CUBIXEL

Functional Specification

TutorPoint

Group 2
MEng Software Engineering Project
Department of Electronic Engineering
University of York

Document Version Control

History of edits and alterations to the CUBIXEL functional specification, including the document version, date, author, and description of the edits.

Version numbering is based on the significance of change.

Key: (major-change/milestone**.**new-section**.**section-edit)

Version	Author	Date	Section Modified	Remarks
0.1.0	OS	13/01/20	All Sections	Document created and basic structure laid out.
0.1.1	JG	14/01/20	All Sections	Added new sections. Added project flow diagram. Started product description.
0.1.2	OS	15/01/20	2. Project Scope 6. Product Requirements	Started project scope. Started requirements. Amended client agreement.
0.1.3	EW	16/01/20	3. Risks	Started risks.
0.2.0	CM	16/01/20	6.5.1 Mutual Non-Disclosure	Added new section. Defined NDA between client and company.
0.2.1	JG	16/01/20	5. Product Features	Added details of the product features that the customer wants the app to include.
0.2.2	CM, CS, JG, OS	17/01/20	All Sections	Finishing draft for submission.
0.2.3	OC	17/01/20	3. Risks	Separated program and financial risks.
0.2.4	OS	18/01/20	3. Risks	Rephrased all risks.
1.0.0	**OS**	**18/01/20**	**Multiple Sections**	**Initial draft complete.**
1.0.1	OS	19/01/20	All Sections	Formatting for submission.
1.0.2	OS	19/01/20	All Sections	Proofreading for submission.
1.0.3	JG	21/01/20	All Sections	Proofreading and further detail added to feature descriptions.
1.0.4	EW, SM, OS	23/01/20	6. Product Requirements	Changed the requirements to a list format in response to feedback.
2.0.0	**All Members**	**27/01/20**	**All Sections**	**Document approved by team.**
2.0.1	OS	04/03/20	2. Project Scope	Small corrections.

Document Approval

All authors of the document are required to proofread, mandate, and sign-off before the document's official submission to the client.

Author	Signature	Date
Daniel Bishop		27/01/2020
Oliver Clarke		27/01/2020
James Gardner	J.	22/01/2020
Stijn Marynissen		23/01/2020
Che McKirgan		27/01/2020
Cameron Smith	smith	24/10/2020
Oliver Still		21/01/2020
Eric Walker		23/01/2020

Preface

This document is the Functional Specification report to provide the reader with an understanding of what services CUBIXEL are offering for the client in order for the company to develop the proposed application (the SERVICES). Included in this document will be detailed functional requirements, including use cases, program inputs, outputs and process flows.

All brands and trademarks of CUBIXEL used are registered trademarks of CUBIXEL, unless otherwise indicated. It is forbidden to use, copy, reproduce, republish, upload, forward, transmit, distribute or modify in any way brands or logos owned by CUBIXEL, without the prior written approval from the respective company itself.

The company, Cubixel (“we”, “our”, “us”), may be stylised as “CUBIXEL” and may be referred to as the COMPANY.

The client (“you”, ”their”, ”them”), may be referred to as the CLIENT.

The proposed application, TutorPoint, may be referred to as the APPLICATION, SOFTWARE, PROJECT, PROGRAM, or the GOODS.

Contents

1. Product Description

TutorPoint is a desktop based, Java application intended to provide a digital lecture environment for both students and lecturers. It is open to any subject that a tutor would like to teach but with a particular focus on STEM subjects at the college and university level. It provides the tools to enable delivery of lecture content live to an audience with instant feedback from users on topics being covered. Users will be able to create an account as either a student or a lecturer and begin watching or producing content on a subject of their choice.

The aim of TutorPoint is to provide an online platform for students and specialists to share, converse and develop their respective fields. The platform is to be structured in a way as to aid education above any other purpose, incorporating a screen for XML presentations, an interactive whiteboard, and text/video chat. The goal of TutorPoint is to nurture limitless education in all fields globally, enabling new fields of studies and advancement in developing countries and niche areas of expertise.

2. Project Scope

2.1 Deliverables

The project will be delivered to the client in the form of iterations at regular two-week intervals leading up to the discussed deadline. The iterations will consist of demonstrational concepts or early functional versions of the overall program, delivered in closed meetings between company and client representatives to allow for discussion and feedback of features, time scales, and projected costs.

2.2 Costs

The predicted cost for the project’s development will be delivered alongside this functional specification document as a ‘Financial Business Plan’ detailing any upfront deposits, advance payments, and additional licence costs.

2.2 Deadlines

The first customer demo will be presented to the client at **12:00pm Friday, 13th March 2020.**

The final customer demo will be presented to the client at **12:00pm Thursday, 4th June 2020.**

Future deadlines, including the project’s final iteration and customer demo, will be discussed and scheduled as required.

Two-week project iterations will be delivered and assessed internally by the company where an optional project summary log will be produced for the client to view if requested.

3. Risks

The project is subject to several risks, during both the development and after the application's release. By agreeing to commit the project's development to the company, the client acknowledges these potential risks and agrees to the precautionary measures that the company and client will adhere by to minimise such risks from occurring.

3.1 Program Risks

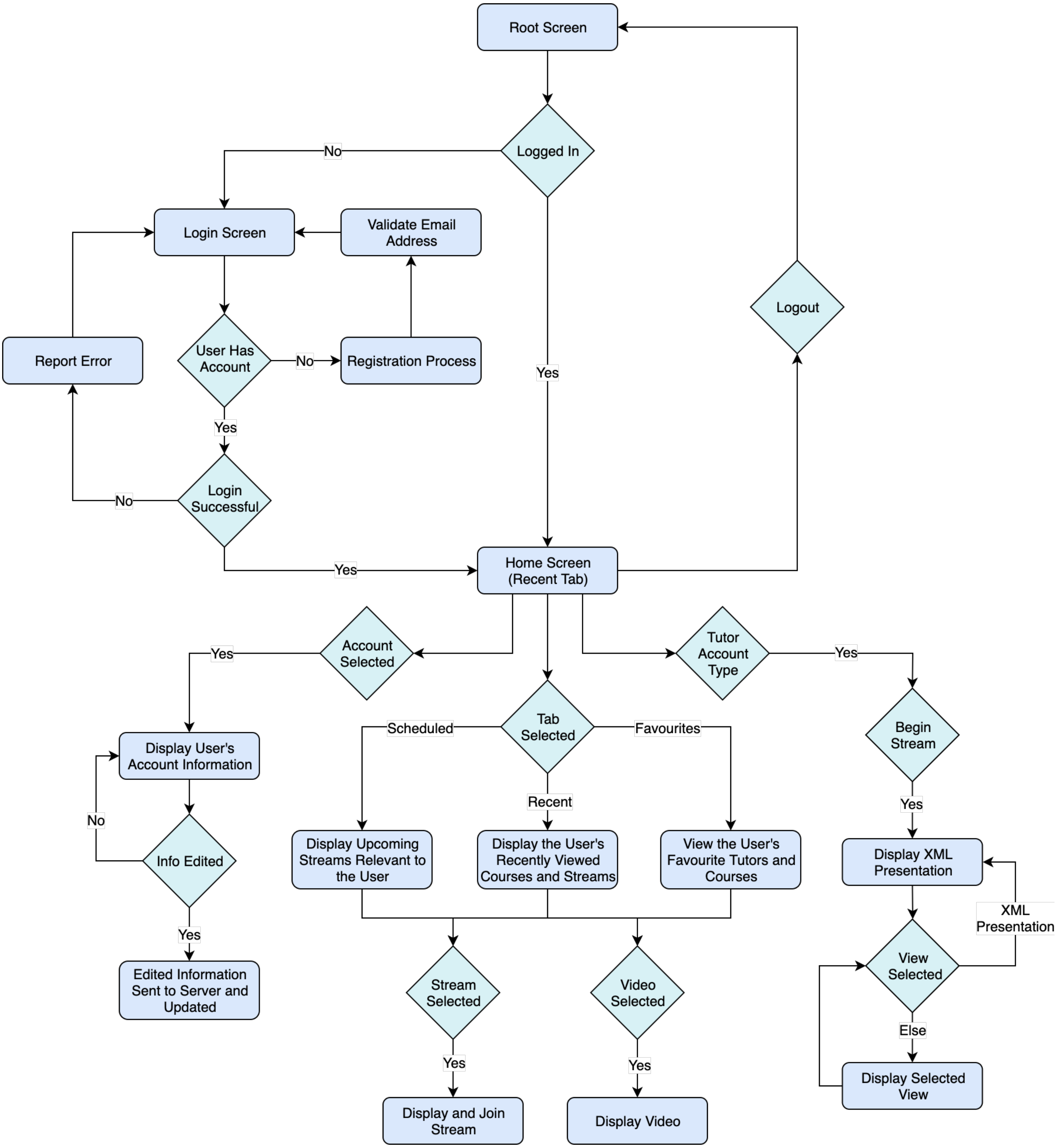
- Compatibility with third-party software used within the program after an update or discontinuation.
  - *Likelihood: LOW, Impact: HIGH*
  - The program will use third-party API's provided through the popular plugin Maven. This carries the risk of incorporating new bugs, security flaws being exposed, or software incompatibility issues.
  - The risk will be managed by developing with the latest update of all third-party software at all times to ensure security and dependability at the next iteration deadline. Each third-party API will be tested for security and compatibility before being incorporated and updated to the main branch.
- Large demand on servers as user base expands.
  - *Likelihood: HIGH, Impact: HIGH*
  - The risk will be managed by the client hosting and supporting sufficient servers, so the responsibility of the risk falls upon the client.
- Users stream copyrighted material on servers for use in their courses.
  - *Likelihood: HIGH, Impact: LOW*
  - The risk will be managed by requiring all users to agree to a terms of service agreement during the registration process, coupled with general moderation of courses and users.
  - This makes them legally liable for any copyright infringement or damages caused, therefore the responsibility falls upon both the client and the user.
- Program becomes unstable and prone to bugs.
  - *Likelihood: LOW, Impact: MEDIUM*
  - The risk will be managed by the company using Agile Development during the software's development, attempting to produce bug-free code as it is incorporated into the software to ensure that the product is stable throughout development up until the release.
- Scope creep results in ever-expanding requirements that cannot be fulfilled before the project deadline.
  - *Likelihood: LOW, Impact: MEDIUM*
  - The risk will be managed by producing and agreeing to a predefined product specification (this document) that will only be changed when absolutely necessary.
- Discrepancy between the company's delivered iteration and the client's expectations.
  - *Likelihood: LOW, Impact: HIGH*
  - The risk is addressed in section 5 and section 6.3 of this document.

3.2 Financial Risks

- Company runs out of budget before the project is complete.
    - *Likelihood: LOW, Impact: HIGH*
    - The risk will be managed by the keeping of a budget with predictions of how long each stage of the product will take to complete, while ensuring that this time remains within budget.
- Accumulating amounts of money owed which is not affordable.
    - *Likelihood: LOW, Impact: HIGH*
    - The risk will be managed by composing sensible and affordable contracts with customers and investors, ensuring the project budget can accommodate all financial contracts or investments.
- Financial backer removes funding.
    - *Likelihood: LOW, Impact: HIGH*
    - The risk will be managed by setting realistic and reachable project expectations to demonstrate progress between the company and the financial backer.
    - A background check on the financial backer will be performed to avoid any financial or fraudulent misconduct.

4. Functional Specification

4.1 Program Flow Diagram

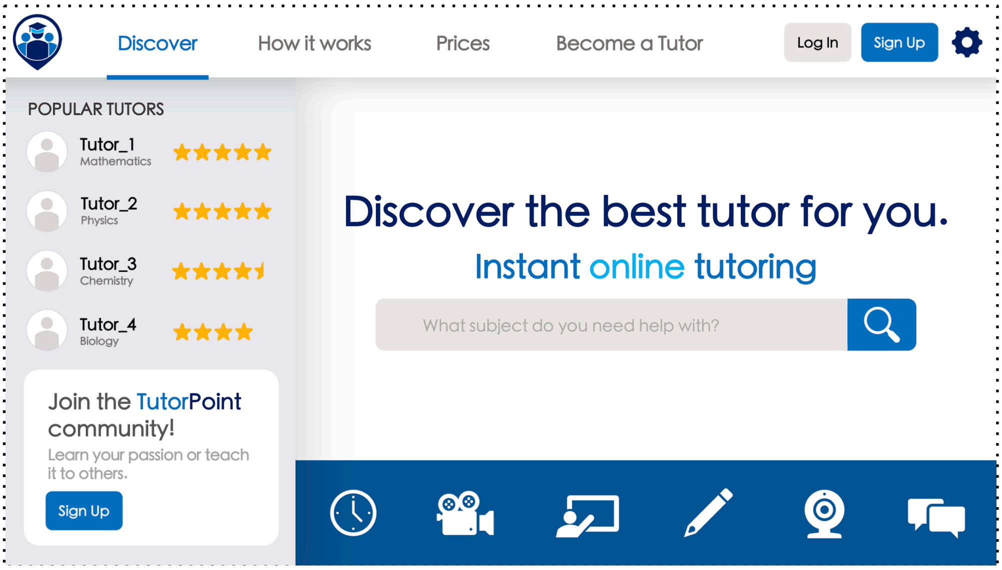
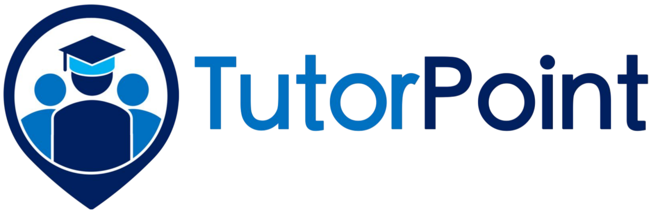
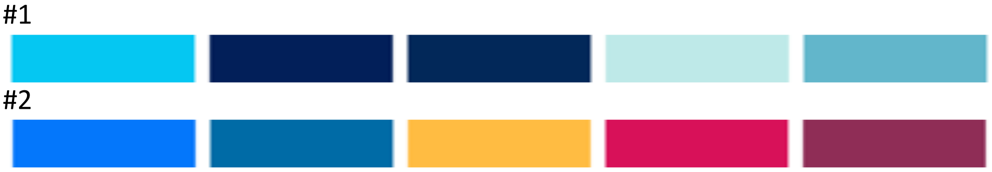


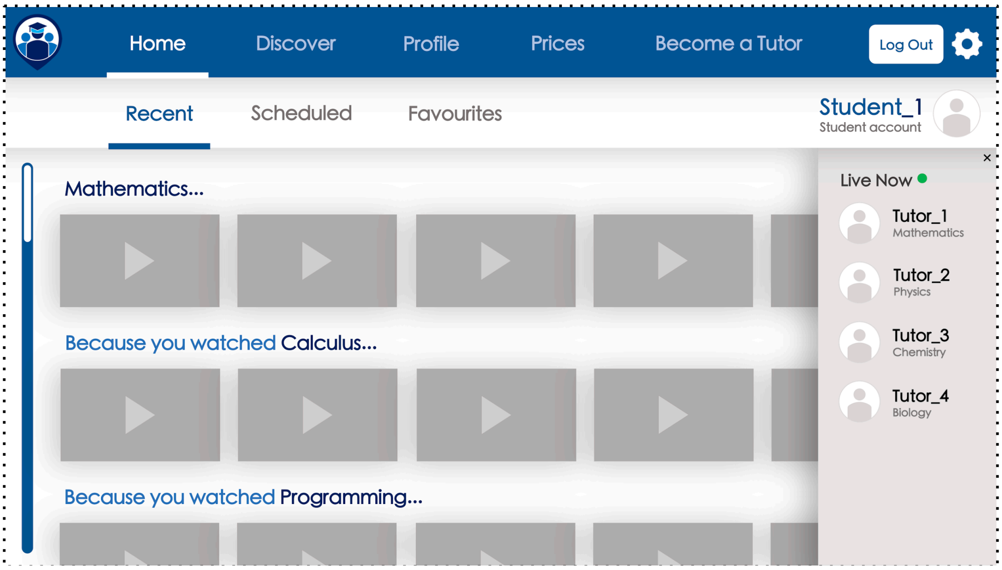
5. Product Features

The application is subject to adaptations and feature removals in each project iteration with prior discussion and agreement between the company and the client, only if the project scope is predicted to lie outside the bounds of the client's cost or timeframe.

5.1 High Priority

Feature ID	Feature Description
TP001	User registers a new account and the information is stored on a secure server. Information required from the users is just a username, email and password. How the database functions is unimportant other than that passwords are stored securely. *Design Parameters:* - Username does not contain vulgar language. - Usernames only ASCII characters, no spaces. - Password strength, at least one uppercase, one symbol, and one number. - Usernames must be unique. - Username must not contain any personal information. - Username cannot be the same as password. - Reset link for lost password. - Email confirmation for setting up account. - Username or Email to log in. - Users can have different public names to the username.
TP002	Users can login into an existing account once and be remembered if they choose to be from the login screen. If the user has chosen to be remembered their account information is stored and encrypted locally and is automatically checked brought directly to the home screen. *Design Parameters:* - Checkbox to remember being logged in. - Users will be logged out after a week of inactivity.
TP003	The login screen user interface has limited access to the application. The screen is presented if the user opens that app but has not logged in or has not chosen to be remembered. If the user has chosen to be remembered they are brought directly to the home screen. Access includes the ability to search for courses, but not to view them.

	Potential Design Layout for Login Screen:  Discover How it works Prices Become a Tutor Log In Sign Up POPULAR TUTORS Tutor_1 Mathematics Tutor_2 Physics Tutor_3 Chemistry Tutor_4 Biology Join the TutorPoint community! Learn your passion or teach it to others. Sign Up Discover the best tutor for you. Instant online tutoring What subject do you need help with?
TP004	The home screen user interface has easy access to all the main features of the app. It will show to the user a list of their currently subscribed courses and recommends new content they might be interested in. *Potential Product Logo:*  *Potential Colour Schemes:*  *Design Philosophy:* Design philosophies extend from the likes of other subscription-based companies such as YouTube or Netflix, utilising the video carousel feature to view previously recorded streams. Users can view subscribed courses and tutors that are currently streaming, as well as recommended content through the 'Discover' tab. The colour palettes are both a derivative of the product logo and the likes of Apple's approach to design, utility, and the user experience.

	Potential Design Layout for Home Screen:  Home Discover Profile Prices Become a Tutor Log Out Recent Scheduled Favourites Student_1 Student account Mathematics... Because you watched Calculus... Because you watched Programming... Live Now Tutor_1 Mathematics Tutor_2 Physics Tutor_3 Chemistry Tutor_4 Biology
TP005	A user should be able to easily view all their account information and data in a transparent way. It should be easy to update their account information. *Design Parameters:* - Any updates require the user to enter their password. - Must use the reset link to change the password.
TP006	If a user signs up as a tutor they should have an account that provides them with extra features available that the students do not. *Design Parameters:* - It should be possible to upgrade to a tutor account. - Tutor application process managed by the client; this should prove some level of competence in the desired teaching subject.
TP007	Tutors can use a microphone to communicate with students in the presentation. *Design Parameters:* - File type doesn't matter. - Audio quality should be at least 128kbps. - Latency should be less than five seconds.
TP008	The app should be able to load in XML based presentations based on a predefined standard (the schema). It should be able to display the content of the XML presentation correctly with text, images, videos and graphics. *Design Parameters:* - Videos only need to be HD. Max file size of 1GB. - Videos will be streamed in from the server. - Total max presentation size of 20GB. - Graphics have a maximum of 1000 vertices. - Images, text, graphics and video can be displayed on a single slide. -

TP009	Users can choose to follow tutors. A list of their followed tutors should determine what the user sees on their app home screen, including previously recorded streams and their recommendations. No limit on the number of tutors followed. Display should show only the first ten tiles and then have a ‘see more’ button.
TP010	A tutor can create a stream that other users can join, this should display the XML presentation on both user’s screens. It should show the current slide that the Tutor is on at all times for all users viewing the stream as the tutor progresses through the presentation in real time. Maximum presentation latency of five seconds.

5.2 Mid Priority

Feature ID	Feature Description
TP011	Both students and tutors have access to an interactive whiteboard. This can be used either whilst watching a stream or not. If the tutor is in a one-to-one session with a student, they can set the whiteboard to be shared so that both the tutor and the student can write on the board in a collaborative manner. The student shouldn’t be able to remove lines drawn by the tutor, but the tutor can remove the student’s lines. It should have a set of basic shapes that can be quickly dropped onto the whiteboard.
TP012	Whilst in a stream, the tutor can swap between the presentation and the whiteboard views with one click. It should remember the position of the slideshow and continue from where it left off. When the tutor returns to the whiteboard all the drawings and text should still be there. Each section is on a separate tab.
TP013	Users should have public profiles that show some basic information about them. Such as username, followed subject/tutors, number of hours watched, lessons/courses complete, user rating, content they have created, display name.
TP014	Users can review tutor’s streams and content. Higher rated tutors will get priority in the recommendations. A user’s rating should be viewable on their public profile. Ratings are out of five stars. Video ratings are public to all.
TP015	Users can ask questions during a stream in a public chat room that other users can all see. Other users can then ‘like’ a question if they want and higher rated questions will be shown to the tutor during the stream to provide feedback on misunderstood topics. Users are limited to three questions every fifteen minutes. 200 Character question length (including punctuation). This only applies to a live stream in a one-to-many session.
TP016	There should be a public chat room alongside each stream for users to comment on. There should be some way to report comments/users for bad language or bullying. This is separate to the ‘question asking’ section. There is a filter to block general bad language. An updatable list of disallowed words provided by the client.

5.3 Low Priority

Feature ID	Feature Description
TP017	It should be possible to live stream video from a webcam to the users watching the stream. In a one-to-one session it should be possible to have two-way video streaming. Report functionality. Can close a video stream if you don't want to watch. Video up to HD Quality.
TP018	Tutors can create question papers that can be completed by students within the app. Questions from students and feedback on the document is collaborative enabling refinement of the questions and teaching material. Everyone gets the same worksheet but answers separately. Statistics of users' results sent to the tutor.
TP019	Academic institutions can register and provide a hub for their tutors and subjects. This can be used to offer accredited courses around the world. You can follow institutions like a tutor. Videos displayed to users are for the courses you've paid for by that institution. Institutions have lots of options to customise their pages.
TP020	Full capture of live streamed lectures including any whiteboard, slideshow or video interactions for viewing at a later time. Tutor can edit videos and take videos down. Tutors can choose if they are recorded/uploaded in the first place. Maximum stream length of three hours.
TP021	The whiteboard should come with some extra useful images and characters related to STEM subjects (scientific symbols) that would make learning easier. Extra icons sets can be provided by registered institutions to suit their needs.

6. Product Requirements

6.1 User Requirements

Users are required to have:

1. A computer or device of the minimum software requirements to run the program.
2. Access to a stable internet connection in order to communicate with the client's servers.
3. A valid and accessible personal email address.
    a. Users are therefore required to give consent to the client to handle their digital information on the client's servers in order to use the service. The security and privacy of the user's information is discussed in detail in section 6.4.

Furthermore, if the user consents to registering their personal information, they admit and agree to:

4. Follow the set terms of service defined by the client.
5. Report content or other users that break the terms of service.

Users with approved tutor accounts are required to either:

6. Own all material used in presentations.
7. Explicitly disclose the source of any content they do not own themselves.
    a. Therefore, users retain all rights and ownership to their own course material only if the material adheres to the terms and conditions defined by the client.

6.2 Program Requirements

The program itself is required to have:

8. Access to a stable and sufficient internet connection with firewall access, with the launching and quitting of the program being fast, reliable and safe.
9. Servers of sufficient capacity and redundancy to ensure constant uptime.
10. User submitted presentations to be written in accordance with the XML schema, as defined by the company.
11. User submitted presentation file content to be within the size limit, as defined by the client.
12. A means of detecting errors and reporting such errors or crashes to the client's software development team.

With respect to the performance, the program is required to be:

13. Smooth and reliable, regardless of the user's machine specifications.
14. Quick and consistent in its response to user input or interaction, with little or no jarring amounts of delay.
15. Smooth in video playback with little to no buffering.
16. Fast in image load and display times, along with smooth transitions.

6.3 Client Requirements

The client is required to provide:

17. The cost and support of sufficient services and server time to host the server-side of the application.
18. All of the program's terms and conditions, including the application process for tutors, before the program's date of launch.
19. A client representative as a full-time member of the team on site throughout development.
    a. They should have a good understanding of their product vision to answer questions regarding the product and instruct the application's design.
    b. They must be available for meetings at the end of each iteration.
    c. They must be able to view the current progress of the software and provide feedback to be implemented in the next iteration in guidance of the client's image and expectations.

In addition, after the application's official launch, the client is responsible for:

20. Providing their own software development team to deal with error correction on any found bugs or crash reports relating to the program functionality that have been resourced post-release, as well as any updates post-release.
21. Policing of the content present on the platform (or make sure policing is active on the platform), ensuring that there are no laws being broken (copyright protection, age-restricted content, etc.).

6.4 Security & Privacy Requirements

The user's personal account login information, including email(s), password(s), and any other type of sensitive data, must be stored securely using sufficient levels of data encryption.

Requirements must conform to Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation, abbreviated to GDPR).

When offering an online service directly to a child in the UK, in order to be compliant with article 8 of the GDPR, only children aged 13 or over are able to provide their own consent; for children under this age you need consent from whoever holds parental responsibility for the child. Therefore, users must confirm they are over the age of 13 when registering for the service themselves.

6.5 Client Agreement

6.5.1 Mutual Non-Disclosure Agreement

1. Each of the parties to this Agreement intends to disclose information (the Confidential Information) to the other party for the purpose of the development of the application specified (the Purpose).

2. Each party to this Agreement is referred to as 'the Recipient' when it receives or uses the Confidential Information disclosed by the other party.

3. The Recipient undertakes not to use the Confidential Information disclosed by the other party for any purpose except the Purpose, without first obtaining the written agreement of the other party.

4. The Recipient undertakes to keep the Confidential Information disclosed by the other party secure and not to disclose it to any third party except to its employees who need to know the same for the Purpose, who know they owe a duty of confidence to the other party and who are bound by obligations equivalent to those in clause 3 above and this clause 4.

5. The undertakings in clauses 3 and 4 above apply to all of the information disclosed by each of the parties to the other, regardless of the way or form in which it is disclosed or recorded but they do not apply to:

   a) any information which is or in future comes into the public domain (unless as a result of the breach of this Agreement); or

   b) any information which is already known to the Recipient and which was not subject to any obligation of confidence before it was disclosed to the Recipient by the other party.

6. Nothing in this Agreement will prevent the Recipient from making any disclosure of the Confidential Information required by law or by any competent authority.

7. The Recipient will, on request from the other party, return all copies and records of the Confidential Information disclosed by the other party to the Recipient and will not retain any copies or records of the Confidential Information disclosed by the other party.

8. Neither this Agreement nor the supply of any information grants the Recipient any licence, interest or right in respect of any intellectual property rights of the other party except the right to copy the Confidential Information disclosed by the other party solely for the Purpose.

9. The undertakings in clauses 3 and 4 will continue in force indefinitely**.**

10. This Agreement is governed by, and is to be construed in accordance with, English law. The English Courts will have non-exclusive jurisdiction to deal with any dispute which has arisen or may arise out of, or in connection with, this Agreement.

6.5.2 Service-Level Agreement

The following signatures confirm that an agreement between the CLIENT and the Project Manager, on behalf of the COMPANY (CUBIXEL), has been made to proceed with the development of the software application according to the preceding functional specification document.

The CLIENT hereby agrees to engage the COMPANY to provide the SERVICES defined in this document.

The CLIENT further acknowledges and accepts responsibility for the respective program related and financial risks set out by the COMPANY during the project's development.

In addition, both the COMPANY and CLIENT agree to the terms laid out in section 6.5.1 to non-disclose all information relating to the project and company operations.

Work shall only commence at an agreed time following the completion and documentation of this agreement.

Signed by:	*Signed by:*
for and on behalf of	*for and on behalf of*
THE COMPANY	***THE CLIENT***
Signature:	**Signature:**
Date:	**Date:**