

# CCT College Dublin

## Assessment Cover Page

<b>Module Title:</b>	Advanced Numerical Methods
<b>Assessment Title:</b>	Algorithms, Architecture and Design Patterns Object Oriented Analysis and Design
<b>Lecturer Name:</b>	Sam Weiss
<b>Student Full Name:</b>	Bekezhan Abdykarimov (2020297) Jefferson de Oliveira Lima (2020373) Thiago de Oliveira da Costa (2020005)
<b>Student Number:</b>	2020297, 2020293, 2020005
<b>Assessment Due Date:</b>	19th December 2022 by 23:59
<b>Date of Submission:</b>	26th of December 2022

### Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

# Contents

<b>Problem definition:</b>	<b>3</b>
<b>List of requirements:</b>	<b>3</b>
<b>User story:</b>	<b>3</b>
<b>Software development methodologies:</b>	<b>4</b>
Project analytic structure	4
Work Cronogram:	5
Class diagram	7

# EirVid App

## Problem definition:

The ÉirVid movie rental service will be integrated into RTÉ Player, a video streaming service, in order to allow users to rent movies.

Currently, RTÉ Player does not offer a way for users to rent movies, which limits the options available for streaming content.

The ÉirVid system will solve this problem by providing a platform for users to browse, rent, and watch movies through RTÉ Player.

## List of requirements:

- Allow users to create and log in to accounts using an email and password. This will allow users to access their rental history and recommendations.
- Allow users to browse a list of available movies and view information such as the title, genre, and price.
- Allow users to rent movies by selecting a title and entering payment information.

Initially, movies and prices will be provided in a csv file. This will allow for easy updating of the movie selection and pricing.

- For the prototype, output customer, movie, and price information to the console when a customer is charged. This will serve as a placeholder for a more complete payment system in a future version of the service.
- Rental duration will be 1 minute for each movie. This will be a temporary measure for the prototype, and rental duration may be adjusted in future versions of the service.
- Keep track of which movies a user has rented. This will allow users to easily access their rental history and re-watch movies they have previously rented.
- Recommend the 5 most rented movies of the past 5 minutes to users. This will allow users to discover popular movies and make informed decisions about what to rent.

## User story:

As a user of the ÉirVid service, I want to be able to filter the list of available movies by genre so that I can easily find movies that match my interests.

## Software development methodologies:

The team employed an agile software development methodology in order to efficiently plan and develop the ÉirVid system. This approach allowed us to focus on specific features and functionality and make adjustments as needed. We held regular meetings to discuss progress and make any necessary changes to the plan.

UML diagrams:

Use case diagram:

Class diagram:

We also used user stories to guide our development efforts and ensure that the system met the needs of users.

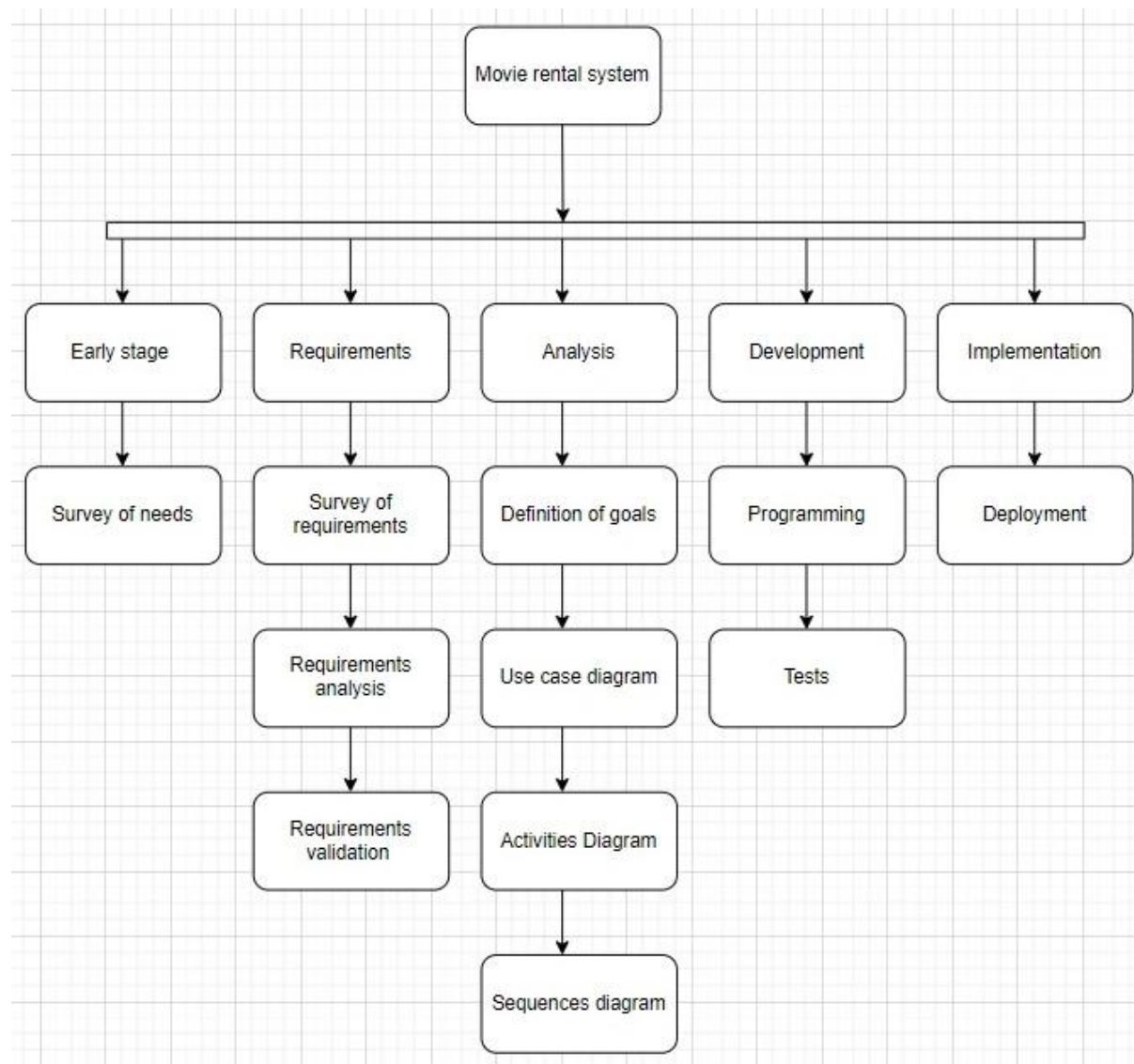
Bekezhan was responsible for creating an analytic structure for a Movie Rental System by following a few steps.

First, we conducted a survey of needs to determine what features the system should have and what the user needs are.

Second, we analysed the requirements, which included defining goals, creating a use case diagram, creating a work timeline, and creating sequence diagrams.

Third, we moved onto the development stage, which included programming and testing. Finally, we moved to the implementation stage, which included deploying the system and conducting a system acceptance test. Overall, this process helped us to create an analytical structure for the Movie Rental System that would be effective and efficient for users.

### Project analytic structure



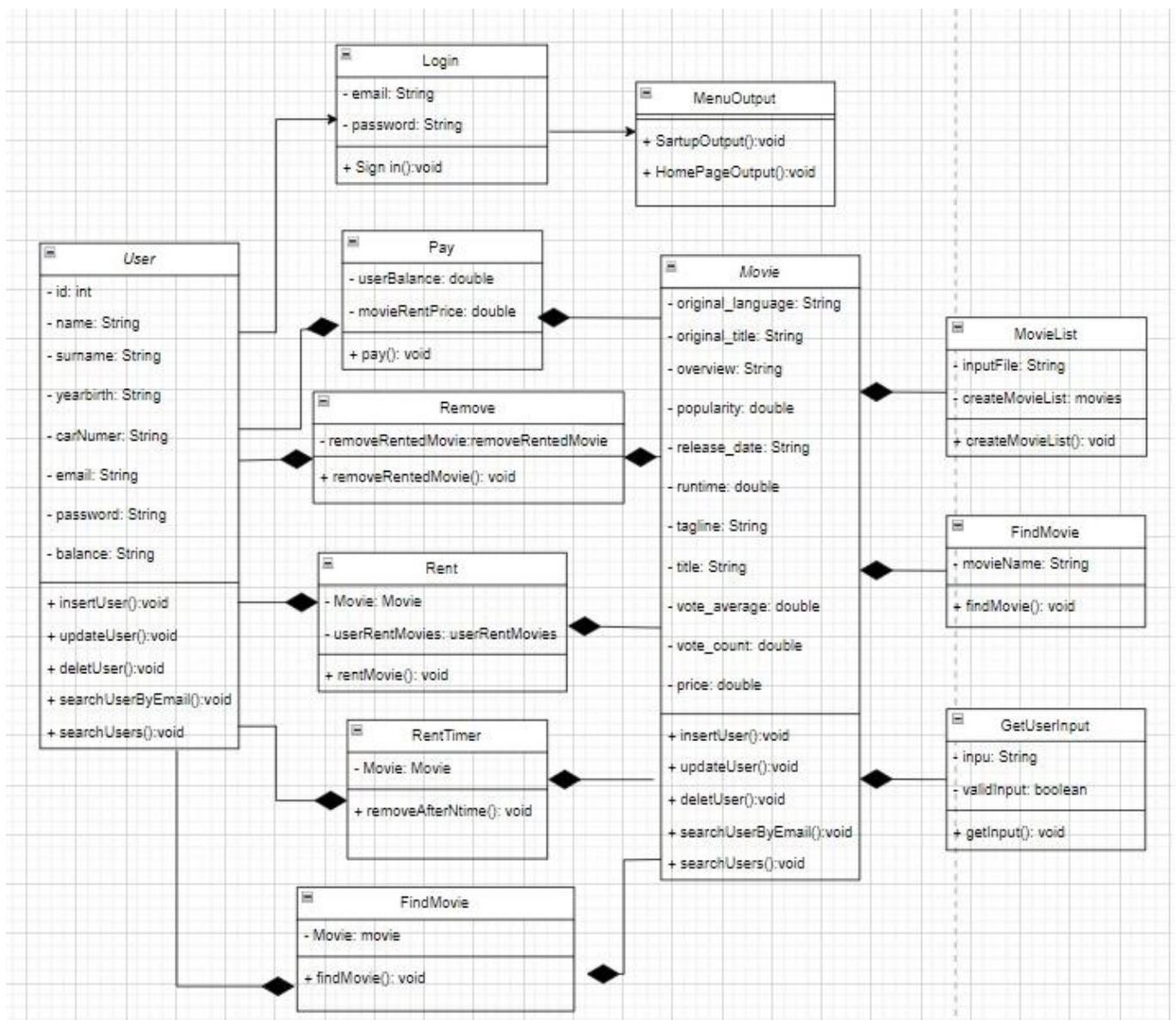
### Work Cronogram:

This was one of the tools that guided and helped us planning the project. Unfortunately some of the tools we started using in the middle of the project but that made us better understand the benefits of starting from them which would have made it a lot easier in terms of clarifying where we were and where we were going to. Jefferson came with the idea of using this Work timeline, the fields was decided in a meeting and was fed from all the integrants.

Work timeline																					
Fase?Month	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10
Survey of needs																					
Requirement specifications																					
Use case diagram																					
Classes diagram																					
Sequencies diagram																					
Coding																					

Jefferson was in charge of the Class Diagram for a Movie Rental System. This system includes a User Object, which can be logged in by the Login Method. Once the user is logged in, the MenuOutput Method will display a menu of options. The User Object will be connected to the following five methods: Pay Method, Remove Method, Rent Method, RentTimer Method, and FindMovie Method. These methods will facilitate the user's ability to rent movies and manage their account. The Movie Object will be connected to the MovieList Method and the GetUserInput Method. The MovieList Method will allow the user to browse movies and select the one they want to rent. The GetUserInput Method will allow the user to enter their payment information. As long all these methods are implemented, the Movie Rental System will work perfectly. This system will allow users to easily rent movies and manage their account with ease.

## Class diagram



## Reference list

<https://stackoverflow.com>

<https://www.geeksforgeeks.org>

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>

<https://www.javatpoint.com/uml-class-diagram>