



# Архитектура на приложението

Мария Павлова

# Разпределение на екип по роли

Име на отбора	Pixel Pixies
Итерация	2

Роля	Студент (имена)
Developer 1	Невена Георгиева Мандева
Developer 2	Мария Веселинова Павлова
QA	Теодора Миткова Пачелийска
Project manager	Иванина Атанасова Иванова
Product owner	Маноела Димитрова Неделчева

# Многослойна архитектура

- Това е архитектура от тип ( N-слойна архитектура) клиент-сървър.
- Избрах я защото:
  - интерфейсът, бизнес логиката, съхранението и обработката на данни са логически разделени на отделни модули.
  - осигурява модел, благодарение на който при разработката ще бъде възможно да се създаде гъвкаво приложение, чиято употреба е многократна.
  - предоставя възможност за добавяне или променяне на отделен слой независимо от останалите, вместо да бъде преработено цялото приложение.
- Архитектурата ще е трислойна:
  - Presentation layer: чрез него ще се репрезентира какво вижда потребителят на брауъра.
  - Application layer: в него се състои бизнес логиката на проекта.
  - Data layer: съхранява данните на приложението

# Архитектура- съдържание по леъри и взаимодействия

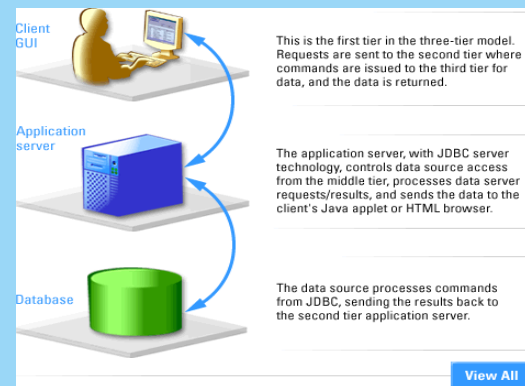
## ➤ Presentation Layer:

Представява интерфейса на приложението. Разработен е с **React**. Потребителят на нашето приложение в случая ще види начален екран, от където ще може да избере категория предмет, по който да гледа даден клип или реши тест, профил екран, на който е изписано името и имейла му, както и всички запазени клипове и тестове до момента.

- От **Presentation Layer** се изпращат заявки към **Logic Layer**.

## ➤ Data layer: Тук се обработват командите, идващи от **JDBC**.

В този слой се съхраняват данните на приложението. Използваме **MySQL**. Базата данни се състои от таблици с полета за потребителско име, имейл адрес, парола, клипове, тестовите по различните предмети и запазването на резултатите на ученика след направата на дадения тест.



## ➤ Logic Layer:

- В този слой се състои цялата бизнес логика на приложението.  
Тук се намират **Rest API** и **End points(Spring Boot)**, чрез които **Logic Layer** комуникира с **Presentation Layer** чрез риквести. След като получи заявката я обработва чрез **JDBC**, изпраща команди до **Data layer** и след като получи данните връща резултатите към **HTML** браузъра.
- Съдържа класовете, функциите и сървисите, които представляват бизнес логиката и позволяват на юзъра да комуникира и управлява приложението.  
Такива класове и функции за нашето приложение са: на потребителите, клиповете, тестовите, категория на предметите, коментари, функции за гледане на клип, поставяне на коментар под дадения клип, даване на лайк или дислайк под даден клип или коментар, решаване на тест.

# Архитектура- Клас диаграма

Diagram.mdj — StarUML (UNREGISTERED)

Format Model Tools View Window Debug Help

g Diagrams

lain — Model

(Basic)

ss

urface

ociation

ected

tion

regation

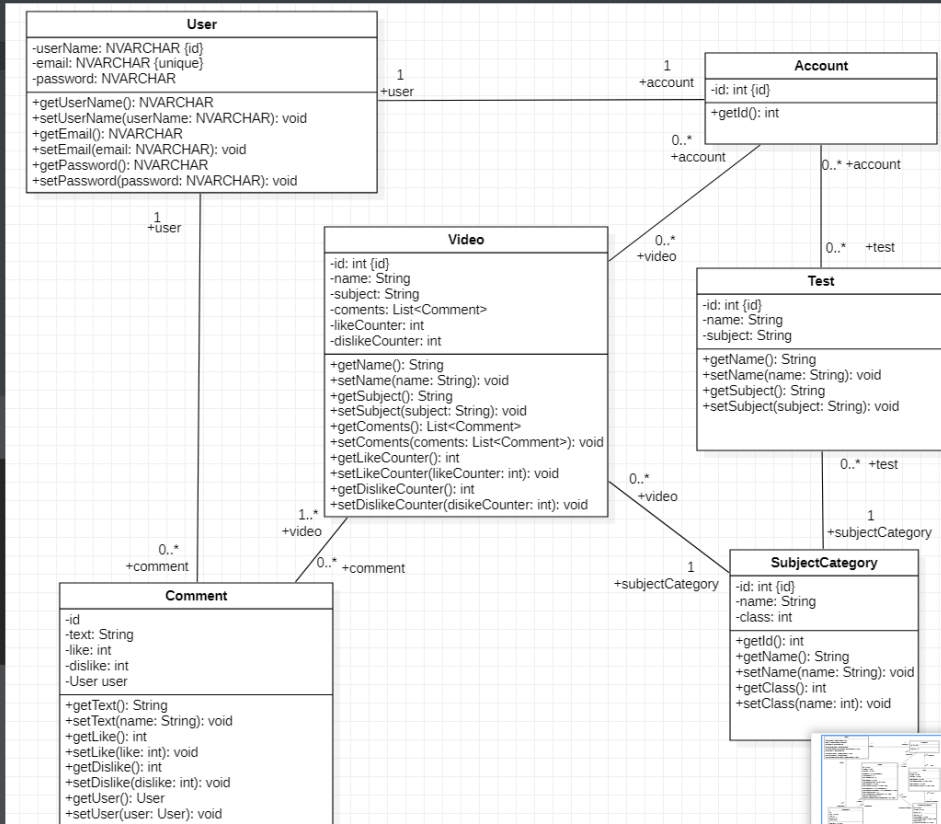
nposition

endency

eralization

urface

tion



Model Explorer



▶ (A

▶ Videos

▶ id

▶ na

▶ su

▶ ce

▶ lik

▶ di

▶ ge

▶ se

▶ ge

▶ se

▶ se

▶ ge

Editors

