Project Brief

⬛ Old 🟧 New from peer review 🟩 New

Document Information

Project name:	**Astratutor**
Date:	30/01/2021
Author:	CS3505 Team 4
Owner	CS3505 Team 4
Version:	3

Approval

Date	Name and Signature

Project Brief

Definition

Background:

Remote learning is very topical at the moment due to COVID-19.

Schools are closed and students cannot meet in person. Some individuals are not receiving any education and may look to other means for assistance.

There is a demand for an online learning solution to replace existing offline learning

Main Goal:

Create an all-in one platform for e-tutors to provide students with one on one lessons for various subjects

Desired Outcomes:

Simple sign up process for both students and tutors with simple verification of tutor credentials.

Tutors can provide a headshot, description and their qualifications to teach a subject.

Tutors can register under a subject and provide their price & availability

Students can find a tutor for a given subject and arrange a time for a one on one class.

An online classroom with file sharing, whiteboard, screen sharing and webcam streaming features.

Tutors are paid upon completing a class.

Tutors can pay out their balance to a bank account

Constraints and Assumptions:

The project length is 8 weeks

WebRTC Is required for video streaming

WebRTC web support isn’t consistent.

An external STUN/TURN server may be required for punching NAT/relaying for WebRTC

We assume tutors will provide sufficient documentation during the signup process.

Linux is required to host the backend and serve the web content

The system is containerized using Docker and will be deployed using Docker Compose

	The external payment processor will be Stripe The database will be PostgreSQL The backend language will be Go lang The frontend framework will be React (with TypeScript) A classroom will support at minimum 2 webcam feeds and 1 screen feed
Interfaces:	We will interface with our payment processor Stripe via their API. Our backend will call Stripe API endpoints when payment is required.
Project Approach:	We will use the Scrum agile framework for tracking our progress We will use daily stand-ups via voice/slack message to report our sprint progress Weekly scrum meetings will be held to: ● track sprint progress (weekly) ● have an overview of the product backlog with the product owner ● communicate feedback from the product owner We will hold sprint planning meetings to select product backlog items for our sprint and decide on a sprint goal A Kanban board will be used to track the Sprint Backlog progress. After each sprint we will review the current project progress

Project Brief

Outline Business Case

Online learning is becoming increasingly more prominent due to recent circumstances such as Covid-19, lockdowns, etc. This has meant that remote learning platforms have become a requirement for effective learning. Once the pandemic ends there will still be a strong need for online platforms due to increased public awareness of such learning methods.

By building a service that connects students with tutors and provides an online space for facilitating learning, we could take a 16-33% cut of any fees paid for tutorship based on market practices.

Our only costs are:

- Hosting
- Maintenance
- Moderation
- Marketing

We are aiming to avoid the dependencies we saw with other platforms we researched:

- The project is scoped to be subject-agnostic, we don't place a dependency on a particular curriculum, any subject can be supported (some competitors hyper-focus on particular curriculums in their implementation)
- We are also avoiding the term 'grinds' (while "grindsapp" is being used as a development codename) to prevent the project being HIberno-centric and avoid placing limits on the future scaling of the project outside of Ireland

Hopefully by doing this our target audience remains broad and accepting of anyone looking for a tutor regardless of their curriculum.

By building an all-in-one system (with a video classroom), we have the full power to improve the learning experience rather than relying on external services like Zoom

The market can be competitive as many solutions exist (MyTutor, FirstTutors, Skooli, Chegg, JumpAGrade, ClassHub) however there is large room for expansion. While the project scope is defined as above, new features/potential stretch goals could include:

- Marketplace for notes
- Pay-per-assistance
  - e.g. request a tutor for assistance in solving a problem for a small fee (1-2 euro)
- Arranging in-person tutorship (when COVID-19 has subsided)

Project Brief

Key Stakeholders

Major Stakeholder	Notes
Product Owner	
Scrum Master	
Development Team	

Project Objectives

	Target	Tolerance
Scope	Accounts System Classroom System Invoice System Tutor Request System Notification System	
Time	6 weeks	1 week
Cost	5 FTE over 7 weeks	5 FTE over 6 weeks
Quality	Intuitive user experience	Functional UI
	Video streaming	480p minimum
	Quickly find suitable tutorial time	Provide functionality to coordinate time slot for a lesson between a student and a tutor
	Verify qualification of Tutors	Vetting process where tutor must upload proof of qualifications
	Accessible to users with disabilities	Although the accessibility of the platform to users with disabilities is still something we need to discuss and consider, we want to follow the best practices for web design to provide accessibility to users who may be

		colourblind or deaf and use screen readers. Something for us to consider is having categories for tutors who specialize in teaching users with disabilities.
Risks	WebRTC might experience peering issues because of NAT resulting in a hindered user experience.	Some devices may not be supported and will need to rely on a TURN server.
	Time-overrun due to difficulty implementing classroom system	Federate to an external service (i.e schedule meetings over Zoom via Zoom API)
	Ensuring a fair payment system	Ensuring a fair payment system will be a challenge, our current strategy revolves around not letting a student book a tutor until they've entered verified payment details, once they book a lesson an invoice will be generated at that time so that a tutor can't change the price of a lesson and have that affect the students expectation with regards to price, once the lesson has ended the student will be charged the invoice. Note: If a student doesn't show up they will still be charged for the lesson. We are looking into possibly requiring a minimum attendance time for a tutor such as being there for 95% of the hour which should account for any unexpected disconnections but still ensure that a tutor showed up for the lesson. We will also provide a payment dispute system if a student or tutor has any concerns.

Benefits	All in one platform for e-learning	Find a tutor, schedule a time slot, provide a online classroom with whiteboard,file sharing and video streaming
	Increase tutor availability exponentially as they can do back to back sessions online due to a robust scheduling system.	

Project Management Team

Role	Reports to	Appointee
Product Owner	H.O.D	Jason Quinlan
Scrum Master	Product Owner	Rotated by developer per week
Graphic Designer **Go Backend Developer** **React Frontend Developer** **Documentation Developer** **Code Monkey**	Scrum Master	Eric Moynihan
UI/UX Designer **React Frontend Developer** **Go Backend Developer** **Documentation Developer**	Scrum Master	Aleksei Ivanov
Go Backend Developer **Documentation Developer**	Scrum Master	Raymond Reynolds
React Frontend Developer **Go Backend Developer** **Documentation Developer**	Scrum Master	Oisin Canty

Go Backend Developer	Scrum Master	James Cotter
Documentation Developer **WebRTC Developer**		

Project MoSCoW

Must Have

- List of subjects available
    - Tutor can sign up to tutor a subject
        - Description of your teaching methods
        - Price per hour
    - Student can see a list of tutors for a subject
    - Tutor can stop tutoring a subject
- List of tutors
    - Can be filtered based on subject
    - Student can request a lesson from a tutor
        - Request a time slot
        - Description of what they want to go over
- Sign up as Tutor
- Sign up as Student
    - if under 16 request parents email
- Create and modify profile with:
    - First Name
    - Last Name
    - Description (Tutor only)
    - Qualifications (Tutor only)

- Work Experience (Tutor only)
- Availability (Tutor only)

- List of lesson requests
    - As a tutor...
        - See lesson requests students sent you
            - Accept or Dey
            - Reschedule (Suggest alternative time for lesson)
    - As a student…
        - See lesson requests you sent to tutors and their status
            - Accepted
            - Denied
            - Rescheduled
            - Requested
- List of lessons
    - See list of upcoming lessons
        - Subject, Tutor, Date-Time
        - Join button (Greyed out until lesson time +5 minutes before start)
- Join Lesson
    - Enter initial lobby where you can configure microphone/webcam settings
    - Enter lesson
        - mute/unmute functionality
        - hide/show camera
        - share screen
        - upload file
        - chat

Should Have

- Employee accounts

- Admin panel
  - Verify qualifications
  - Add additional subjects
- Tutor/Student can request a subject be added
- Include more sorting/filtering functionality in tutor listing
  - sort on price, rating, etc
  - filter on subject, tag
- Include tags on subjects (e.g. Leaving Certificate, Ireland)
- Add whiteboard to lesson
- Cancel/Reschedule lesson
- Payment system
- Report system
- Payment Dispute system

Could Have

- Ratings for Tutors and Students
- Group lessons
- Prompt scheduling next lesson after lesson complete
- Sell ‘Resources’ (e.g. notes, videos)
- Feedback on previous lessons
- Schedule offline lesson (In real life lessons)

Won’t Have

- Social media sign up

Project Brief

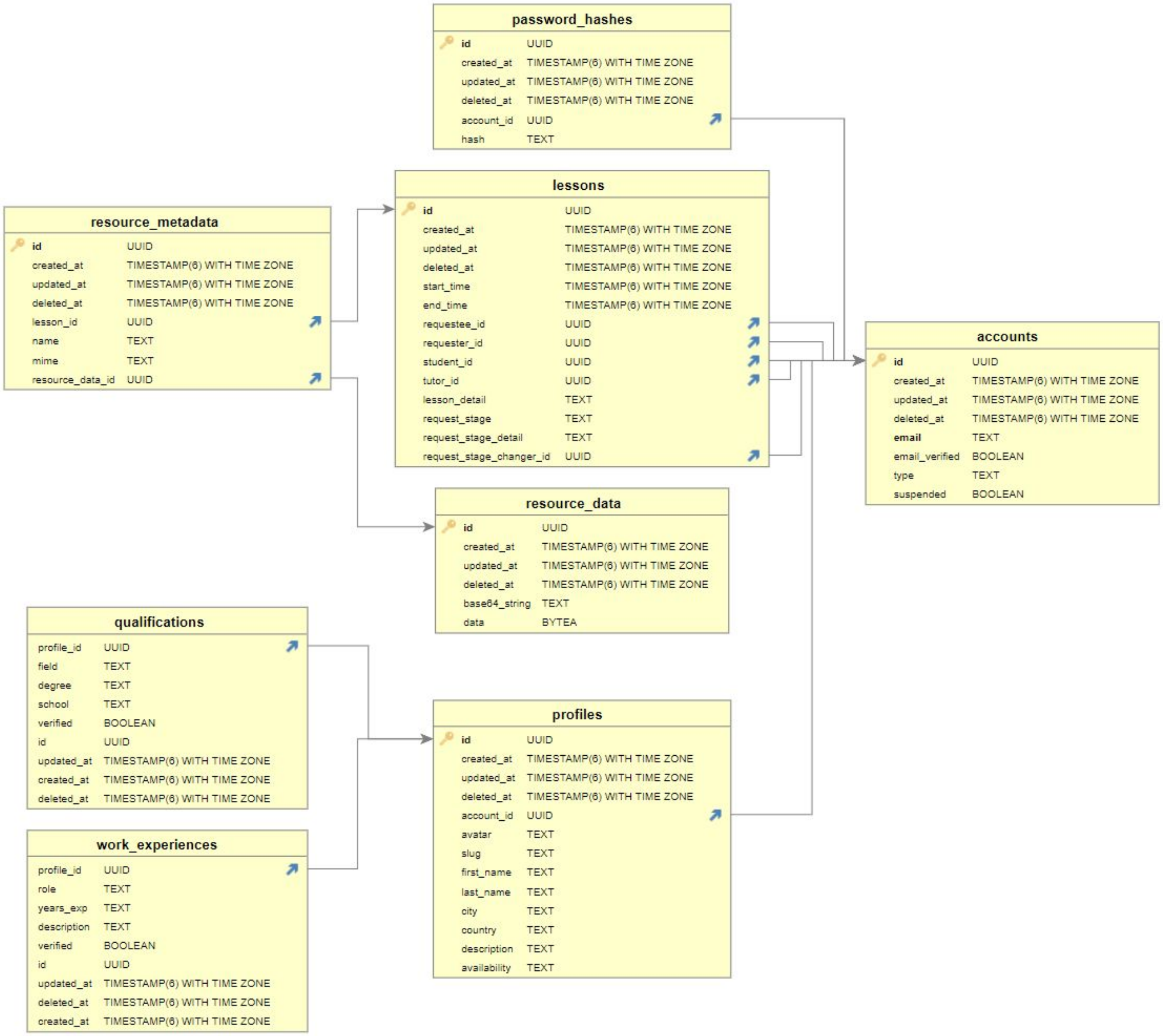
Project Endpoints

Method	Endpoint	Auth	Description
POST	/auth/login	None	Log in
POST	/accounts	None	Create an account
GET	/accounts/{id}	None	Get an account by id
POST	/accounts/{id}/verify	None	Verify account email
GET	/tutors/{id}/profile	None	Get tutor profile
POST	/tutors/{id}/profile	Tutor	Create tutor profile
POST	/tutors/{id}/profile/avatar	Tutor	Update profile avatar
POST	/tutors/{id}/profile/description	Tutor	Update profile description
GET	/tutors/{id}/profile/qualifications	Both	Get all qualifications
POST	/tutors/{id}/profile/qualifications	Tutor	Add a qualification
DELETE	/tutors/{id}/profile/qualifications/{id}	Tutor	Delete a qualification
GET	/tutors/{id}/profile/work-experience	Both	Get all work experience
POST	/tutors/{id}/profile/work-experience	Tutor	Add work experience
DELETE	/tutors/{id}/profile/work-experience/{id}	Tutor	Delete work experience
POST	/tutors/{id}/profile/availability	Tutor	Set availability
GET	/tutor/{id}/lessons	Tutor	Get list of lessons
POST	/tutor/{id}/lessons	Student	Request a lesson
GET	/tutors/{id}/billing/payouts	Tutor	Get payout history
POST	/tutors/{id}/billing/payouts	Tutor	Generate payout
GET	/tutors/{id}/billing/balance	Tutor	Get outstanding balance
POST	/tutors/{id}/billing/bank-account	Tutor	Update bank account details

GET	/tutors/{id}/subjects	None	List of subjects taught by tutor
POST	/tutors/{id}/subjects/{id}	Tutor	Begin tutoring subject
POST	/tutors/{id}/subjects/{id}/cost	Tutor	Update cost of lesson
POST	/tutors/{id}/subjects/{id}/description	Tutor	Update lesson description
GET	/students/{id}/profile	Student	Get student profile
POST	/students/{id}/profile	Student	Create student profile
GET	/students/{id}/lessons	Student	Get list of lessons
GET	/students/{uuid}/billing/invoices	Student	Get all invoices
GET	/subjects	None	Get list of subjects
GET	/subjects/tutors?filter=Maths,English	None	Get list of tutors for all subjects, filtered

Project Brief

Project Database UML



password_hashes

id	UUID
created_at	TIMESTAMP(6) WITH TIME ZONE
updated_at	TIMESTAMP(6) WITH TIME ZONE
deleted_at	TIMESTAMP(6) WITH TIME ZONE
account_id	UUID
hash	TEXT

lessons

id	UUID
created_at	TIMESTAMP(6) WITH TIME ZONE
updated_at	TIMESTAMP(6) WITH TIME ZONE
deleted_at	TIMESTAMP(6) WITH TIME ZONE
start_time	TIMESTAMP(6) WITH TIME ZONE
end_time	TIMESTAMP(6) WITH TIME ZONE
requestee_id	UUID
requester_id	UUID
student_id	UUID
tutor_id	UUID
lesson_detail	TEXT
request_stage	TEXT
request_stage_detail	TEXT
request_stage_changer_id	UUID

resource_metadata

id	UUID
created_at	TIMESTAMP(6) WITH TIME ZONE
updated_at	TIMESTAMP(6) WITH TIME ZONE
deleted_at	TIMESTAMP(6) WITH TIME ZONE
lesson_id	UUID
name	TEXT
mime	TEXT
resource_data_id	UUID

accounts

id	UUID
created_at	TIMESTAMP(6) WITH TIME ZONE
updated_at	TIMESTAMP(6) WITH TIME ZONE
deleted_at	TIMESTAMP(6) WITH TIME ZONE
email	TEXT
email_verified	BOOLEAN
type	TEXT
suspended	BOOLEAN

resource_data

id	UUID
created_at	TIMESTAMP(6) WITH TIME ZONE
updated_at	TIMESTAMP(6) WITH TIME ZONE
deleted_at	TIMESTAMP(6) WITH TIME ZONE
base64_string	TEXT
data	BYTEA

qualifications

profile_id	UUID
field	TEXT
degree	TEXT
school	TEXT
verified	BOOLEAN
id	UUID
updated_at	TIMESTAMP(6) WITH TIME ZONE
created_at	TIMESTAMP(6) WITH TIME ZONE
deleted_at	TIMESTAMP(6) WITH TIME ZONE

profiles

id	UUID
created_at	TIMESTAMP(6) WITH TIME ZONE
updated_at	TIMESTAMP(6) WITH TIME ZONE
deleted_at	TIMESTAMP(6) WITH TIME ZONE
account_id	UUID
avatar	TEXT
slug	TEXT
first_name	TEXT
last_name	TEXT
city	TEXT
country	TEXT
description	TEXT
availability	TEXT

work_experiences

profile_id	UUID
role	TEXT
years_exp	TEXT
description	TEXT
verified	BOOLEAN
id	UUID
updated_at	TIMESTAMP(6) WITH TIME ZONE
deleted_at	TIMESTAMP(6) WITH TIME ZONE
created_at	TIMESTAMP(6) WITH TIME ZONE