

TV Time
Analysis and Design Document
Student: George Cimpoides
Group: 30432

TV Time	Version: <1.0>
	Date: <dd/mmm/yy>
<document identifier>	

Revision History

Date	Version	Description	Author
<dd/mmm/yy>	<x.x>	<details>	<name>

TV Time	Version: <1.0>
	Date: <dd/mmm/yy>
<document identifier>	

Table of Contents

I.	Project Specification	4
II.	Elaboration – Iteration 1.1	4
1.	Domain Model	4
2.	Architectural Design	4
2.1	Conceptual Architecture	5
2.2	Package Design	5
2.3	Component and Deployment Diagrams	5
III.	Elaboration – Iteration 1.2	6
1.	Design Model	6
1.1	Dynamic Behavior	6
1.2	Class Design	6
2.	Data Model	6
3.	Unit Testing	6
IV.	Elaboration – Iteration 2	7
1.	Architectural Design Refinement	7
2.	Design Model Refinement	7
V.	Construction and Transition	7
1.	System Testing	7
2.	Future improvements	7
VI.	Bibliography	7

TV Time	Version: <1.0>
	Date: <dd/mmm/yy>
<document identifier>	

I. Project Specification

Design and implement a client-server application for users to be able to follow their favorite TV shows or find new TV shows. Each TV show has the following information associated: number of seasons and their episodes, date when it was launched, the date when the next episode is going to be run or the date when it ended, and what type of TV show is (e.g. drama, romantic, fantasy etc.).

The application should have two types of users (a regular user, and an administrator user) which have to provide a username and a password in order to use the application.

The regular user can perform the following operations:

- Add a new TV show to their following list
- Search new TV shows
- Mark episodes or seasons as watched
- Choose which TV series from their following list to appear on their profile (e.g. still running, last watched, ended etc.)

The administrator user can perform the following operations:

- CRUD operations on user accounts.
- CRUD operations on TV shows.

In case a new TV show appears on the market the administrator should add it to the database. Also when a new episode of the TV shows followed by the user is appearing in the next 2 days the user should receive a notification.

II. Elaboration – Iteration 1.1

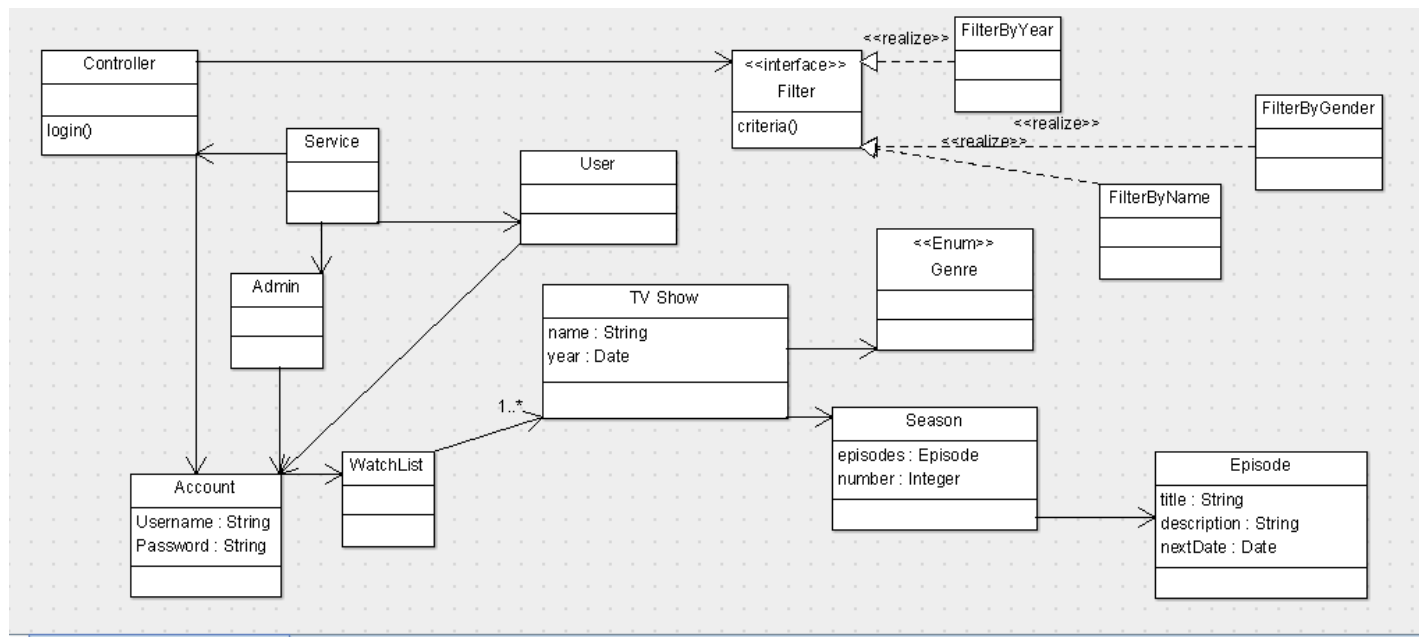
1. Domain Model

On the diagram below the relationships between models are presented. A regular user, or an admin must have an account in order to do any operations.

An account can have a watch-list that is composed of several different TV Shows. A TV Show is composed from a season and a season from multiple episodes, the TV show also has different attributes such as name, year or a genre.

The filter interface is then implemented by different concrete classes and help the user to find a certain TV show by different criteria.

TV Time	Version: <1.0>
	Date: <dd/mmm/yy>
<document identifier>	

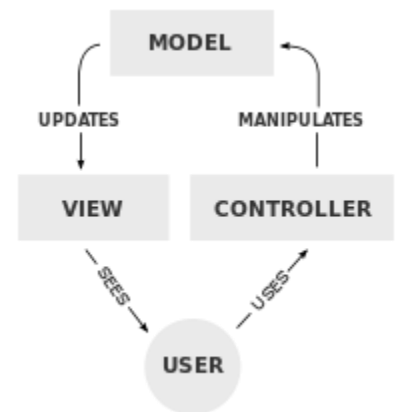


2. Architectural Design

2.1 Conceptual Architecture

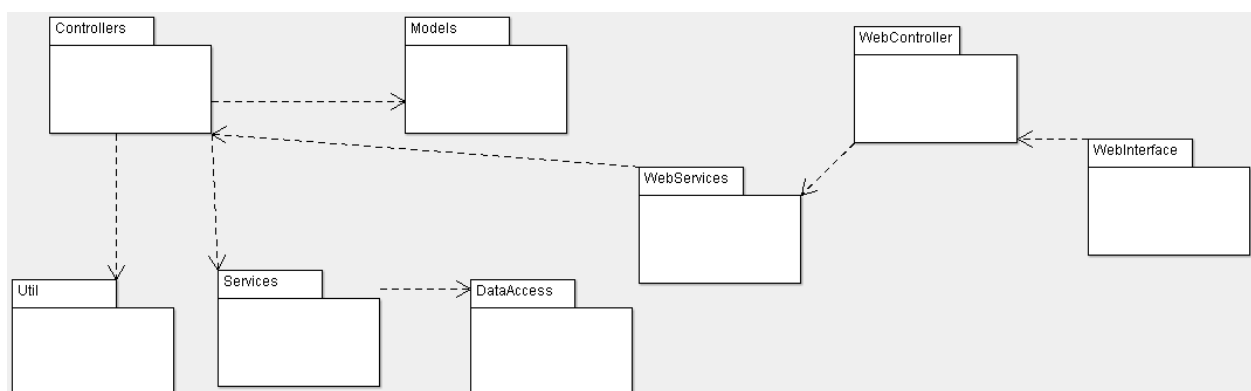
The main architectural pattern used in the project will be the **MVC pattern**.

It divides the application into three interconnected parts in order to separate internal representations from the ways that information is presented to the user, and accepted from the user. The MVC design pattern decouples the view from the controller, the front-end from the back-end, allowing for efficient code reuse and parallel development.



And the second one will be the **layer architectural pattern**. This pattern groups classes into functional layers. A layer is a group of components that are reusable in similar circumstances.

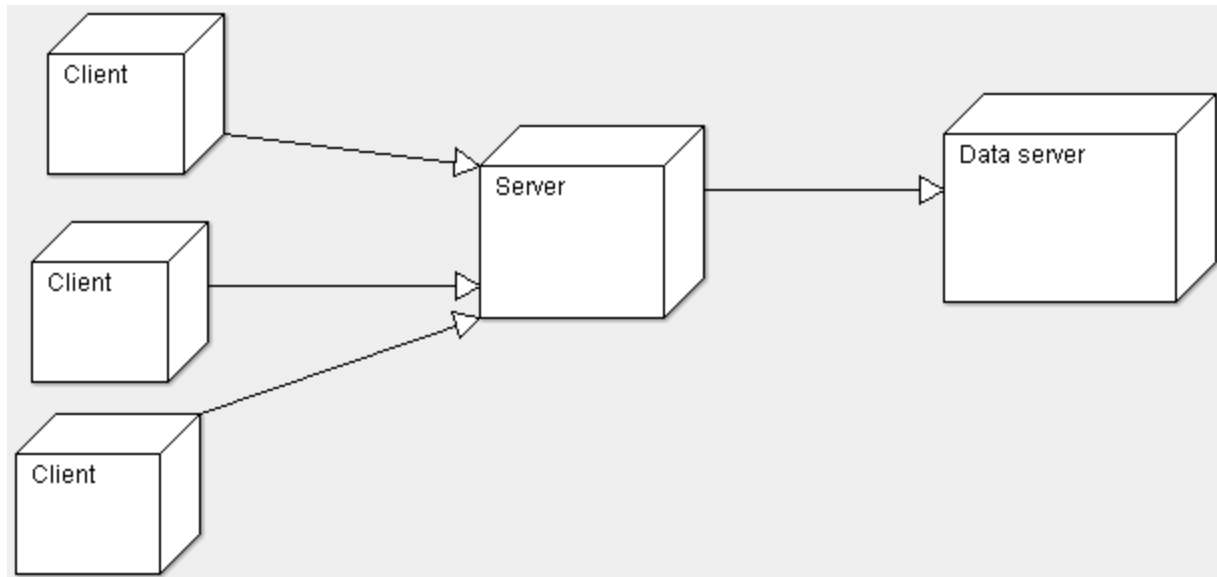
2.2 Package Design



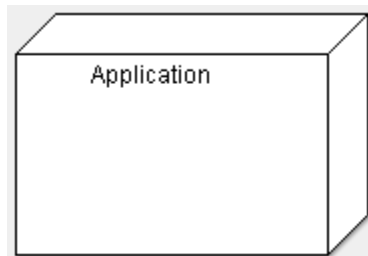
TV Time	Version: <1.0>
	Date: <dd/mmm/yy>
<document identifier>	

2.3 Component and Deployment Diagrams

2.3.1 Deployment Diagram



2.3.2 Component Diagram



III. Elaboration – Iteration 1.2

1. Design Model

1.1 Dynamic Behavior

[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]

1.2 Class Design

[Create the UML class diagram; apply GoF patterns and motivate your choice]

2. Data Model

[Create the data model for the system.]

3. Unit Testing

[Present the used testing methods and the associated test case scenarios.]

TV Time	Version: <1.0>
	Date: <dd/mm/yy>
<document identifier>	

IV. Elaboration – Iteration 2

1. Architectural Design Refinement

[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]

2. Design Model Refinement

[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]

V. Construction and Transition

1. System Testing

[Describe how you applied integration testing and present the associated test case scenarios.]

2. Future improvements

[Present future improvements for the system]

VI. Bibliography