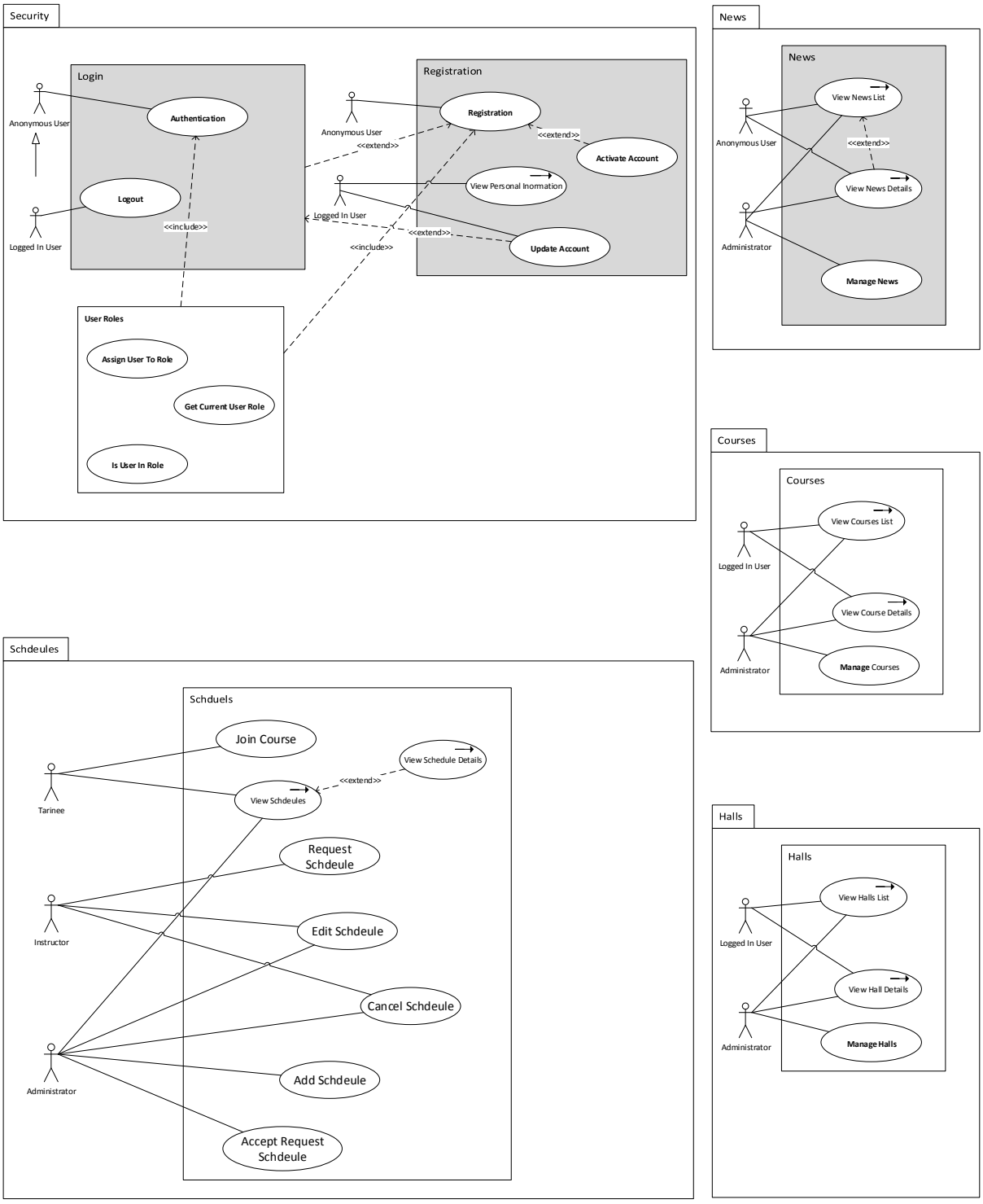

Fig. 26: Use case diagrams for training center managment system

- Web system reuse with a platform-independent is important issue as web system can be accessed via different platforms such as different devices with different operating systems

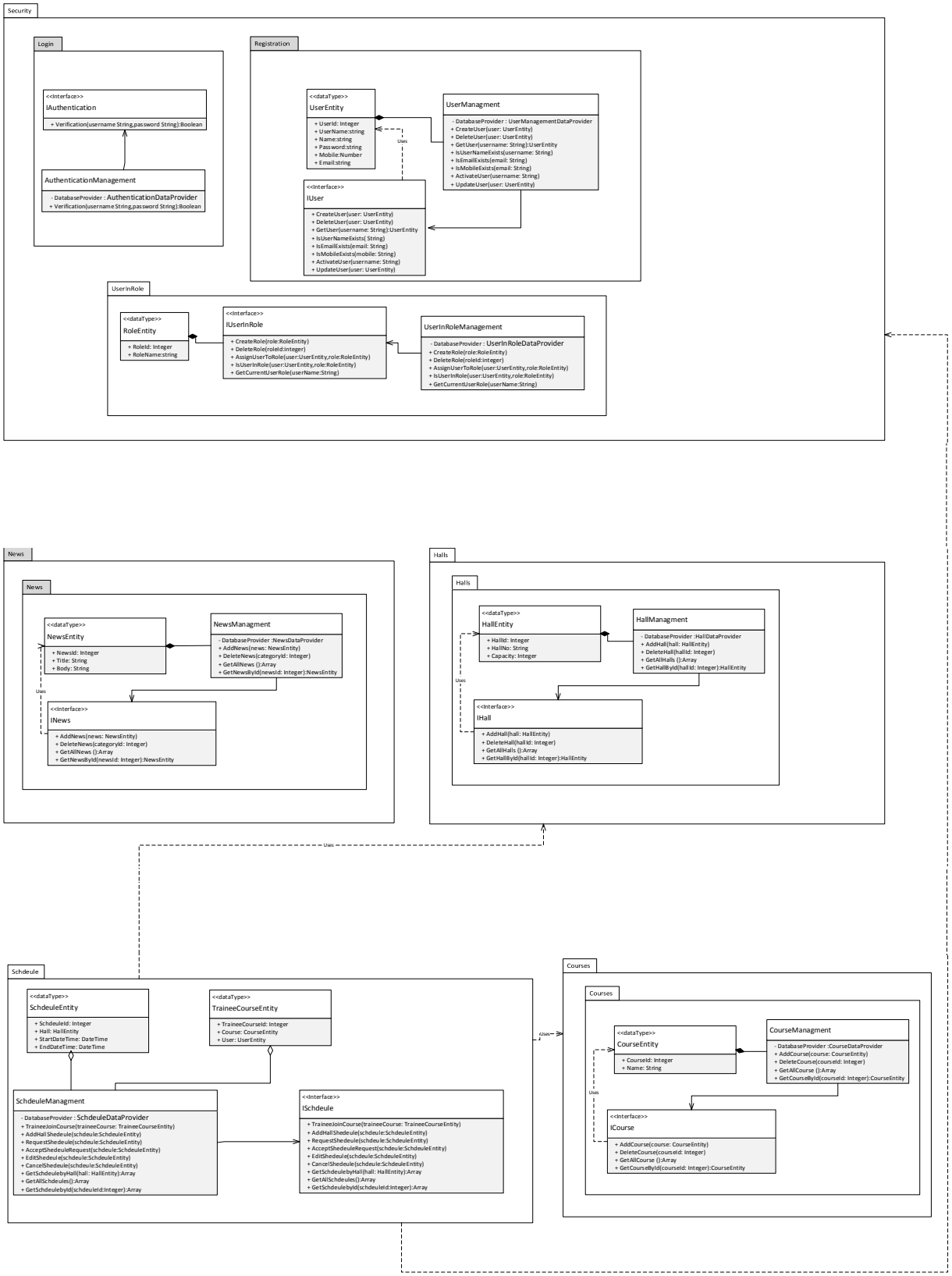

Fig. 27: Entity object model diagram for training center managment system

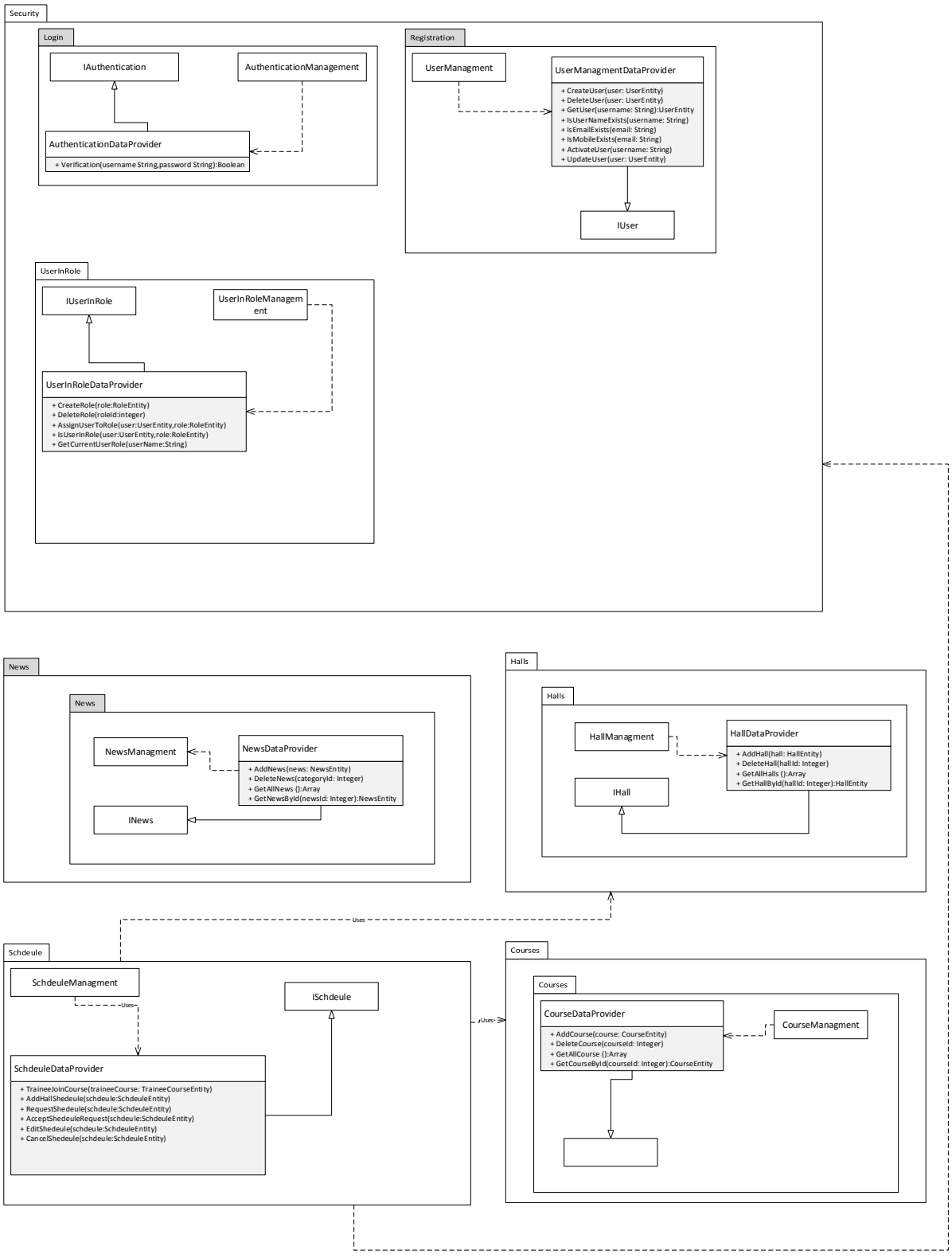

Fig. 28: Data provider model diagram for training center managment system

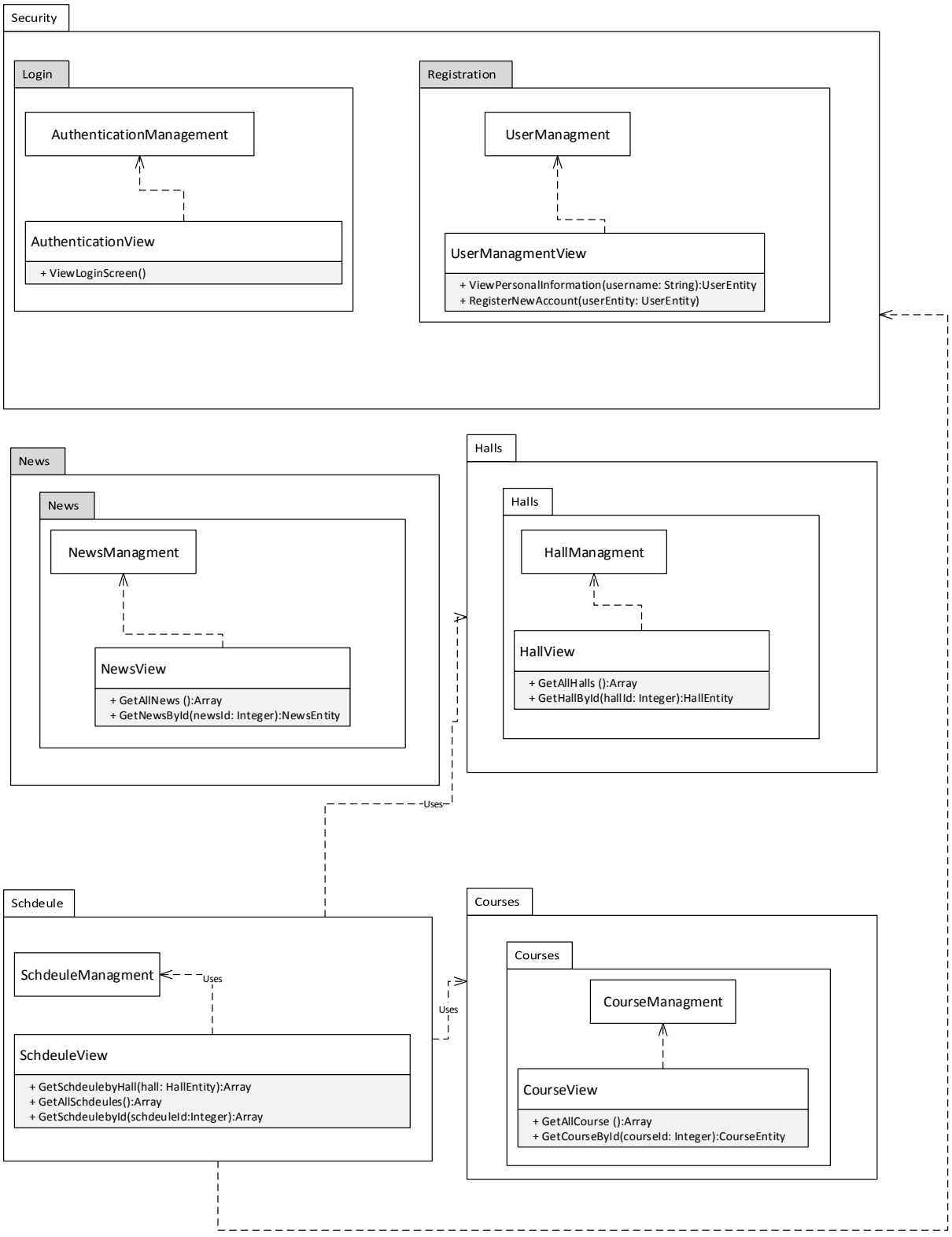

Fig. 29: Rendering model diagram for training center managment system

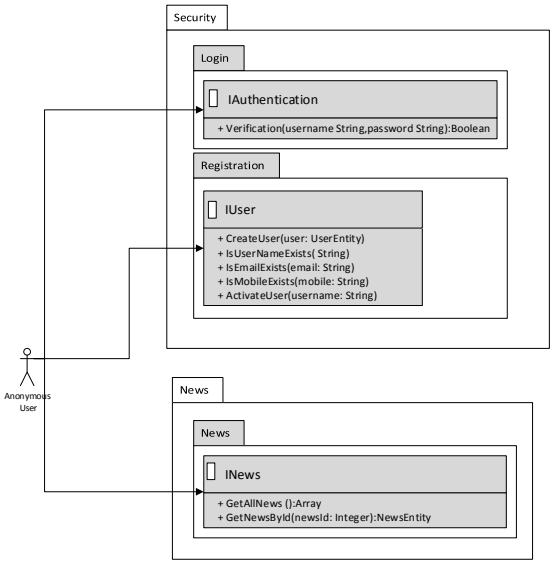

Fig. 30: Navigation space model for training center managment system from anonymous user prespective

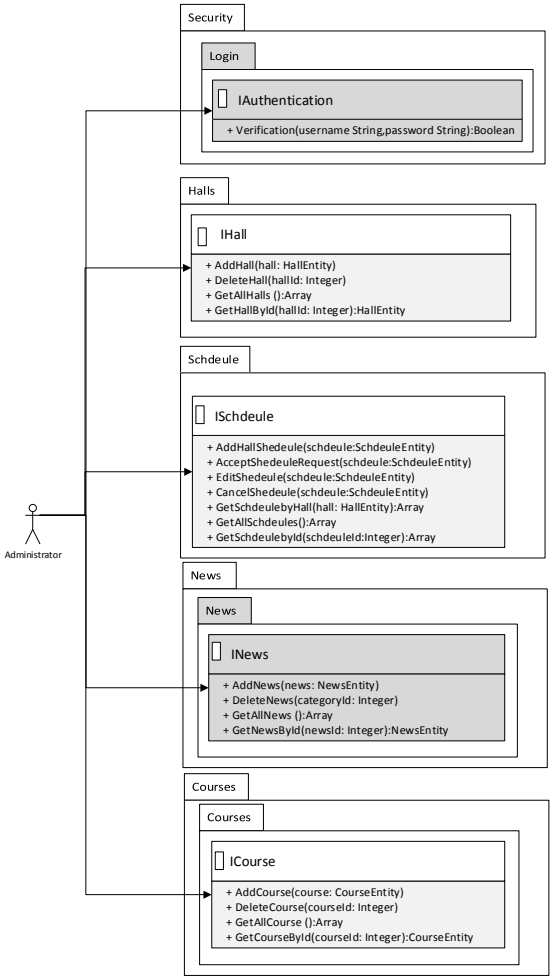

Fig. 31: Navigation space model for training center managment system from administrator prespective

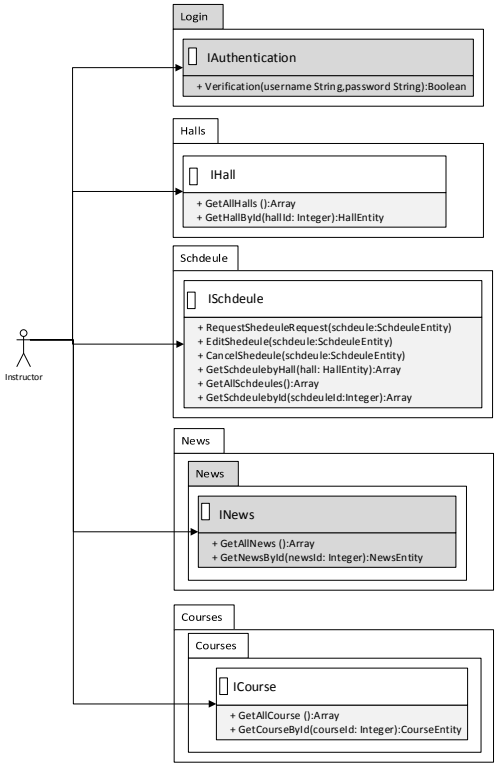

Fig. 32: Navigation space model for training center managment system from instructor prespective.

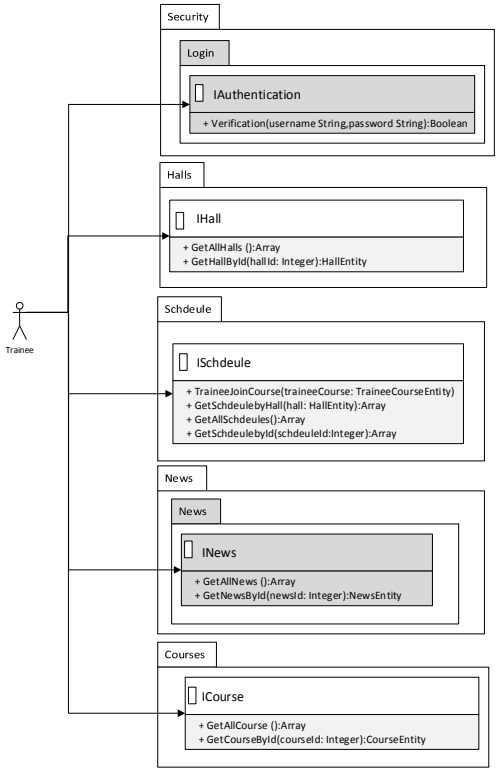

Fig. 33: Navigation space model for training center managment system from trainee prespective

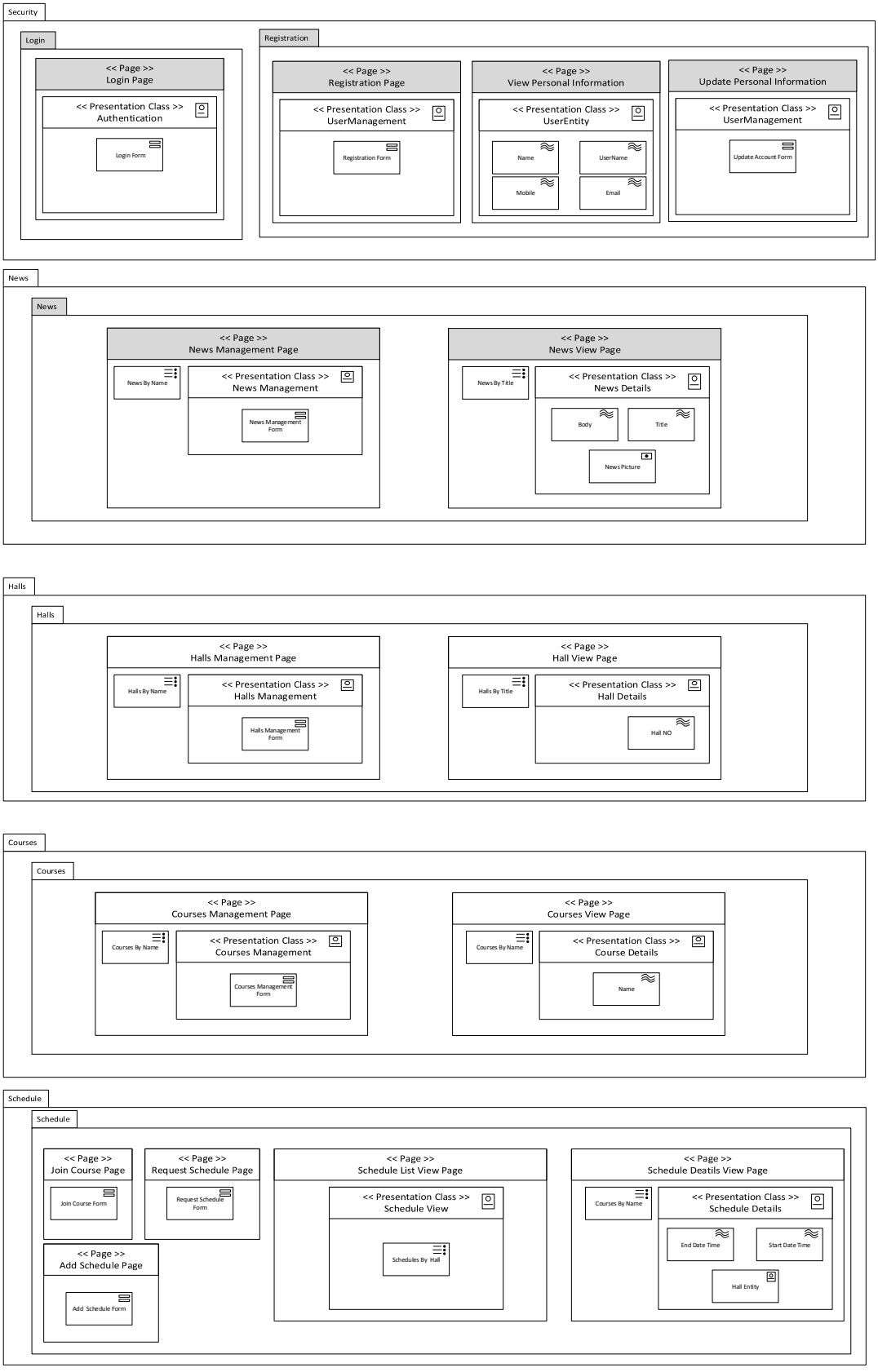

Fig. 34: Static presentation for training cener managment system

CONCLUSION

The MDWER approach helps to build new reusable model driven web engineering through integrating component-based architecture with service oriented architecture approaches, it breaks web application domain vertically and horizontally helps to component reuse and application domain reuse; it also involves reusability in the early development phases to relates reusability with the development process. This approach uses UML and UML-based web engineering to express the MDWE and stores these artifacts in a case tool with its related implementation codes and test cases.

We will continue to work on this approach to provide a Query Modeling Languages (QML) that can be used to search in the stored modeling artifacts, also we will work on using the modeling artifacts to provide multi-view for web system to different stakeholders, in addition to provide a set of rules that helps to automatically transform from one development phase to the next.

REFERENCES

Ahmed, M., 2011. Towards the development of integrated reuse environments for UML artifacts. Proceedings of the 6th International Conference on Software Engineering Advances, October 23-29, 2011, Barcelona, Spain, pp: 426-431.

Beyer, H.J., D. Hein, C. Schitter, J. Knodel, D. Muthig and M. Naab, 2008. Introducing Architecture-Centric Reuse into a Small Development Organization. In: High Confidence Software Reuse in Large Systems, Mei, H. (Ed). Springer, Berlin, Germany, ISBN: 978-3-540-68062-8, pp: 1-13.

Braga, R.M.M., C.M.L. Werner and M. Mattoso, 2006. Odyssey-search: A multi-agent system for component information search and retrieval. J. Syst. Software, 79: 204-215.

Das, D.K., 2012. An approach to software reuse: Its benefit, opportunity and issue. Global J. Manag. Sci. Technol., 1: 10-14.

De Almeida, E.S., A. Alvaro, D. Lucredio, V.C. Garcia and S.R.L. Meira, 2005. A survey on software reuse processes. Proceedings of the IEEE International Conference on Information Reuse and Integration, August 15-17, 2005, Las Vegas, USA., pp: 66-71.

Elminir, H.K., M.A. Elsoud and A.M. El-Halawany, 2011. UML-based web engineering framework for modeling web application. J. Software Eng., 5: 49-63.

Gerard, R., R.R. Downs, J.J. Marshall and R.E. Wolfe, 2007. The software reuse working group: A case study in fostering reuse. Proceedings of the IEEE International Conference on Information Reuse and Integration, August 13-15, 2007, IEEE, Las Vegas, IL., USA., pp: 24-29.

Keswani, R., S. Joshi and A. Jatain, 2014. Software reuse in practice. Proceedings of the 4th International Conference Advanced Computing and Communication Technologies, February 8-9, 2014, IEEE, pp: 159-162.

Koch, N., S. Melia-Beigbeder, N. Moreno-Vergara, V. Pelechano-Ferragud, F. Sanchez-Figueroa and J.M. Vara-Mesa, 2008. Model-driven web engineering. Upgrade-Novatica J., 9: 40-45.

Mascena,, J.C.C.P., S.R.L. Meira, E.S. de Almeida and V.C. Garcia, 2006. Towards an effective integrated reuse environment. Proceedings of the 5th International Conference on Generative Programming and Component Engineering, October 22-26, 2006, Portland, OR., USA., pp: 95-100.

Moreno, N., J.R. Romero and A. Vallecillo, 2008. An Overview of Model-Driven Web Engineering and the MDA. In: Web Engineering: Modelling and Implementing Web Applications, Rossi, G., O. Pastor, D. Schwabe and L. Olsina (Eds.). Chapter 12, Springer, London, UK., ISBN-13: 978-1-84628-923-1, pp: 353-382.

Robinson, W.N. and H.G. Woo, 2004. Finding reusable UML sequence diagrams automatically. IEEE Software, 21: 60-67.

Rouille, E., B. Combemale, O. Barais, D. Touzet and J.M. Jezequel, 2013. Integrating software process reuse and automation. Proceedings of the 20th Asia-Pacific Software Engineering Conference, December 2-5, 2013, IEEE, Bangkok, Thailand, pp: 380-387.

Rubin, J. and M. Chechik, 2012. Combining Related Products into Product Lines. In: Fundamental Approaches to Software Engineering, De Lara, J. and A. Zisman (Eds.). Springer, Berlin, Germany, ISBN: 978-3-642-28871-5, pp: 285-300.

Sommerville, I., 2011. Software Reuse. In: Software Engineering, Sommerville, I. (Ed.). 9th Edn., Chapter 16, Addison-Wesley, Boston, MA., USA., ISBN-13: 9780137053469.