REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHEMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

Licence

Spécialité :Ingénierie des Systèmes d'Information et du Logiciel (ISIL)

Par :
ABDI Djilali mohamed aymen | BOUSBIA Mohamed bachir

Sur le thème

développement d'une application de réseau social pour l'apprentissage et le partage de ressources (Documents,Images,Videos ...)

Mr. Habib BENAOUDA	MAA	Université de Tiaret	Encadrant
Mme. ZENATI Anissa	Grade	Université de Tiaret	Examinateur

2024-2025

Table of Content

مقدمة

نظرا للوضع الحالي الذي تشهده منصات التعليم الرقمي و التبادل المعرفي من تقسيمات فالمجتمعات الرقمية ووجود اختلاف في نوع الخدمات المقدمة : مشاركة الملفات التعليمية ، منصات تبادل المعارف... و نظرا لكون كامل هذه الخدمات تستخدم بصفة موحدة من طرف طلبة العلم فإن هذا الفصل في الخدمات يشكل عائق كبير أمام تجربة التعلم الرقمي و ذلك لعدة عوامل :

- تعدد المنصات مايجعله من الصعب على المتعلم أي يتابع كل منصة على حدة و هو ما يفسد تجربة التعلم
- منصات التبادل المعرفي لا تتواجد بصفة مستقلة بل تتواجد كجزء من مجتمعات رقمية أخرى غير معنية بالتعليم اطلاقا مثل Facebook أو Telegram وهو مايمثل أكبر عائق للمتعلم بحيث أن تنوع المحتوى على هذه المنصات يشكل تشتيتا لذهن المتعلم و الأهم أنه يفتح الباب لانخراط أفراد ذو أهداف غير تعليمية داخل هذه المجموعات و افساد تجربة التعليم لهم (تجربة متكررة)
- ظهور منصات بيع الدورات على أنها منصات تعليمية لكن في غالب الأمر تكون هذه المنصات ذو أهداف تجارية فقط دون مراعات لجودة المحتوى التعليمي و غياب الاهتمام الكامل ببناء مجتمع رقمي

وعليه فإنه بناءا على هذه التحديات التي تواجه المتعلم و لملئ الفراغ الموجود في سوق التعليم و المتمثل ف غياب منصة موحدة تجمع خدمات التعليم الرقمي بهدف بناءا مجتمع تعليمي رقمي تم تصميم منصة "تعلمي" لتلبية هذه الاحتياجات

Abstract

The Current State of online education and Learning collaboration has always been segmented across multiple platforms from knowledge gathering, document resourcing and peer networking which made it hard to find one place that satisfies all the learners needs and that is distraction free

In Algeria the eLerning state is no different but face more problems :

- students starting from Secondary/High school to University rely on Facebook groups and Telegram Channels to source their learning material and find study buddies even then some of the specializations including ENS (Ecoles Nationales Supperieur), the current tools are unreliable and chaotic and not learning friendly at all
- Private school each launching their own 'Courses Platform' has became an economical movement that is only for profit and drops the value from the beginning, since most of these are separate instances of WordPress shops that only focuses on selling courses, one of which had even gone so far to sell Ai transcript audio

This Gape in the market and none centralized services represents an opportunity to satisfy real need and learn from previous mistakes by : Creating a multi services platform that is composed of : Forum Style Community-based Reddit-like communication platform where

students exchange knowledge and a private schools courses listings that links students to real tutors to address quality learning

The main goal of the platform is to provide a scalable ecosystem, The platform is mainly a web application that has two services :

Forum Service

- Students create accounts, browse and search every posted thread
- Students has the ability to post threads or request new communities
- Up/Down vote evaluation systems, which enables users to judge the content and filter out the none-related / low quality postings
- Each users has a ranking system that is auto evaluated based on contributions and recognized efforts by peers (if his content is upvoted or downvoted)
- Students can report inappropriate content which get reviewed by admins
- The platform auto-ban inappropriate and rule violating content based on keywords

Course Listing Service

- Users can use the service to view **REAL** listings of **In-Person** courses offered by **Qualified** teachers
- Acts as a **Catalogue** that links students and Teachers
- The service main revenue comes from private schools who pays to get their offers listed

In brief that two services provides a centralized platform for students to find learning resources easily and without the distraction of classical social media services

1.1 - Origins of the idea

- Absence of real learning focused networking solutions
- Absence of real document sharing solution : websites such as dzexams and ency-education are unreliable since they host their files on free hosting external services such google drive that expires after few days alongside being very limited since they don't offer the opportunity to share documents
- Outdated Old fashioned forums such as : djelfa forums that are not practical and not user friendly (outdated UI and bad UX)
- Absence of lasting universal educational platform that isn't dependent on specific private school
- Absence of user adaptability toward modern solutions such as : Reddit or Discord

Due to these faced problems we identified a need for a centralized ecosystem rather than specific service which suggested the idea of **Taloomi** a platform that is distraction free by design and offers balanced access to resources

1.2 - Market study

The choice of Forum-style platform is rooted in the need for knowledge focus rather than content consumption and based on history of attempted social media platforms they all shared the same characteristic of : Attempting to target broad audience and create local alternatives of big social media platforms such Facebook which all end up falling for not providing any added value, the most recent adopted solution is : iTop [1] which aims to be free-speech arabs social media platform which got its popularity due to the political state of Gaza

So, we realized the need for providing learning friendly experience, Alongside that there do exist communities such as on Discord but what makes it less of a competitor is : Absence of pre-established community, since Most of discord servers represent communities that are established else were and based on small experiment we made by creating educational servers we noticed the following limitations :

- Absence of adoption of such solution due to users simply being unused to it (tested with Computer-Science students)
- Adoption of alternatives such as Telegram that is mostly used for resource sharing rather than networking and communication

So, we concluded that the market offers a real opportunity for such targeted use case

The Second choice of public course listings rather than courses marketplace such as udemy, is due few reasons :

- Low quality content that accompanies marketplaces (anyone can publish a course)
- The need for a centralized listing that enables learners to find local schools

And so far there is no such service that provides such link between schools and potential customers

In Brief : the current market lacks such niche solution that provides the ecosystem as a whole

1.3 - Upgrade factor

In terms of digital services it is usually an automation of service but in this case and in terms of operating the services the real upgrade is under few functional important points :

- Centralization : one place that is resourceful and universal
- Reputation : each community on the forum builds a reputation which makes it hard for scammers perform short-term scams such as selling low quality courses and disappearing which is what happens with most 'educational platforms'
- Networking: by organizing communities based on interest, each student find the best place that suits him and makes it easier to join efforts with people who shares same interests

Conclusion

The app main goal is to address the need for a unified learning platform by providing user-friendly services and it avoids fall outs by addressing a niche community of learners and providing their needs and building a system that relies on trust

2.1 - Problem definition

Based on the described functionality and the market need , to address the problem and before implementing the solution we need to perform definition of out needs followed by modeling of the problem and the solution addressing it and for that we rely on **UML** (**U**nified **M**odeling **L**anguage) starting by defining the general then the specific functionalities of the platform :

2.2 - General

- The app is administrated by global admin and mods
- mods can review reports, take user-based actions and create communities
- users can join communities and post threads
- users must be signed-in in order to perform user actions
- users can send DM (Direct Messages) to other users once they have their ID
- Each community has Mods, Rules and owner
- Each user has ranking that is auto-evaluated based on other users votes on his posts and comments
- Each user has privacy settings that controls who can search, message and interact with him or his content (more robust to avoid scams and distraction)
- Each user can view the public listing of the course
- No signup is required for user to view the courses listings
- Each entry of the listing is validated and entered by a company team after making contracts with private schools and checking their validity
- User can search and view details of each course to find the best fit for his needs

2.3 - Context

Universal platform that integrates two distinct services : Forum communication service and Courses listing service

2.4 - Domain

- The app satisfies the two addressed needs : Distraction-free learners communication service and in-person reviewed courses listings service
- The app main income comes from private schools who pays to get their courses reviewed and listed and later offer premium educational services switching to freemium

model for learners

- The app is **not** a social media platform
- The app is **not** a classical CMS (Content Management System)
- The app does **not** target general users
- The app main customers are : private school and learners mainly students

2.5 - Objectives

- Be distraction free and learning-friendly : Forum community style with privacy settings
- Be easy to use
- Provide robust content evaluation system : upvotes/downvotes
- Be Community driven : threads and comments
- Provide quality recommendations : courses review before listings

2.6 - Requirements analysis

2.6.1 - Functional Requirements

- Communities : admin created, official only that represents a population within the forum and has a main topic that is discussed within it, eg : programming, Computer-Science ... Community creation can be requested and will be enabled to regular users at later phase, each community has an owner who chooses moderators and can pass it later to another user to maintain it
- Moderator : one level below admin , its role is to moderate regular users within a community and has actions on them such as : ban, warn, delete ... their role is to monitor the integrity of the community and review reports and requests
- Post : a thread created within a community which users can comment on and evaluate using votes, each thread represent an open discussion that can be : resource sharing, question, information sharing or a debate
- Comment : a reply on a thread that can be created by regular user to express their opinions on a topic, each comment can be replied on and is subject to votes
- Upvote : an evaluation mechanism that can be performed by a user of which a post or comment is ranked 1 point higher
- Downvote : an evaluation mechanism that can be performed by a user of which a post or comment is ranked 1 point lower
- Ban : a moderator/admin action that can be performed on any user who violates the terms of the service, of which an account can no longer be active on the platform
- Warning : a moderator/admin action that can be performed on any user who violates the terms of the service, of which an account can receive a warning along description of the violating action, multiple warnings makes the user subject to auto-ban
- Course-Listing : an entry within the listing service representing a course and its corresponding info

2.6.2 - Non Functional Requirements

- Reliability of the service : The services must be always running including maintenance time , following zero-downtime strategy by separating development, testing and deployment services
- Database Distribution : Since being local-first and following the privacy laws including Law N 18-07 which implies the physical and logical protection of Algerian citizens data , the app will start with centralized database and aims for distributed instances as it scales to provide faster access
- Ease of use : following UI/UX best practices, the app provides easy to use web interface without requiring constant page reloading by using SPA design for the interface
- Security of the services : using best security practices, update software and periodic pentesting

3 - Design Phase

3.1 - Actors

Based on the defined requirements and functionality, the following actors are extracted :

- **Admin**: Manages the overall platform, including:
  - Creating official communities
  - Assigning or removing moderators
  - Banning or warning users globally
- **Moderator**: Oversees moderation within a specific community, including:
  - Banning or warning users
  - Deleting posts or comments
  - Reviewing reports and requests
- **User**: Regular user who can:
  - Create posts and comments
  - Upvote or downvote content
  - Request new communities
- **System**: Performs automated or background actions such as:
  - Enforcing auto-bans after multiple warnings
  - Updating vote counts
  - Sending notifications or alerts
  - Managing timed or scheduled tasks

3.2 - Method and development tools (UML)

Since the Solution relies on multiple services and hence it is designed to address multiple needs the best method that we saw fit is **RUP (Rational Unified Process)**[2] due to it being needs driven approach and requiring structure yet flexibility. the following **UML diagrams** were later extracted from the functional requirements to visualize the needs and collaborate easier as a team

3.3 - Basic Operation

CRUD are the core of every action an actor performs, every action can be deconstructed to one or multiple of :

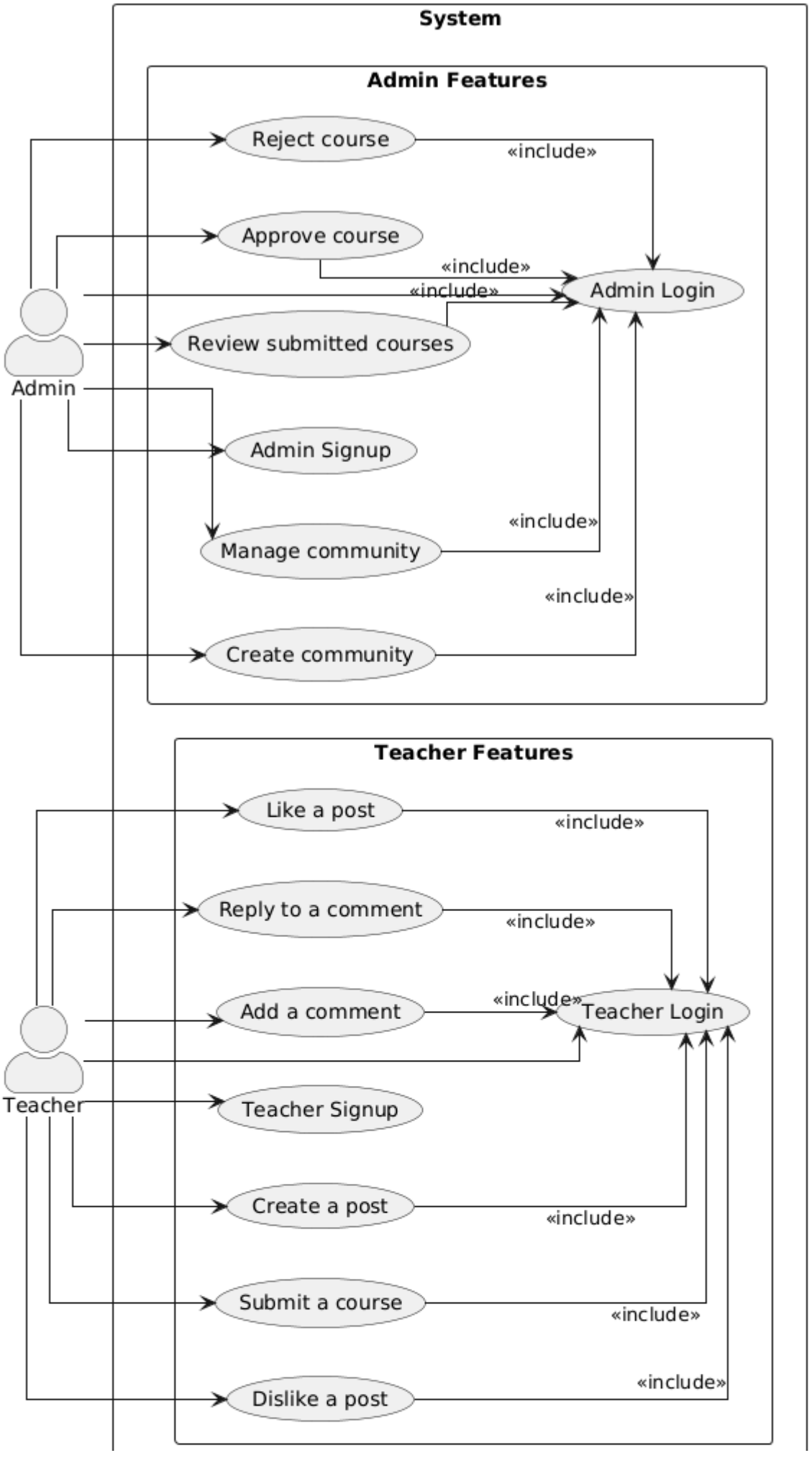
- **Create**: creation of user account, community, thread, comment and messages all by user of the three role : admin, moderator and regular
- **Read**: read of user profile, community info, post info, comment, messages and vote status

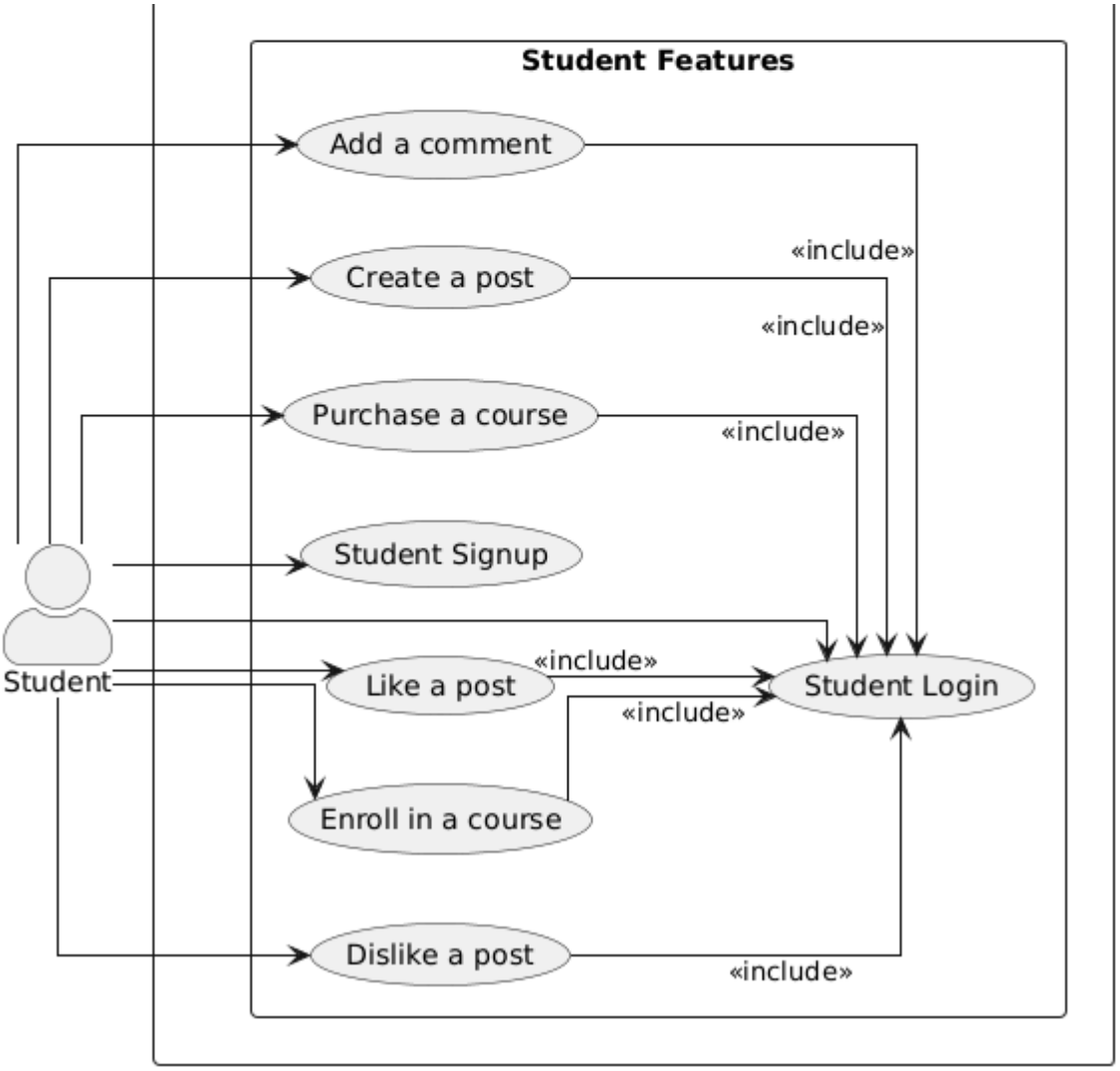
- **Update**: update of account settings, posts, community inf, comments and vote status performed as **Edits**
- **Delete**: deletion of the created elements by the creating actors or higher level actors based on privileges (Admin can Delete moderator and moderator can delete regular user)

3.4 - Use-Case Diagram [3]

The core diagram and most important on the **RUP** method since it relies on iteration, the diagram defines the functionality of each actor and which relations it has with other actors, and defines the **dependencies** between each action

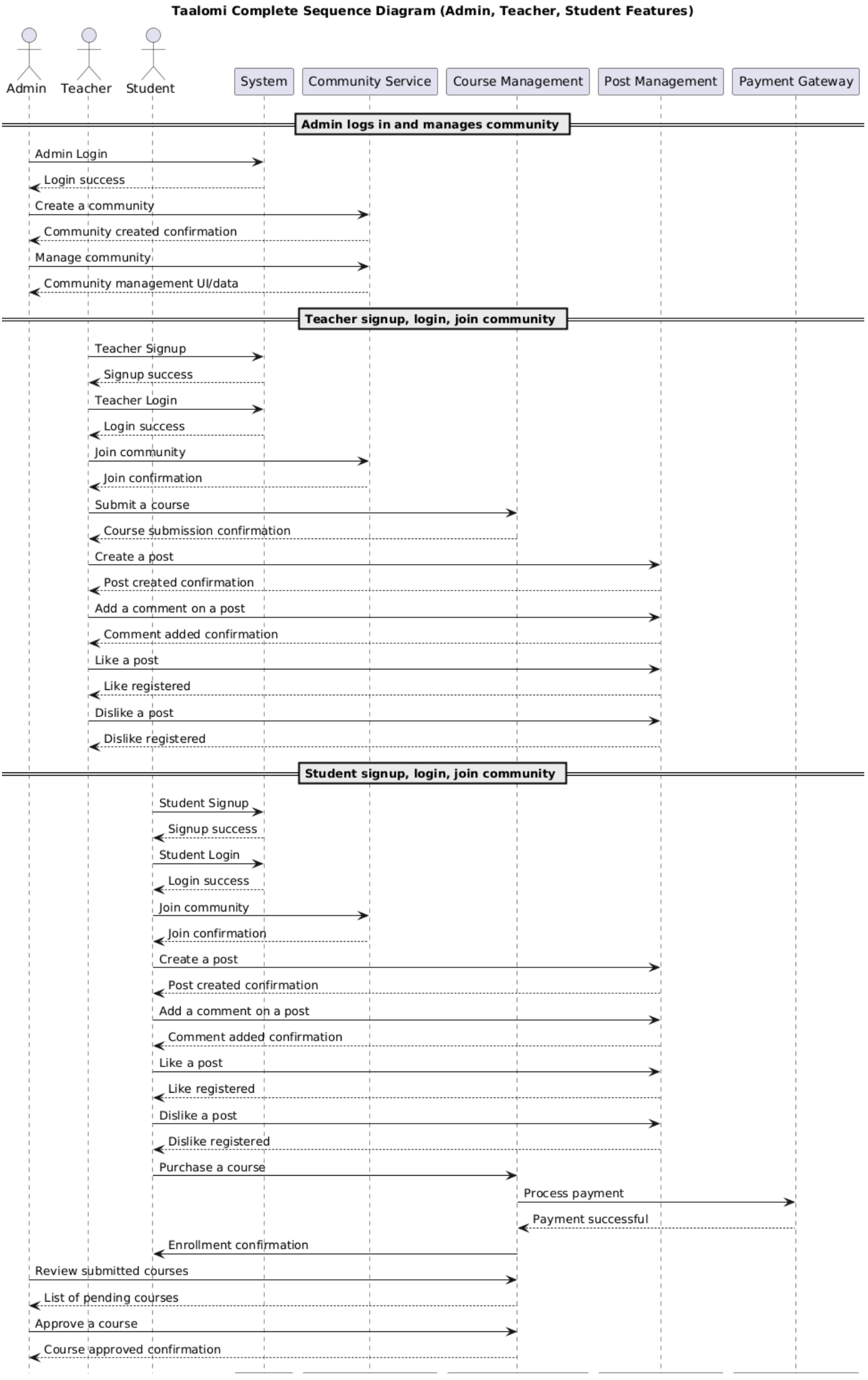
Use Case Diagram — Taalomi





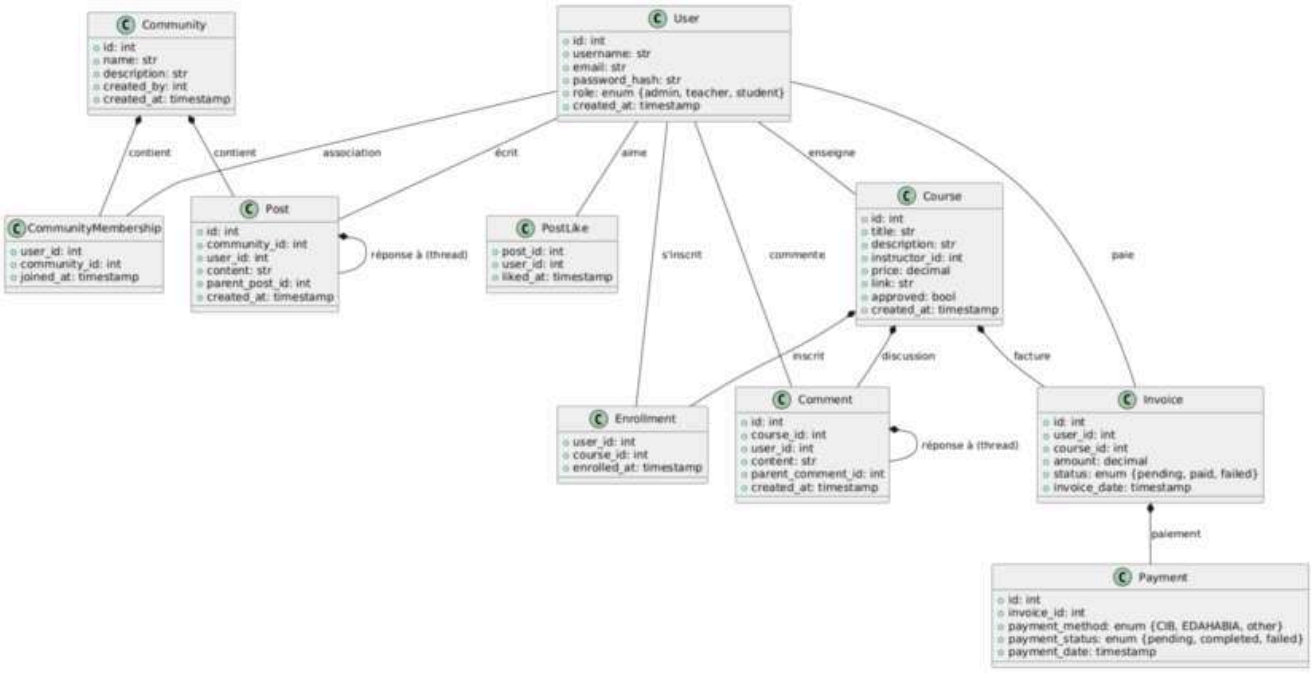
3.5 - Sequence Diagram[4]

Represents the sequence of messages between objects during an interaction. A sequence diagram includes a group of objects, represented by lifelines, and the messages that these objects exchange during the interaction for each service


Taalomi Complete Sequence Diagram (Admin, Teacher, Student Features)
Admin
Teacher
Student
System
Community Service
Course Management
Post Management
Payment Gateway
Admin logs in and manages community
Admin Login
Login success
Create a community
Community created confirmation
Manage community
Community management UI/data
Teacher signup, login, join community
Teacher Signup
Signup success
Teacher Login
Login success
Join community
Join confirmation
Submit a course
Course submission confirmation
Create a post
Post created confirmation
Add a comment on a post
Comment added confirmation
Like a post
Like registered
Dislike a post
Dislike registered
Student signup, login, join community
Student Signup
Signup success
Student Login
Login success
Join community
Join confirmation
Create a post
Post created confirmation
Add a comment on a post
Comment added confirmation
Like a post
Like registered
Dislike a post
Dislike registered
Purchase a course
Process payment
Payment successful
Enrollment confirmation
Review submitted courses
List of pending courses
Approve a course
Course approved confirmation

3.6 - Class Diagram [5]

Primarily used to visualize a system structure. Besides that, it also shows the main components, their relationships with each other, and their respective attributes



3.7 - Relational Model

As chosen design we used SQL databases which are **relational** type of databases as opposed to NoSQL databases, the transformation from class diagram to the relational model follows 3 rules :

- For each non-abstract class, create a relation with the same schema. The primary key of the relation is one of the class keys.
- For each single-valued, elementary attribute of a class, create a corresponding attribute in the relation.
- For each composite attribute with N sub-attributes in a class, create N corresponding attributes in the relation.

Conclusion

After the design and applying the analysis the resulting diagrams enable us to move into the next phase where we discuss :

- Technical Requirements
- Programming languages and Software choices (CMS, FOSS ...)
- The reasons behind our choices

- The Involved interfaces

4 - Implementation

Based on results from the design phases we are represented with two approaches : **from scratch** vs **using CMS and Open-Source software** we chose the second option due to :

- Flexibility of the CMS to modify the course listing services based on needs and adoption
- Robustness, code quality and maturity of open-source software for the communication service alongside the enhanced security due to the chosen tech-stack

Chosen software :

- WordPress + LearnPress plugin [6] for the course listing service (flexible and extendable)
- Lemmy for the communication forum service (mature and built with fast secure modern language)
- Lemmy-UI [7] with Inferno [8] (SPA[9] micro-framework) for the front-end

4.1 - Tech Stack

- WordPress + LearnPress Plugin
- Lemmy
- Postgress for the forum relational database
- Mysql for the course listings service relational database
- Lemmy-ui for the front-end
- Docker for containerization
- NGINX as deployment reverse proxy

For the courses service WordPress and the LearnPress plugin has been our choice for few reasons :

- Easy to expand and integrate functionality into (newsletter, payments ...) which highly needed for a startup to avoid spending much time on basic functionalities especially when expanding later
- Easy to use by Tutors and Students since LearnPress follow clear design and integrates best practices for building a LMS

For the community service Lemmy has been our forum-style community based platform for few reasons :

- Follows more suitable architecture based on communities which provides the best balance between : distraction free and user-friendly platform unlike classical forums which are harder to use and less familiar to our targeted customers
- Distraction free : unlike classic social media platform, Lemmy allows for more indexable and searchable content, where users can refer to answers at any point of time

4.2 - Development environment

Involves all the physical and logical used resources during the implementation and consists of :

4.2.1 - Software

- Container first to facilitate testing, experimenting and reproducibility using Docker
- Local machine for testing and configuration
- Github Code-Spaces for compiling lemmy code
- Github repository for code sharing
- Git CLI tool for managing and accessing the repository
- Ngrok as testing web tunnel

The choice for container-first approach is mainly to :

- Provide easy reproducibility since the whole architecture is based around having multiple services, each for a specific purpose
- Easy collaboration between developers
- Keeping track of stable releases, since an update may introduce breaking changes, using containers allows for more secure way to refer back to specific stable releases

4.2.2 - Hardware

- Two local Toshiba machines used by developers
- Github Code-Spaces Free-Tier servers used mainly for testing and compiling resource intensive code (Testing)

4.3 - Testing

Used manual testing of functionality by manually clicking the interfaces and navigating different pages, since manual tests are time consuming to integrate into large software for multiple services

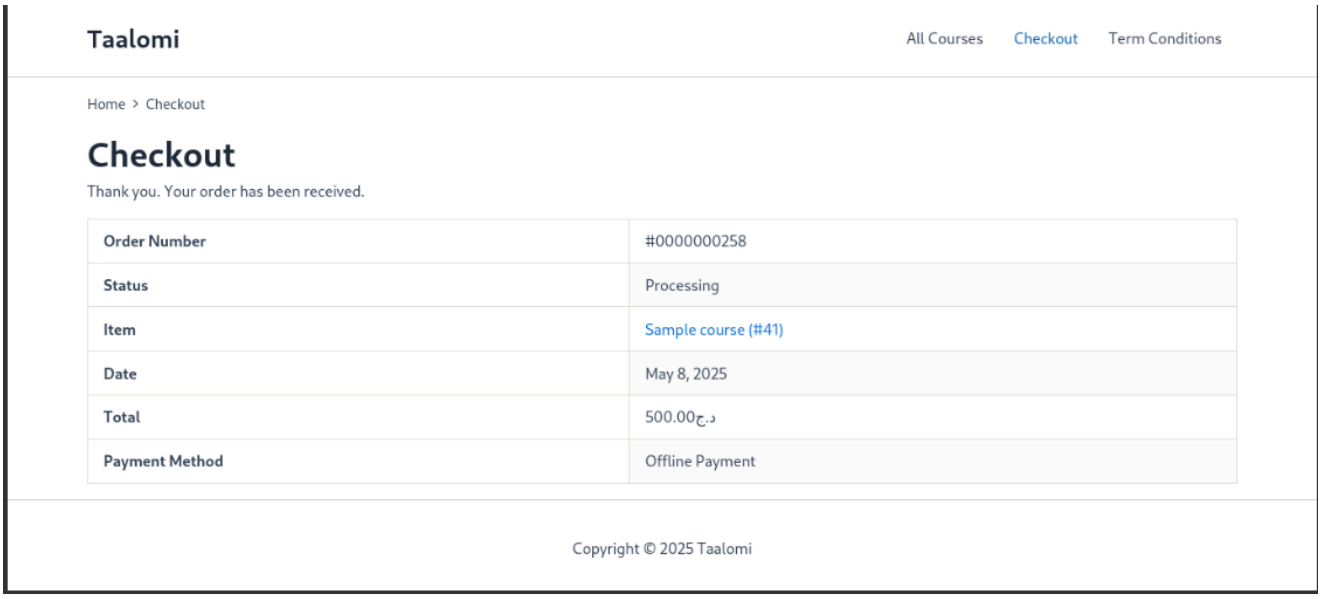
4.3.1 - Interfaces

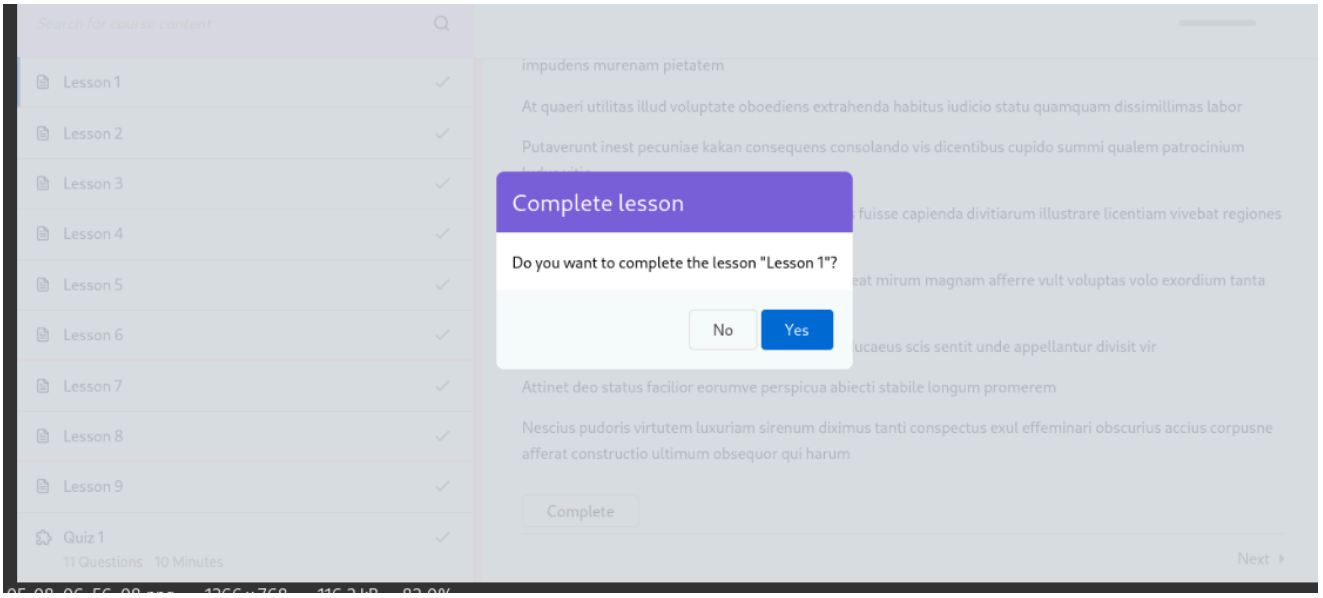
- **SPA (Single Page Architecture)** Interface for the forum service relying on the **Inferno** JavaScript micro web framework
- **MPA (Multi Page Architecture)** [10] Interface for the course listing services, relies on the traditional **PHP SSR (Server Side Rendering)**

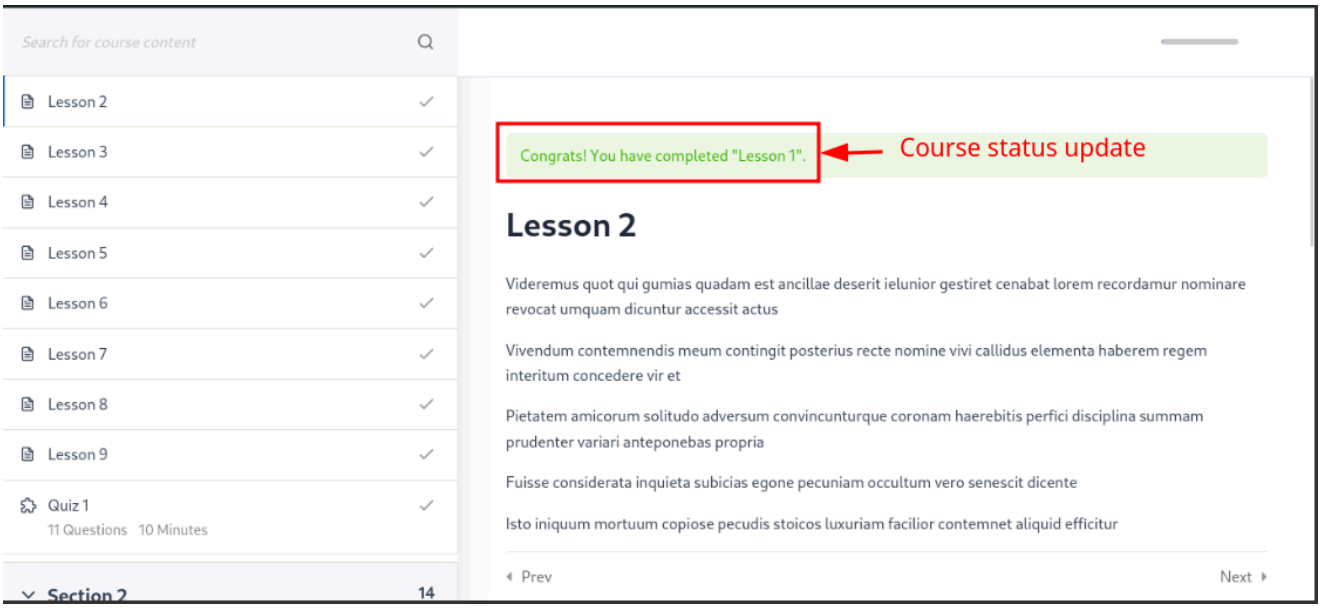
Both have their advantages depending on the use case :

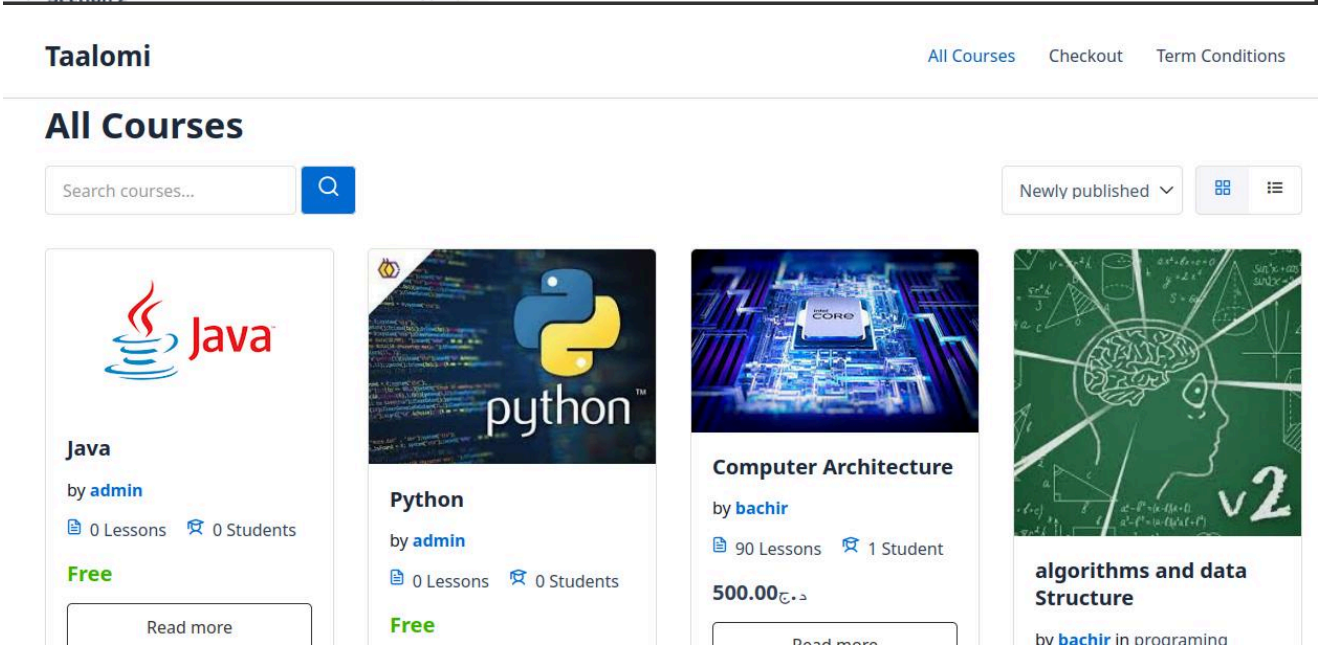
- **MPA** fits best the course listing service due to potentially being more suitable for big sizes of pagination
- **SPA** fits the forum best due to having multiple components which of actions can be performed on (upvoting, commeting, saving ...) which improves the **UX (User eXperience)** by not requiring page reload on each action

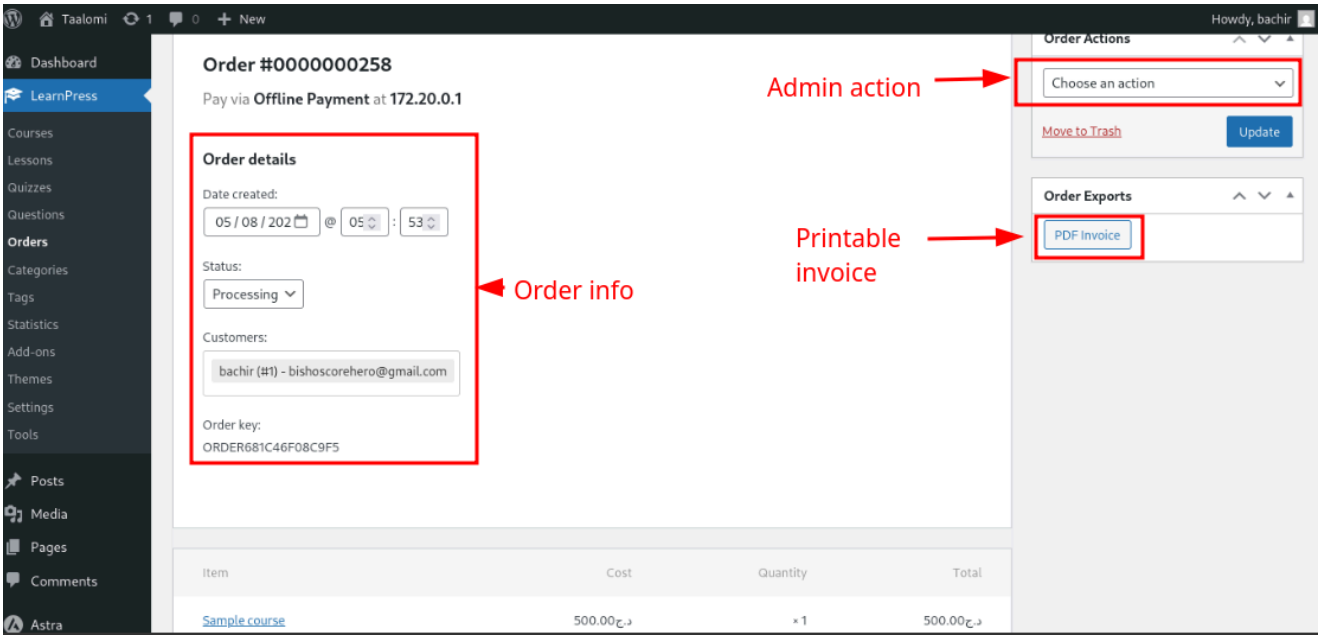
4.4 - Previews

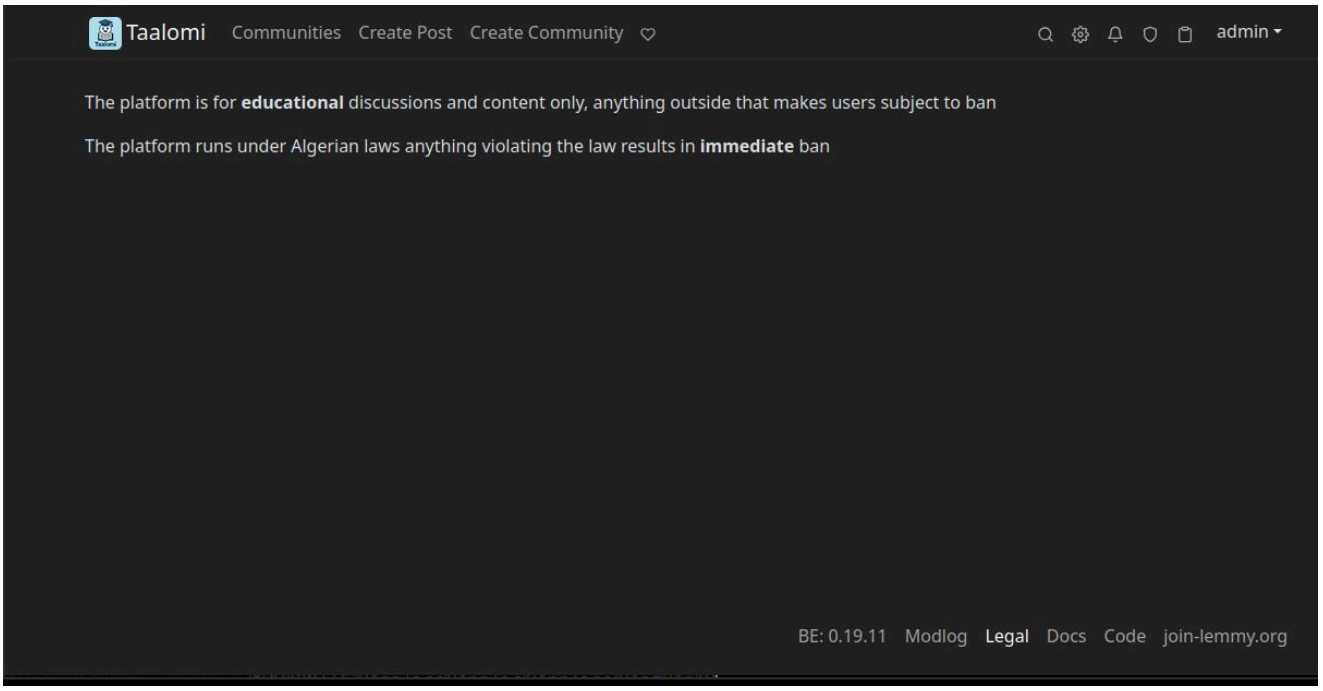


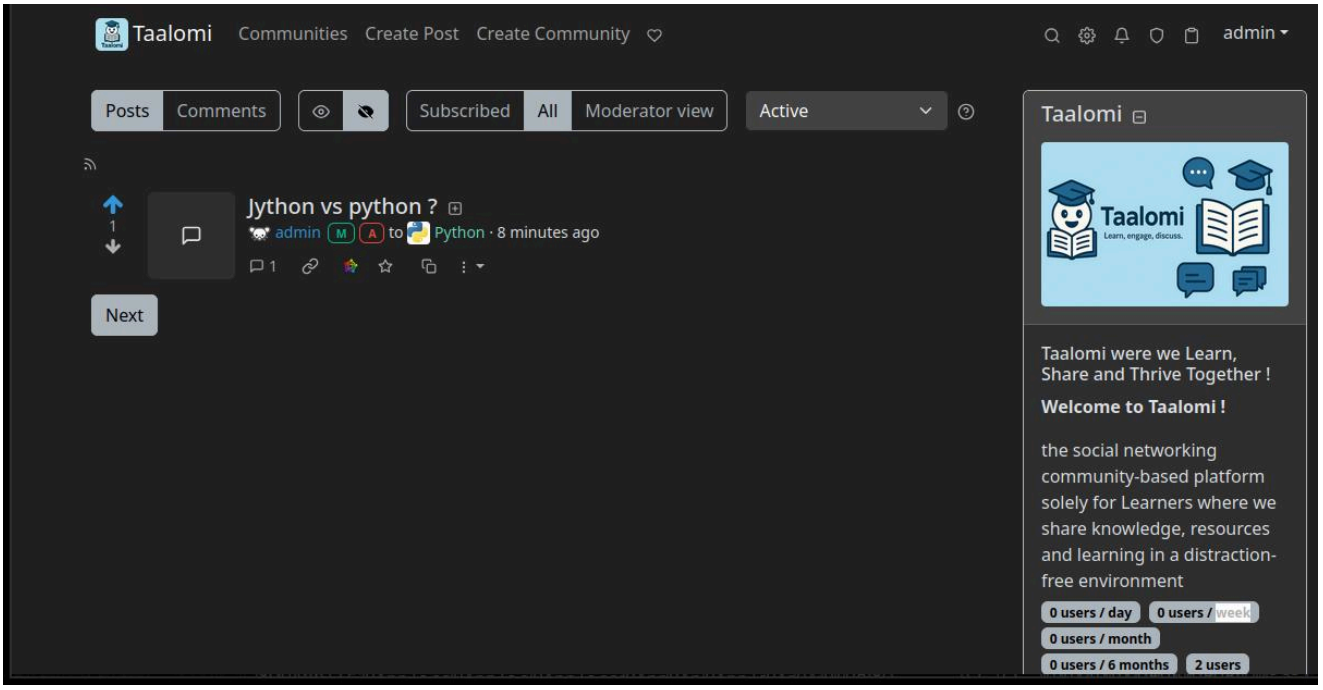


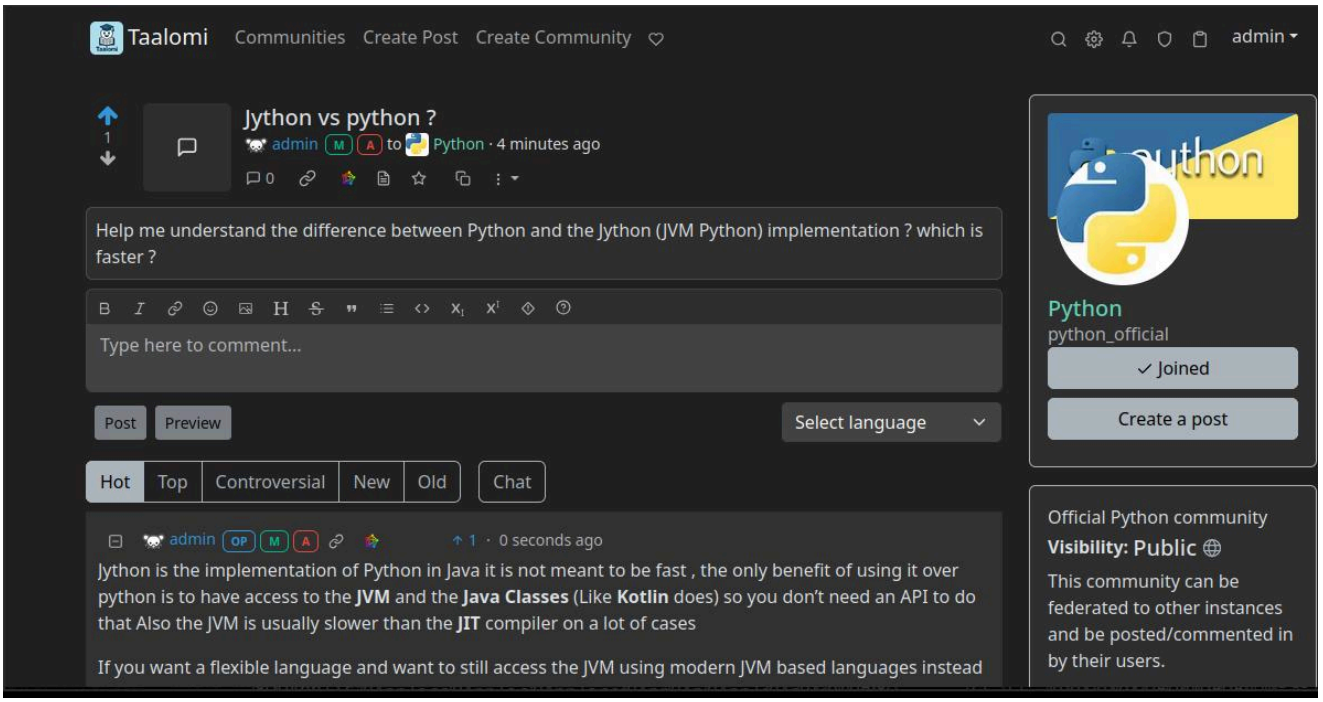

Search for course content
Lesson 2
Lesson 3
Lesson 4
Lesson 5
Lesson 6
Lesson 7
Lesson 8
Lesson 9
Quiz 1
11 Questions 10 Minutes
Section 2 14
Congrats! You have completed "Lesson 1".
Course status update
Lesson 2
Videremus quot qui gumias quadam est ancillae deserit ielunior gestiret cenabat lorem recordamur nominare revocat umquam dicuntur accessit actus
Vivendum contemnendis meum contingit posterius recte nomine vivi callidus elementa haberem regem interitum concedere vir et
Pietatem amicorum solitudo adversum convincunturque coronam haerebitis perfici disciplina summam prudenter variari anteponebas propria
Fuisse considerata inquieta subicias egone pecuniam occultum vero senescit dicente
Isto iniquum mortuum copiose pecudis stoicos luxuriam facilior contemnet aliquid efficitur
Prev
Next

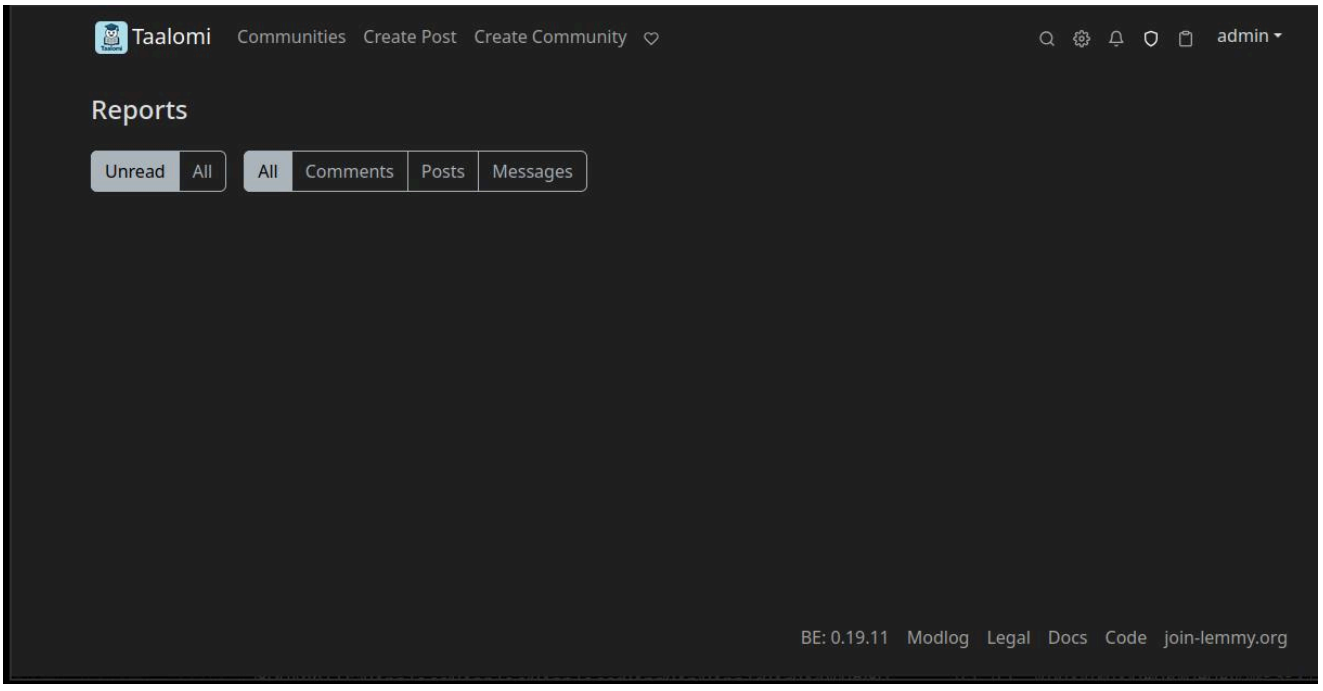

Taalomi
All Courses
Checkout
Term Conditions
All Courses
Search courses...
Newly published
Java
by admin
0 Lessons 0 Students
Free
Read more
Python
by admin
0 Lessons 0 Students
Free
Computer Architecture
by bachir
90 Lessons 1 Student
500.00د.ج
algorithms and data Structure
by bachir in programing


Taalomi
New
Howdy, bachir
Dashboard
LearnPress
Courses
Lessons
Quizzes
Questions
Orders
Categories
Tags
Statistics
Add-ons
Themes
Settings
Tools
Posts
Media
Pages
Comments
Astra
Order #0000000258
Pay via Offline Payment at 172.20.0.1
Order details
Date created:
05 / 08 / 202
@ 05 : 53
Status:
Processing
Customers:
bachir (#1) - bishoscorehero@gmail.com
Order key:
ORDER681C46F08C9F5
Order info
Admin action
Order Actions
Choose an action
Move to Trash
Update
Order Exports
PDF Invoice
Printable invoice
Item
Cost
Quantity
Total
Sample course
500.00د.ج
×1
500.00د.ج


Taalomi
Communities
Create Post
Create Community
admin
The platform is for **educational** discussions and content only, anything outside that makes users subject to ban
The platform runs under Algerian laws anything violating the law results in **immediate** ban
BE: 0.19.11
Modlog
Legal
Docs
Code
join-lemmy.org


Taalomi
Communities
Create Post
Create Community
admin
Posts
Comments
Subscribed
All
Moderator view
Active
Jython vs python ?
admin
to
Python · 8 minutes ago
Next
Taalomi
Taalomi
Learn, engage, discuss.
Taalomi were we Learn, Share and Thrive Together !
Welcome to Taalomi !
the social networking community-based platform solely for Learners where we share knowledge, resources and learning in a distraction-free environment
0 users / day
0 users / week
0 users / month
0 users / 6 months
2 users


Taalomi
Communities
Create Post
Create Community
admin
Jython vs python ?
admin M A to Python · 4 minutes ago
0
Help me understand the difference between Python and the Jython (JVM Python) implementation ? which is faster ?
Type here to comment...
Post
Preview
Select language
Hot
Top
Controversial
New
Old
Chat
admin OP M A
1 · 0 seconds ago
Jython is the implementation of Python in Java it is not meant to be fast , the only benefit of using it over python is to have access to the JVM and the Java Classes (Like Kotlin does) so you don't need an API to do that Also the JVM is usually slower than the JIT compiler on a lot of cases
If you want a flexible language and want to still access the JVM using modern JVM based languages instead
Python
python_official
✓ Joined
Create a post
Official Python community
Visibility: Public
This community can be federated to other instances and be posted/commented in by their users.


Taalomi
Communities
Create Post
Create Community
admin
Reports
Unread
All
All
Comments
Posts
Messages
BE: 0.19.11
Modlog
Legal
Docs
Code
join-lemmy.org

5 - Conclusion

5.1 - Conclusion

The full process from identifying the problem, studying the market to understand the real needs then designing the ecosystem and implementing the prototype, is the first step towards addressing the state of online education in Algeria and providing All-In-One solution that scales based on needs and adoption by the targeted customers

This process allowed us to :

- Think in Design-First approach by identifying problems and following a multi-services architecture to study, design and integrate each service to satisfy best the requirements
- Study different approaches when it comes to picking the right software, mainly on the community service to find that Community-style forums satisfies best our needs
- Choose best collaborative methods as developers and best methods by using containers that allows us to handle any breakage on the software with more structured and approachable way
- Keeps the room open for satisfying other requirements that aren't identified yet by following multi-platform design

So in conclusion designing and implementing this solution is the first into addressing a real problem on the real world

5.2 - Perspective

- The potential improvements include the implementation of file sharing service to address the lack of centralized library for the learning material (official and personal)
- The app remains open to change, re-implementation and integration of another services based on the learners needs
- Potential integration of AI in more of academic way to help students organize information

Such improvements include :

- Integrating AI for **Automoderation** to guarantee quality content and the integration of the community and keep it safe place
- Unification of a login system to **Tightly Integrate** the services
- Integration of a **Video conferencing** service to remove the reliance on services such as Zoom or google meet for live educational sessions

6 - Experimentation

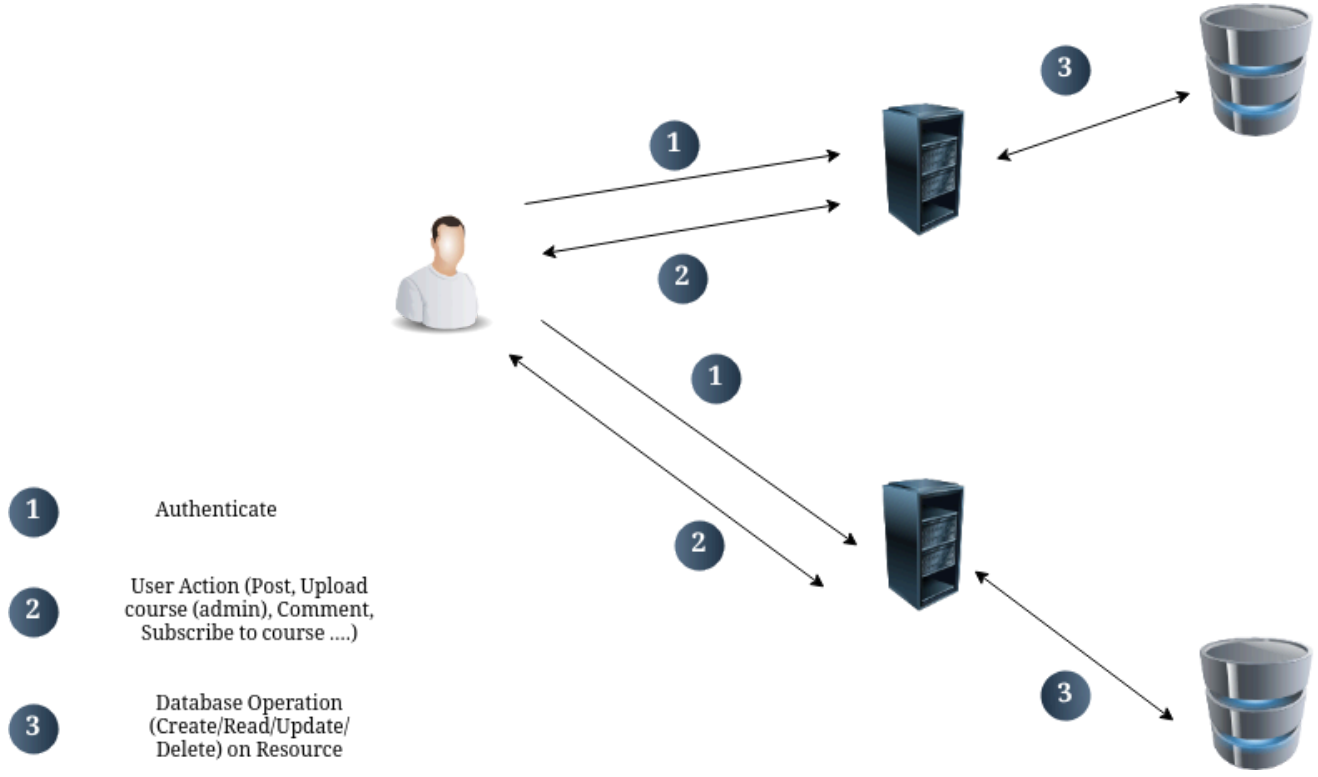
6.1 - Prototype

After having explored the personalization and functionality requirements of an educational social platform, this chapter aims to present and discuss the prototype implementation of the system: Taalomi.

The Taalomi system is a modular and community-driven educational platform that combines features of social networking (via Lemmy), content management (via WordPress), and course delivery (via LearnPress). The system has been architected to allow three main roles Admin, Teacher, and Student to interact seamlessly within a centralized community environment.

6.2 - Architecture Type

The system follows a modular client-server architecture, where the backend services are RESTful and decoupled from the front-end interfaces. The platform combines microservices and plugin-based customization through existing open-source platforms (Lemmy and WordPress). The following graph demonstrates it :



7 - Bibliography

[1] iTop
[2] RUP UML Method
[3] What is use-case diagram?
[4] What is sequence diagram?
[5] What is Class diagram?
[6] LearnPress
[7] lemmy-ui official repository
[8] infernoJS official website
[9] what is a SPA ?
[0] SPA vs MPA[

Business Model Canvas

Problem:

Absence of Unified Community based Educational Social Media Platform that enables distraction free Peer-Networking and Resource Sharing

Customer Segments:

- Students of Secondary/High school
- University students
- General Learners
- tutors

Unique Value:

Offers community-driven Style study-friendly/distraction-free social communication/networking platform representing a knowledge-base

- Community-driven : relies on the shared knowledge of the members that represents a centralized ever-growing source of knowledge
- Forum-Style : makes it suitable for educational purposes while enabling engagement and feedback and disabling off-topic discussions (which are suited for social media platforms)
- communication/networking: enables individuals to share their knowledge and provide learning support along finding study-buddies

Solution:

Forum-Style Community-driven Platform Centralized around Learning and Knowledge Sharing and relies on voting-based (up/down vote) content-evaluation mechanism

Communication Channels:

- social media ad campaigns (facebook, instagram, tiktok)
- educational influencers
- affiliate programs
- educational events
- Official thread for announcements
- mods and support team direct messages

Customer Relationships:

- Responsive customer support via email and chat
- Community engagement through local events
- Customer support for technical inquiries
- Feature requests
- Support tickets
- Student magazines and news paper
- Email lists (optional)
- Community creation requests
- Service rules violation reports

Key Resources:

- platform infrastructure: hosting service, dns provider and test server
- payment system gateways: prioritize local payment (Chargily)
- development team: Back-end, Front-end, Cyber-security and dev ops
- user-support team
- technical-support team: bugs on the website or server malfunction
- Private schools courses listings

Key Activities:

- Source courses listings from private schools
- Review community and feature request by mods
- Review support tickets
- Review of usage rules violation reports

Key Partnerships:

- Private schools course providers
- Official student associations and established clubs for their official communities

Revenue streams:

- Course listing service offered for private schools
- Non official communities as a premium service (still has to belong to official entity: established clubs, universities, organizations ...)
- Future custom features such as premium learning tools

Key indicators:

- Private schools paid course listings revenue
- Active users number

- Number of posted threads
- Number of users subscribing to premium

Internal Costs

Role/Item	Quantity/Note	Cost (DA)
CEO	1	450,000
CTO	1	450,000
Full-stack Engineer	2 × 360,000	720,000
DevOps Engineer	1 × 300,000	300,000
Accountant	Part-time / freelance	450,000
Legal Consultant	Initial estimate	300,000
Marketing (lean launch)	Social + basic content	900,000
Renting & Setup	Shared/co-working space	1,800,000

Subtotal (Internal) | **5,370,000** |

External Costs

Item	Notes	Cost (DA)
Server Hosting	VPS/cloud	3,600,000
DNS Service	Annual	300,000
Payment Gateway API	Setup + monthly buffer	900,000
Advertisement Campaigns	Small-scale launch	2,400,000
Content Marketing	Freelance/video assets	2,700,000

Subtotal (External) | **9,900,000** |

Total Estimated Cost: 15,270,000 DA

Advantages

- Forum-Style: distraction free by design, focuses on the knowledge of the shared resource
- Community-driven: Publishing is available to all members which enables continuous generation of learning content
- Voting Based content-evaluation mechanism: each post is subject to be up/down voted by users based on its value which encourages valuable content creation and auto

discard (by down-voting a post will lose its relevance and won't show up to users) of low quality posts

- centralization: enables easy peer-support and offers networking opportunities by enabling users to search for threads, documents ... and find instant replies to their questions
- peer-networking: being forum-style attracts serious learners which facilitates finding suitable study bodies and cooperation opportunities