ASTRATUTOR - Project Brief

OLD - CHANGES FROM REVIEW - NEW

Document Information

Project name:	**Astratutor**
Date:	26/02/2021
Author:	CS3505 Team 4
Owner	CS3505 Team 4
Version:	6

ASTRATUTOR - Project Brief

OLD - CHANGES FROM REVIEW - NEW

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 with a built in classroom.
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. A robust payment system that allows students to pay for a lesson using a debit card. Tutors can pay out their balance to a bank account
Constraints and Assumptions:	The project length is 8 weeks A team of 5 developers 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. This is required to provide secure online payments in a robust fashion
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

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, this is 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’ 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)
- Creating templated lessons that can be used for providing commonly repeated lessons such as ones on King Lear or possible other topics focused on a current curriculum

Key Stakeholders

Major Stakeholder	
Product Owner	
Scrum Master	
Project Lead	
Development Team	

Project Objectives

	Target
Scope	Accounts System - Login/Logout - Register - Authenticate - View tutor profiles Classroom System - Webcam streaming - Voice streaming - Screen share - Whiteboard - Chat Payment System - Let students pay for a lesson - Let tutors pay out to a bank account Invoice System - Let tutors see their received payments - Let students see their previously sent payments Tutor Discovery System - Find tutors - Should allow filtering based on subject, price, etc - Should show all available subjects

	- Show tutor subjects and price Tutor Request System - Request a lesson from a tutor on a specific date and time - Accept/Deny a lesson - Cancel a lesson - Reschedule a lesson Lesson System - Allow tutors to list subjects they take lessons for
Time	6 weeks 2 weeks review/clean up
Cost	5 FTE over 6 weeks
Quality	Intuitive user interface - Should be simple to use - Require minimal learning - Fast Video streaming - 480p minimum - Share at least 2 webcams and 1 screen Quickly find a tutor - Find a suitable tutor for your desired subject - See vetted tutor experience Flexible scheduling - Scheduling should be easy but also flexible to allow for you to reschedule or cancel a subject is an issue arises Accessible to users with disabilities - Should be open to users with visual and aural impairments - Possible consideration for students who require special assistance would be to have tutors who are vetted to teach students with certain impairments - Site should be both colorblind and screen reader friendly
Risks	WebRTC might experience peering issues because of NAT resulting in a hindered user experience. - Devices may require a turn server to relay information

	Time-overrun due to difficulty implementing classroom system - Classroom system is complex and could be difficult to get right, falling back to an existing service could simplify things if needed Ensuring a fair payment system - Ensuring payments are fair and secure is tough - If a lesson gets cancelled a student should get a full refund - If a tutor fails to attend a lesson a student gets a full refund - If a student fails to attend a lesson a tutor keeps payments - Payment is required before the lesson starts - Include dispute system for case-by-case scenarios
Benefits	All in one platform for e-learning - No need to use multiple services with multiple accounts - Tools are tailored to the job at hand Increase tutor availability exponentially as they can do back to back sessions online due to a robust scheduling system. - No need to travel between lessons allowing for more lessons with less time wasted - Create templated lessons allowing for commonly thought lessons to be arranged faster

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 **Classroom UI Developer** **Accounts Developer** **Documentation Developer**	Scrum Master	Eric Moynihan
Project Lead **React Frontend Developer** **Go Backend Developer** **Documentation Developer**	Scrum Master	Aleksei Ivanov
Go Backend Developer **React Frontend Developer** **Documentation Developer**	Scrum Master	Raymond Reynolds
Payments Developer **Documentation Developer**	Scrum Master	Oisin Canty
Classroom Developer **Documentation Developer** **WebRTC Developer**	Scrum Master	James Cotter

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 Deny ✔
- 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 ✔
    - 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 ✔

Add whiteboard to lesson ✔

Cancel/Reschedule lesson ✔

Payment system ✔

Report system - Students can report tutors for violations

Payment Dispute system

Classroom - File sharing

Could Have

Ratings for Tutors and Students ✔

Group lessons - Schedule a lesson with multiple students

Prompt scheduling next lesson after lesson complete ✔

Sell ‘Resources’ (e.g. notes, videos)

Feedback on previous lessons

Schedule offline lesson (In real life lessons) - Similar to scheduling a lesson but also includes a location

Internationalization - Support multiple languages to support a wider range of learners

Pay-per-assistance - A Student can pay a small fee to have a task completed

Won’t Have

Social media sign up

Project Database UML

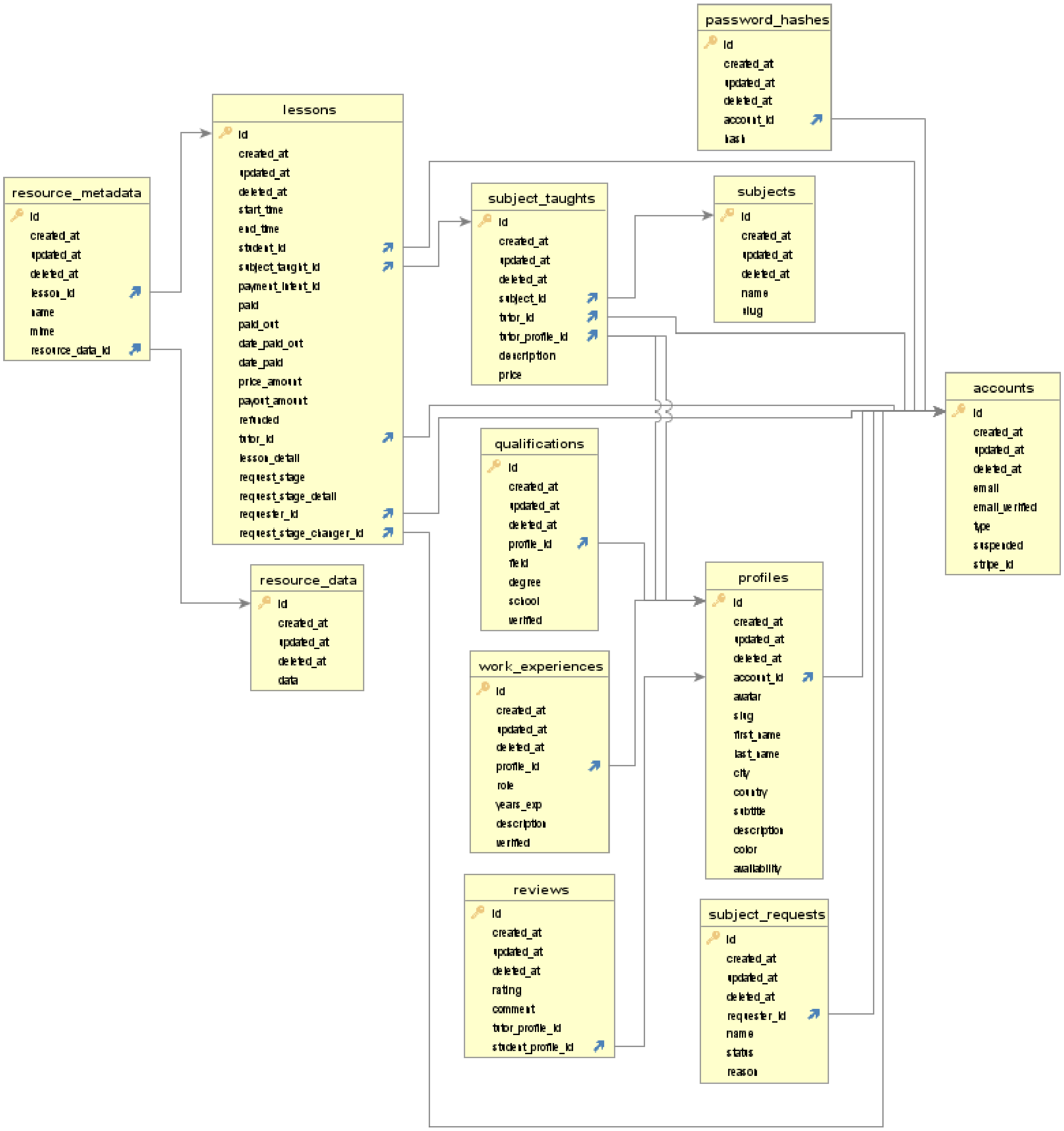

Figure 1; Database Overview

UML provides a run down of how the different entities in our system work together and share relevant information.

Project Architecture

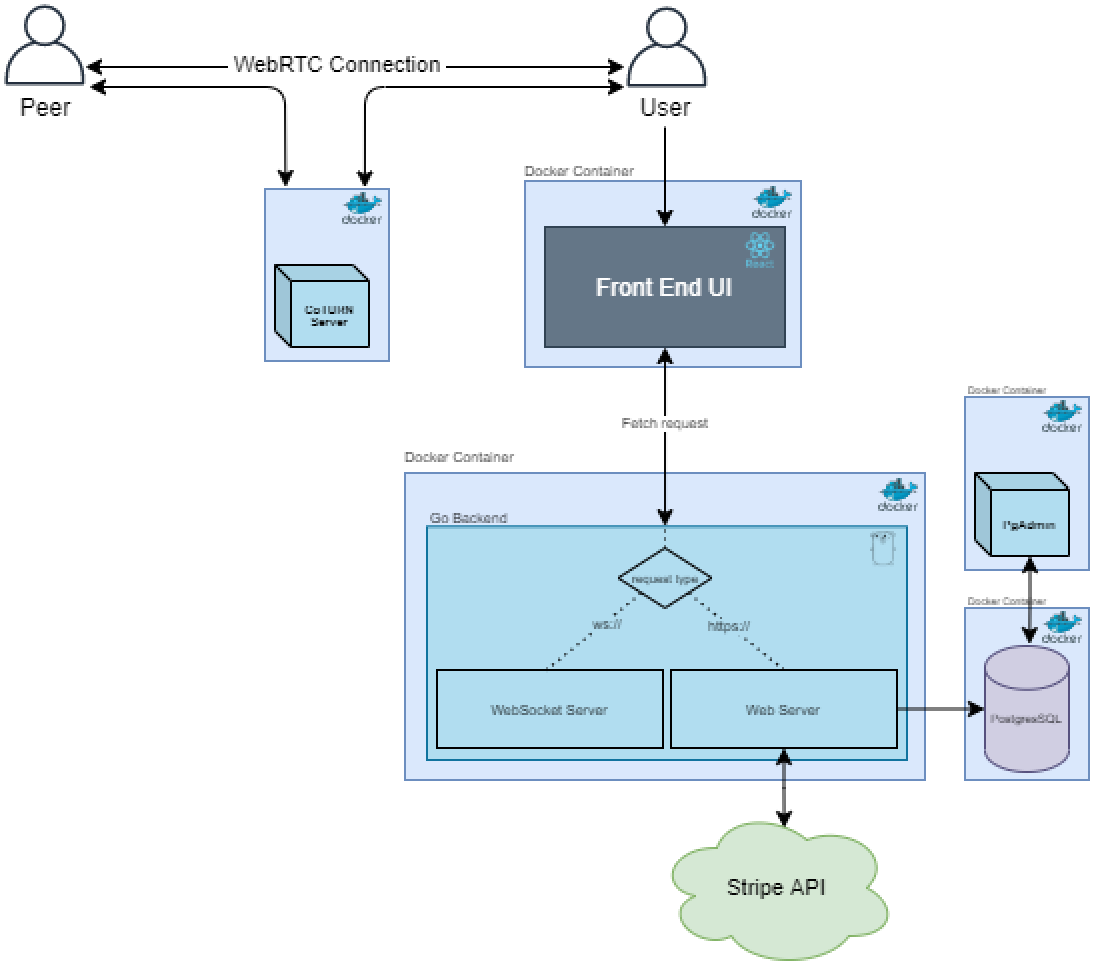

Figure 2; Project Architecture Overview

Using Docker containers we containerized all our applications. When a user is connecting to a peer to establish a connection they first communicate their supported protocols by sending messages over the WebSocket server. Next each peer contacts a STUN server (hosted by Google in our case) to determine their public IP address and NAT type. Depending on their NAT type the peer might establish a connection with the TURN server. Connection information is exchanged between the peers and finally the connection is established directly between the peers or using the TURN server as a relay. We run a PgAdmin instance so that we can administrate our PostgreSQL DB. Our frontend and backend communicate with each other using HTTP Rest requests where json data is sent back and forth.

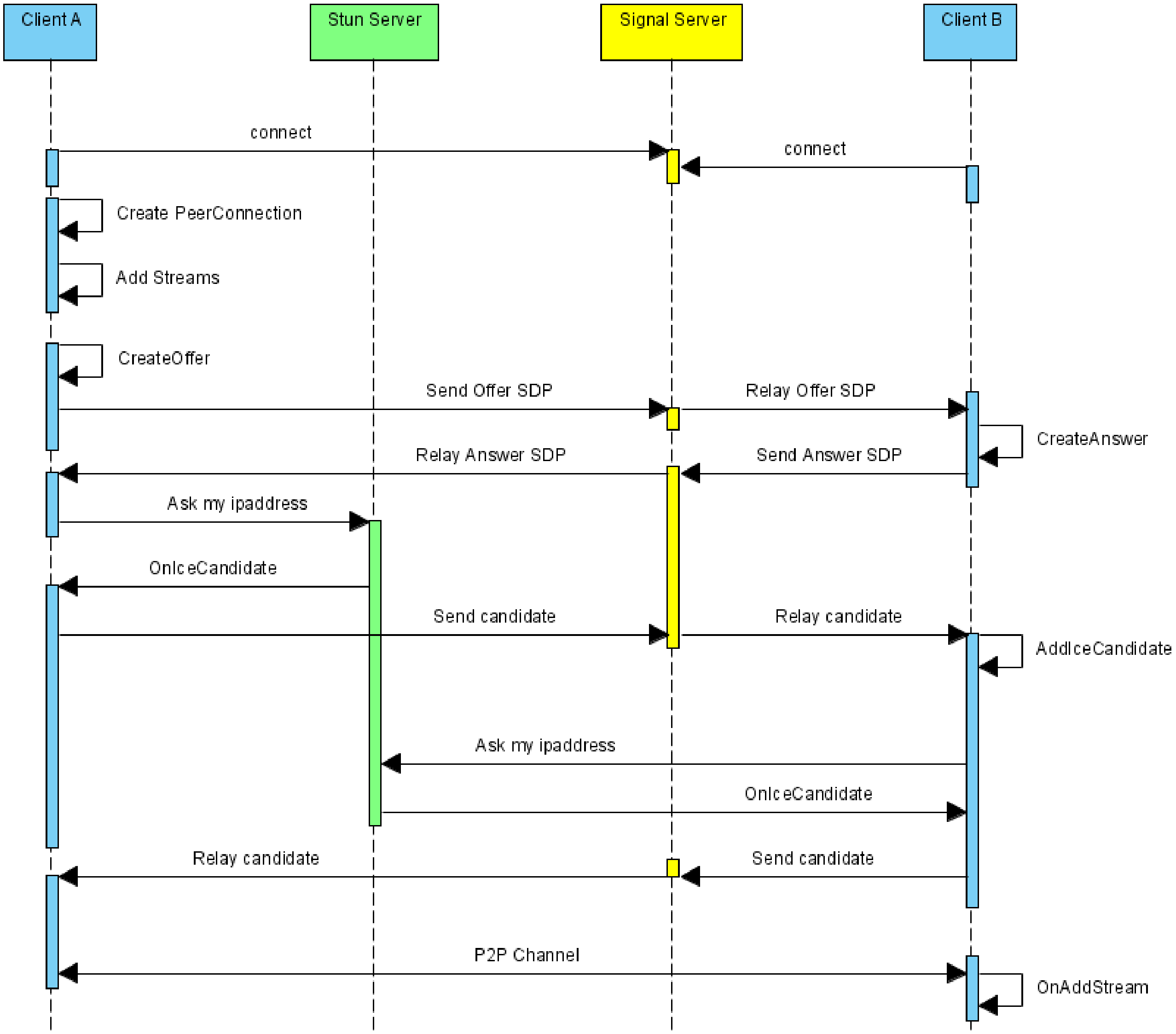

Figure 3; Example Peering Establishment

Figure 3 Shows an example of the sequence of operations to create a WebRTC connection directly between two clients without the use of a TURN server, our WebSocket server performs the role of the Signal Server in the diagram.