Product Backlog

Note: Strikethrough stories were removed due to changing requirements of the clients and will not be implemented

Sprint Backlog Colouring Scheme

Sprint 1 Backlog
Sprint 2 Backlog
Sprint 3 Backlog
Sprint 4 Backlog

~~Introduce SSQ to the public~~

~~As a...~~	~~I want to be able to...~~	~~So that...~~	~~Priority~~
~~Patient~~	~~See some basic information about the SSQ scale~~	~~I can have background context of SSQ to have a basic understanding of the questionnaire I am about to undertake.~~	~~Must-Have~~

Login Interface

As a...	I want to be able to...	So that...	Priority
Clinician	Signup to the system	I can have a clinician account on the system	Must-Have
Clinician	Login to the system	I can conduct and complete questionnaires or modify questionnaires.	Must-Have
Administrator	Login to the system	I can edit information on the website or modify the questionnaires.	Must-Have
Clinician	Signup with personal information	I can be authenticated by the system	Must-Have

View/Complete Questionnaire

As a...	I want to be able to...	So that...	Priority
Clinician	Conduct questionnaire for each section	I can conduct a hearing test through face-to-face	Must-Have
Clinician	Conduct questionnaire for each section	I can conduct a hearing test through Zoom or phone call	Must-Have
Clinician	Allow my patient to complete the test on their own	They can complete the questionnaire even if I am unavailable	Must-Have
Clinician	View the questionnaire	I can ensure the questions are correct and relevant and allow my patients to complete the questionnaire with or without my supervision	Must-Have
Clinician	Only have access to my own questionnaires	I know my own custom questionnaires can be private from others	Must-Have

Administrator manage standardised contents

As a...	I want to be able to...	So that...	Priority
~~Administrator~~	~~Edit SSQ information for general public~~	~~I can inform the public up-to-date information about the SSQ scale.~~	~~Should-Have~~
Administrator	Edit SSQ administering instructions	I can provide up-to-date instructions for clinicians to administer questionnaires	Should-Have
Administrator	Add new questions to the standardised questionnaires	I don't need to create a new questionnaire to include a new question	Should-Have

Administrator	Remove questions from the standard questionnaires	I can remove out-of-date or unnecessary questions	Should-Have
Administrator	Edit existing questions from the standardised questionnaires	My questions can be more flexible and makes sense for the patients	Should-Have
Administrator	Use other questionnaires to create a new customised questionnaire	I can create a customised questionnaire by using previous questionnaires instead of creating them from scratch	Could-Have

Clinicians manage customised questionnaires

As a...	I want to be able to...	So that...	Priority
Clinician	Create a new customised questionnaire	I can have a customized questionnaire for a specific purpose	Should-Have
Clinician	Delete the customised questionnaire	I can remove the out of date content and make my space clean	Should-Have
Clinician	Add new questions to the customised questionnaire	I don't need to create a new questionnaire to include a new question	Should-Have
Clinician	Remove questions from the customised questionnaire	I don't need to create a new questionnaire in order to remove one question	Should-Have
Clinician	Attach comments on specific questions in the questionnaire	I can refer back to them later on	Should-Have
Clinician	Use other questionnaires to create a new customised questionnaire	I can create a customised questionnaire by using previous questionnaires instead of creating them from scratch	Could-Have

View/Export Report

As a...	I want to be able to...	So that...	Priority
Clinician	Generate a report based on the sections of the questionnaire I completed	I can see a summarised version of the questionnaire responses.	Must-Have
Clinician	See some administration information including patient name/id and the time of completion on the top of the report when I view the report	I can identify which patient the report belongs to and when the questionnaire was completed.	Must-Have
Clinician	See a correctly calculated average score (exclude any unanswered questions) of each section of the questionnaire when I view the report	I can gain a global perspective on the patient's hearing disabilities.	Must-Have
Clinician	See a correctly calculated average score (exclude any unanswered questions) of each sub-scale of the questionnaire when I view the report	I can view patient's hearing disabilities from some predefined perspectives.	Must-Have
Clinician	See an ordered list of questionnaire responses sorted by their level of importance and then by frequency when I view the report	I can understand what the patient considers to be the most important hearing problems in their day to day life.	Must-Have
Clinician	See an ordered list of questionnaire responses sorted by performance ratings when I view the report	I can understand what the patient performs most well at in their day to day life of hearing.	Must-Have
Clinician	See any comment the clinician entered (when completing the questionnaire) next to the corresponding response for each item in the ordered list when I view the report.	I can record any important information that was not captured in the questionnaire responses.	Could-Have
Clinician	Have the report emailed to me as a document.	I can store the questionnaire report locally or send it to the parent of the child patient and ensure that no other users are able to view the results of the questionnaire.	Must-Have
~~Clinician~~	~~Upload and Compare 2 or multiple reports of the same type and generate a graph capturing the major differences of the patient responses~~	~~I can explain to the parent visually how the child's hearing abilities have improved or deteriorated~~	~~Nice-to-Have~~

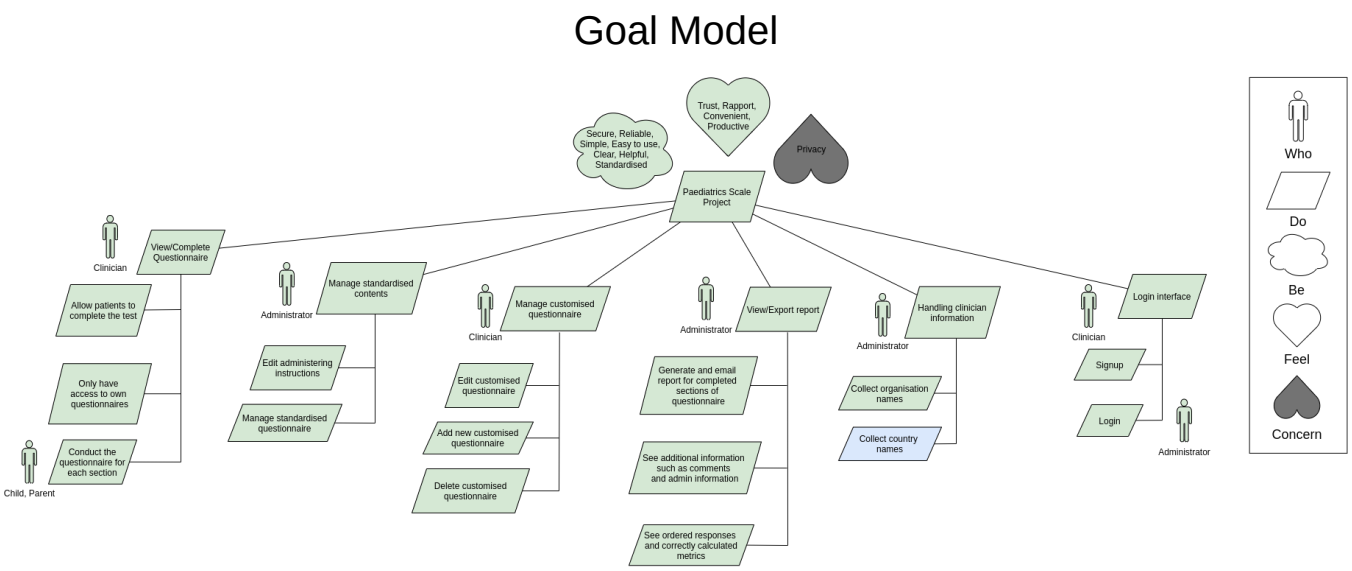
~~Multilingual Interface~~

~~As a...~~	~~I want to be able to...~~	~~So that...~~	~~Priority~~
~~Clinician~~	~~Have multilingual interface~~	~~My patient has a full understanding of the questions asked~~	~~Nice-to-Have~~
~~Clinician~~	~~Have multilingual language options for the SSQ questionnaire~~	~~I can undertake tests and gain comprehensive understanding of my patient's hearing situation even if English was not their first language~~	~~Nice-to-Have~~

Administrator handle clinician information

As a...	I want to be able to...	So that...	Priority
Administrator	collect organisation names of each clinician	I can send out updates and changes to the organisation.	Nice-to-have
Administrator	collect country information of each clinician	I can understand where the system is currently being used	Nice-to-have

Goal Model



WHO/DO/BE/FEEL list Colouring Scheme

New
Modified
Unchanged

WHO/DO/BE/FEEL list 8.0

Who	Do	Be	Feel	Concerns
Administrator	Paediatric Scale Project	Standardised	Productive	Privacy
Clinician	Login Interface	Secure	Trust	
Parent	Signup to the system with personal information	Reliable	Rapport	
Child	Login to the system	Helpful	Convenient	
	View/Complete questionnaire	Simple		
	Conduct the questionnaire for each section through, face-to-face, zoom or phone call	Easy to use		
	Only have access to own questionnaires	Clear		
	Allow patients to complete the test			
	Check correctness and relevance when viewing the questionnaire			
	Manage standardised contents			
	Edit administering instructions			
	Add new questions to the standardised questionnaires			
	Edit existing questions from the standardised questionnaires			
	Remove questions from the standardised questionnaires			
	Use other questionnaires to create new customised questionnaire			

	Manage the customised questionnaires			
	Create a new customised questionnaire			
	Add new questions to the customised questionnaire			
	Remove questions from the customised questionnaire			
	Delete the customised questionnaire			
	Attach comments on specific questions in the questionnaire			
	View/Export report			
	Generate report based on the sections of the questionnaire completed			
	See administration information including patient name/id and the time of completion on the top of the report when viewing the report			
	See correctly calculated average score of each section of the questionnaire when viewing the report			
	See correctly calculated average score for each sub-scale of the questionnaire when viewing the report			
	See an ordered list of questionnaire responses sorted by importance and then by frequency when viewing the report			
	See an ordered list of questionnaire responses sorted by performance ratings when viewing the report			
	See any comment of clinicians entered next to the corresponding response for each item in the ordered list when viewing the report			
	Have the report emailed as a document			
	Handle clinician information			
	Collect organisation names of each clinician			
	Collect country names of each clinician			

Database Diagram

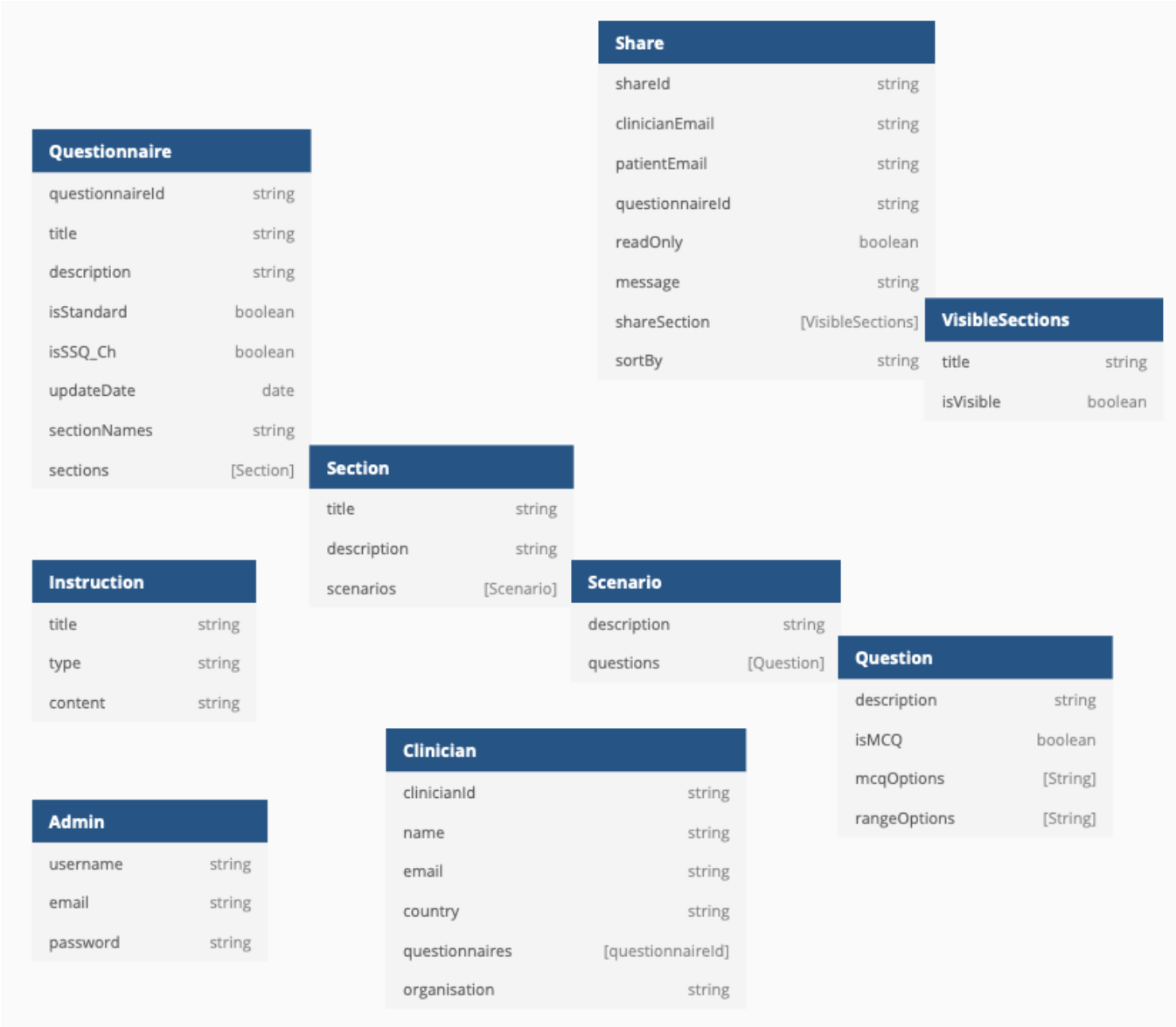
Database Structure:

There are three main roles in the system: **administrator**, **clinician**, and **participant** (parent/child). Administrator information is stored in a separate collection which has no UI elements to manage the information in it, except for log in, this adds on top the security of the system. Basic clinician information is stored on a separate collections as well, however we don't save **any** participant information due to privacy issues related to health information.

Each **questionnaire** contains an array of **sections**, which intern has an array of **scenarios** which has an array of **questions**. This is reflected on our database schema.

The **share** collections contains information related to the shares made by the clinician to participants, each share section has an array of visible sections. **Vi sible sections** reflects the sections of the questionnaire the clinician shared.

Instruction collections holds the instructions for different types of questionnaires.



Schema Used:

Schema

```
// Creating tables
Table Clinician as U {
  clinicianId string // auto-increment
  name string
  email string
  country string
  questionnaires [questionnaireId]
  organisation string
}


Table Admin as A {
  username string
  email string
  password string
}

Table Questionnaire {
  questionnaireId string // auto-increment
  title string
  description string
  isStandard boolean
  isSSQ_Ch boolean
  updateDate date
  sectionNames string
  sections [Section]
}

Table Section {
  title string
  description string
  scenarios [Scenario]
}

Table Scenario {
  description string
  questions [Question]
}


Table Question {
  description string
  isMCQ boolean
  mcqOptions [String]
  rangeOptions [String]
}


Table Share {
  shareId string
  clinicianEmail string
  patientEmail string
  questionnaireId string
  readOnly boolean
  message string
  shareSection [VisibleSections]
  sortBy string
}

Table VisibleSections {
  title string
  isVisible boolean
}
```

```
Table Instruction {
  title string
  type string
  content string
}
```

API Documentation

Our APIs consist of GET and POST HTTP requests, as shown in the tables below.

Some of the APIs need standard authorisation headers as noted in the last column of each table. The clinician token is issued by Auth0. The admin token is issued by our server.

GET

URL	Description	Parameters	Token
questionnaire/${Id}	get an entire questionnaire object given questionnaire ID.	Id = questionnaireId in questionnaires collection	None
questionnaire/standardised	get a list of questionnaire objects (with *sections* removed) for all standardised questionnaires		None
questionnaire/clinician?clinicianId=${clinicianId}	get a list of questionnaire objects (with sections removed) for all customised questionnaires belonging to clinician with given clinician Id	clinicianId = clinicianId in clinicians collection	Clinician
admin/verifylogin/${accessToken}	verifies the admin is logged in given an access token. A status code of 200 = logged in.	accessToken = a jwt token for the admin user issued by the server	None
admin/specificInstruction/${instructionType}	gets a specific instruction object given the instruction type	instructionType = instructionType in instructions collection	None
admin/instructionsSummary	gets a list of instructions summary object containing the title and the type of each instruction for all instructions in the database		None
admin/country	gets a list of registered countries from all clinicians		Admin
admin/country/organisation/${countryName}	get a list of all organisation that the clinicians registered based on the registered country	countryName = country in clinicians collection	Admin
admin/organisation/clinician/${organisationName}	get a list of clinicians based on their registered organisation	organisationName = organisation in clinicians collection	Admin
share/${shareId}	gets a share instance (a data object recording the share action between a clinician and a client)	shareId = shareId in shares collection	None

POST

URL	Description	Body	Token
questionnaire/addStandard	Creates a standard questionnaire.	undefined	Admin
questionnaire/editStandard	Edits a standard questionnaire.	Object of questionnaire type. (Check the database schema)	Admin
questionnaire/deleteStandard	Deletes the standard questionnaire.	questionnaireId = The questionnaire ID of a standard questionnaire in the questionnaire collection.	Admin
questionnaire/copyStandard	Copies a questionnaire to generate a standard questionnaire	isCopyingToCustomisedQuestionnaire = false questionnaireId = Questionnaire ID of the questionnaire we want to copy.	Admin
questionnaire/add	Creates a custom questionnaire.	clinicianId = The ID of the clinician from the clinician collection. isStandard = false	Clinician
questionnaire/edit	Edits a custom questionnaire.	Object of questionnaire type. (Check the database schema)	Clinician
questionnaire/delete	Deletes a custom questionnaire.	CQid = Customised questionnaire ID clinicianId = The ID of the clinician from the clinician collection.	Clinician
questionnaire/copy	Copies a questionnaire to generate a customised questionnaire	copyToCustomisedQuestionnaire = true questionnaireId = Questionnaire ID of the questionnaire we want to copy.	Clinician
admin/login	Logs in as an Admin	username = username for the Admin account. password = password for the Admin account.	None

admin/instructions/${instructionType}	Update specific instruction for admin	instructionType = Any one of these depending on the instruction type CP,CC,RC and RP. [parameter] Instructions	Admin
share/submit/${shareId}	Send questionnaire data after filling out the questionnaire	shareId = The shareId of the share from the Share collection. [parameter] QuestionnaireData = Object of questionnaire response. (Check appendix) clinicianEmail = Clinicians email address. personalDetails = Object of personal details. (Check appendix) QuestionnaireId = Questionnaire ID of the completed questionnaire. comments = An array of arrays of comments for each scenario. sortBy = 'PERFORMANCE' or 'FREQUENCY'	None
clinician/share	Share a questionnaire. It will send an email to the participant to start the questionnaire.	Object of the share type. (Check the database schema)	Clinician
clinician/complete-questionnaire	Clinician submits the response to a questionnaire.	QuestionnaireData = Object of questionnaire response. (Check appendix) clinicianEmail = Clinicians email address. personalDetails = Object of personal details. (Check appendix) QuestionnaireId = Questionnaire ID of the completed questionnaire. comments = An array of arrays of comments for each scenario. sortBy = 'PERFORMANCE' or 'FREQUENCY'	Clinician

Appendix

PersonalDetails Example

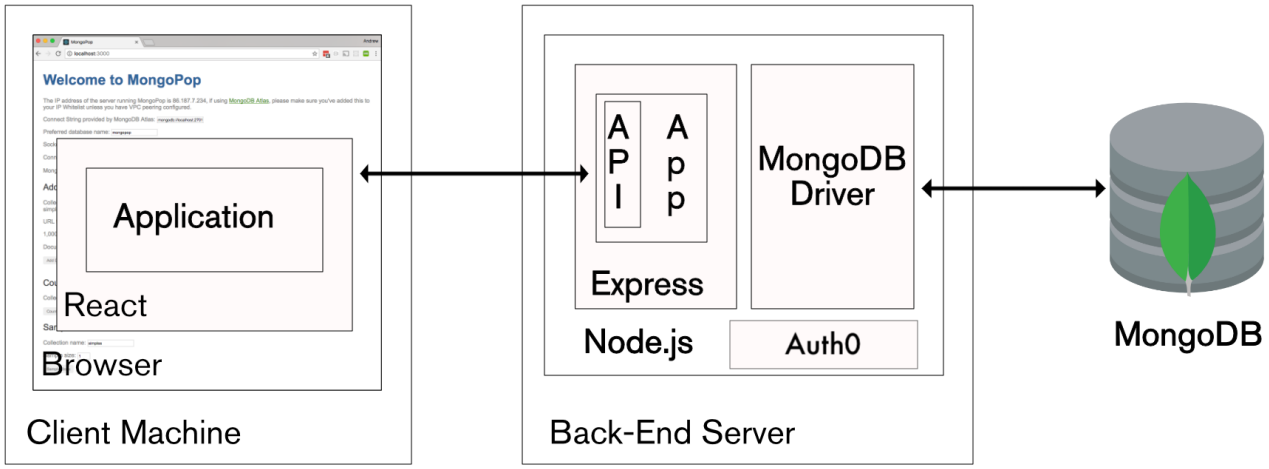
```
  personalDetails: {
    name: 'John',
    date: '2020-10-18',
    rightDeviceType: 'Hearing Aid',
    leftDeviceType: 'Cochlear Implant',
    completedBy: 'clinician',
    completedByRelationship: 'Mother',
    completedByName: 'Jane'
  }
```

QuestionnaireData Example

```
[[{ value: '5.6', supplementaryValue: '' },{value: '', supplementaryValue: 'NA'},{value: 'second item',
supplementaryValue: ''}],[], []]
```

Technical Manual

General web app architecture



Software environment requirements

As we separate our development branch into client-side and server-side, the software environment requirements will also be talked separately. In the client-side, we will list all the supported explorer's versions and dependencies versions. For the server-side, we will discuss the supported deployment hosts and dependencies versions as well.

Client-side:

For the Explorer, we have tested and supported:

- Chrome Version 86.0.4240.75 (Official Build) (64-bit),
- Firefox Version 81.0 (64-bit),
- Microsoft Edge Version 86.0.622.38 (64-bit)

Dependencies and version:

- react@16.13.1
- material-ui@1.0.0-beta.47
- react-router-dom@5.2.0
- prop-types@15.7.2
- react-icons@3.10.0
- react-hook-form@5.7.2
- moment@2.27.0

Server-side:

For Node.js server, most of the computing service host will support it with easy and clear deployment instructions.

Here we will list several popular hosts along with the deployment tutorials.

- AWS

The market-leading cloud platform is one of the best choices to host your modern applications. AWS got thousands of services to meet every requirement.

The tutorial can be retrieved at https://aws.amazon.com/cn/getting-started/hands-on/deploy-nodejs-web-app/

- Azure

The Azure cloud platform is more than 200 products and cloud services designed to help you bring new solutions to life—to solve today's challenges and create the future. Build, run, and manage applications across multiple clouds, on-premises, and at the edge, with the tools and frameworks of your choice.

The tutorial can be retrieved at https://azure.microsoft.com/en-au/develop/nodejs/

- Heroku

Heroku is part of the Salesforce organization. They offer a FREE plan which includes 512MB memory, one web or worker to get you started. Heroku is an excellent choice for beginners and experienced developers, thanks to its fantastic documentation. Our product is currently deployed on Heroku, we have the specific deployment guide about Heroku. But there are also official tutorials provided by Heroku.

The tutorial can be retrieved at https://devcenter.heroku.com/articles/getting-started-with-nodejs

- Digital Ocean

DO is famous among developers and you can get it started from as low as $5 per month (Get $10 credit). If you want to install Node.js on your own, then you can request a plain droplet with your choice of OS and install it. DO offers other infrastructure services such as load balancer, object storage, cloud firewall, etc. which helps you to build an enterprise-ready application.

The tutorial can be retrieved at https://marketplace.digitalocean.com/apps/nodejs

Dependencies and version:

- mongoose@5.9.13
- jsonwebtoken@8.5.1
- uuid@8.1.0
- path@0.12.7
- express@4.17.1
- jwks-rsa@1.9.0
- express-jwt@6.0.0
- nodemailer@6.4.6
- fs@0.0.1-security
- pdfkit@0.11.0
- body-parser@1.19.0
- bcryptjs@2.4.3

Deployment guide

Owner Gmail Account = noreply.ssq@gmail.com

Auth0

We use Auth0 to authenticate clinicians to our application. We created 2 Auth0 tenants because Auth0 cannot allow login from both localhost and a deployed website URL simultaneously. Both tenants can be managed by the owner Gmail account. Note the user stores for the tenants are not synchronised. This means that users registered for each tenant are completely independent. However, developers can access the data of the same user locally and on deployed site if they register that user on both platforms, simply because the user-specific data is stored based on the same email address in MongoDB.

	Tenant Name	Usage
1	pediatric-scale	used locally
2	ssq	used by Heroku

Changes to the logic or UI for login/registration page can be made under *Universal Login*.

We make use of webhooks to send user information (e.g. email address, country, etc) from Auth0 to MongoDB after clinician registration. The sending end point is configured under *Post User Registration* in the *Hooks* page in Auth0

We make use of pre-existing *rules* in Auth0 to attach the clinician's email address into the web token that is sent back from Auth0 to the client application. This web token is then sent to the server where the email address is verified to make requests that are user-specific (e.g. retrieve or update customised questionnaires) When we pass around this Auth0 token, we set the *audience* in the source code of the client and server to be linked to the ClinicianAuth API defined under *APIs*.

MongoDB Atlas

Our database is deployed in the cloud on MongoDB. It is a NoSQL database that stores data in documents. It stores standardised/customised questionnaires, instructions, existing shares (the event that clinician has shared a questionnaire which has not yet been completed) and clinician information.

Note that clinician credentials are not stored in the database, they are stored in Auth0.

Note that questionnaire responses are not stored in the database for privacy reasons.

We make use of webhooks to send user information (e.g. email address, country, etc) from Auth0 to MongoDB after clinician registration. The receiving endpoint is configured in MongoDB *Realm* under *3rd party services*. We can modify the webhook by going into the *trial* service and edit the *register clinicians* webhook.

The database can be managed using the owner Gmail account.

Heroku

Our server and the client application are both deployed on Heroku, corresponding to the apps: *ssq-server* and *ssq-client*. Any updates on the GitHub production branches (production-client, production-server) are automatically deployed to the Heroku application.

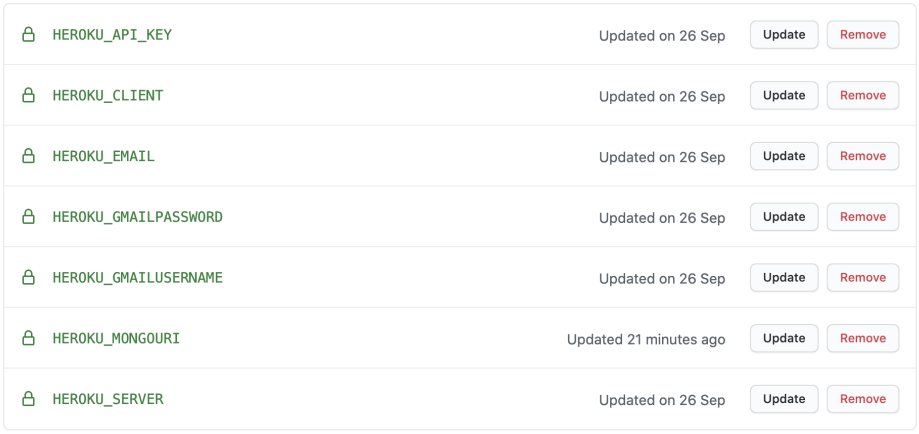
The credentials for the Heroku account are the same as the owner Gmail account.

GitHub

Repository: https://github.com/mayankshar21/SWEN90013-2020-PS

The GitHub repository is set up with a CI/CD pipeline to provide continuous integration to Heroku deployment via GitHub Actions. GitHub automatically deploys commits made to *production-client* and *production-server* branches.

The environment variables that the Heroku apps and MongoDB use are encrypted secrets that can be configured under *Setting*s in GitHub.



We have created a GitHub account for the client and the credentials as the same as the owner Gmail account. This account is also connected to the codefactor tool.

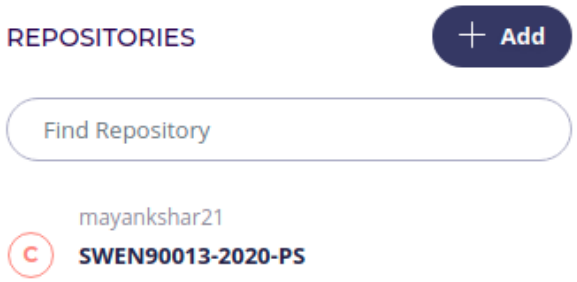
CodeFactor

We are using a new tool for our git repository. Codefactor gives an overview of our code. It tracks issues for every commit and pull request and gives a rating to our repository. It also shows critical issues and provides comments on pull requests on GitHub.

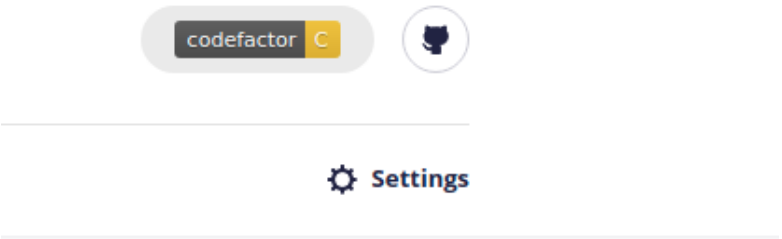
Codefactor - https://www.codefactor.io/repository/github/mayankshar21/swen90013-2020-ps

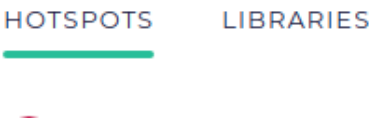
To invite a team member to the codefactor simply follow the steps below:

1. Click on the repository SWEN90013-2020-PS from the left panel of the dashboard

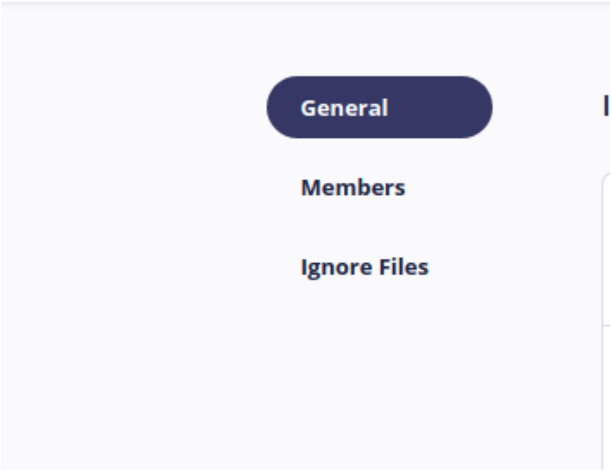


2. Select the Settings button on the right of the navbar



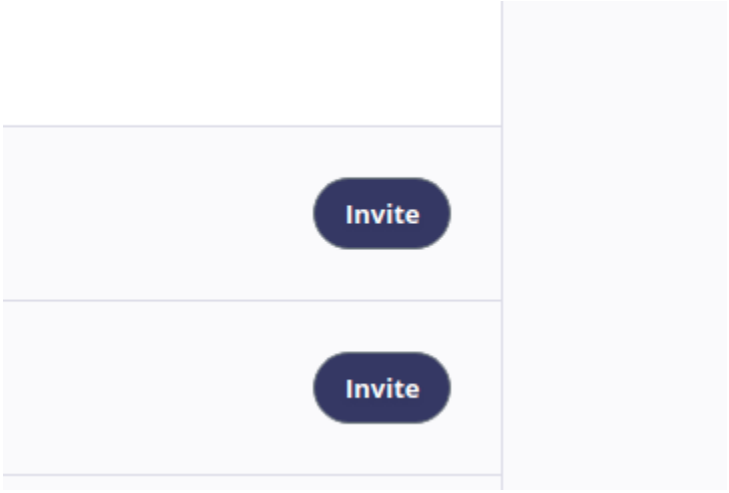


3. On the right menu select Members



4. Click invite beside the team member whom you want to invite

5. The team member should receive an invite link in their email to join



6. If you receive an error on the top of the page, copy the link and send to the team member

OR

7. Ask the team member to connect their GitHub account to codefactor first from the home page before sending the invite request by clicking the invite button from the members page

Home page: https://www.codefactor.io/

AWS(future possibility)

AWS is a viable option when deploying the client and server application to the cloud. It provides all the necessary components to deploy, manage and scale the application in order to serve more users and prevent system downtime. The client and server can be deployed using the following services:

Client

The client can be deployed on AWS Amplify. AWS Amplify allows developers to connect their GitHub repository, configure build settings and eventually deploy the client to the cloud. It comes with its own pipeline that allows developers to test the app before releasing a stable build.

To deploy the client application on AWS Amplify, use the *production-client* branch and follow the on-screen instructions on the AWS console. More information on how to set up the client can be available at https://aws.amazon.com/amplify/.

Server

The server can be deployed on AWS Elastic Beanstalk. AWS Elastic Beanstalk allows developers to deploy, manage and scale the application. It takes care of the underlying infrastructure and maintenance while allowing the developers to focus on developing code and release a stable build.

To deploy the server application on AWS Elastic Beanstalk, use the *production-server*. More information on how to set up the server can be available at https://aws.amazon.com/elasticbeanstalk/.

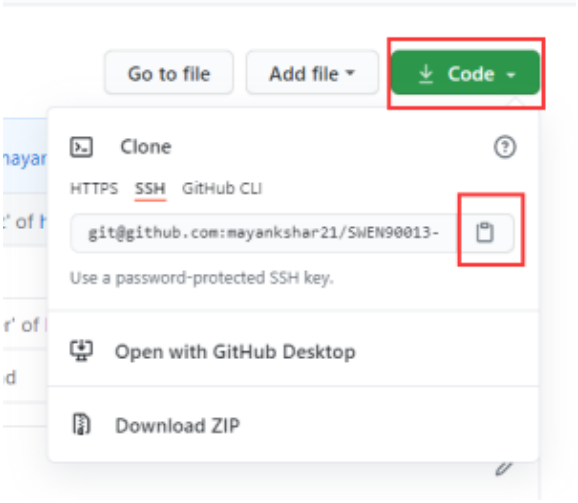
GitHub Actions

The client and the server application can also be connected to AWS Amplify and AWS Elastic Beanstalk using the GitHub Actions pipeline. Guidelines for writing the actions file can be obtained at https://docs.github.com/en/free-pro-team@latest/actions.

Multiple GitHub Actions to connect with AWS and other tools and platforms can also be available from the GitHub Marketplace at https://github.com/marketplace.

How to run the software locally

1.Our branch is https://github.com/mayankshar21/SWEN90013-2020-PS. Go to the page and copy the ssh



2. open cmd line or terminal, input

```
git clone git@github.com:mayankshar21/SWEN90013-2020-PS.git
```

Now the folder will look like this

If you did not install Node on your machine, check this tutorial https://treehouse.github.io/installation-guides/windows/node-windows.html

After installing Nodejs, go to the server folder, open command line, input

```
npm install
```

```
npm start
```

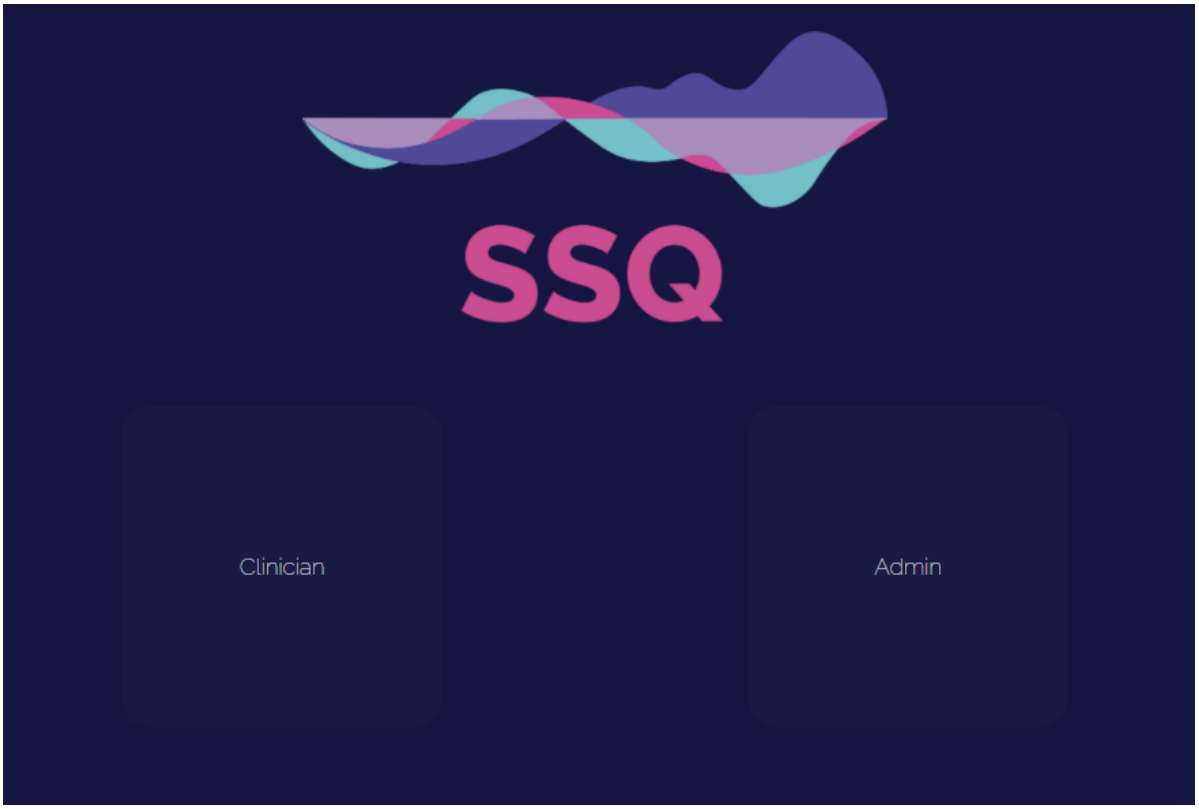
```
λ npm start

> pediatric-ssq-server@0.0.0 start G:\P-2020S1\Advanced Master Project\Program\Back-end\SWEN
90013-2020-PS
> node app.js

server is running on port 3001!
(node:1172) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated
, and will be removed in a future version. To use the new Server Discover and Monitoring eng
ine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
(node:1172) DeprecationWarning: collection.ensureIndex is deprecated. Use createIndexes inst
ead.
```

After successfully running the server, we will start to run the client.

Go to the Client folder, input

```
npm install
```

```
npm start
```



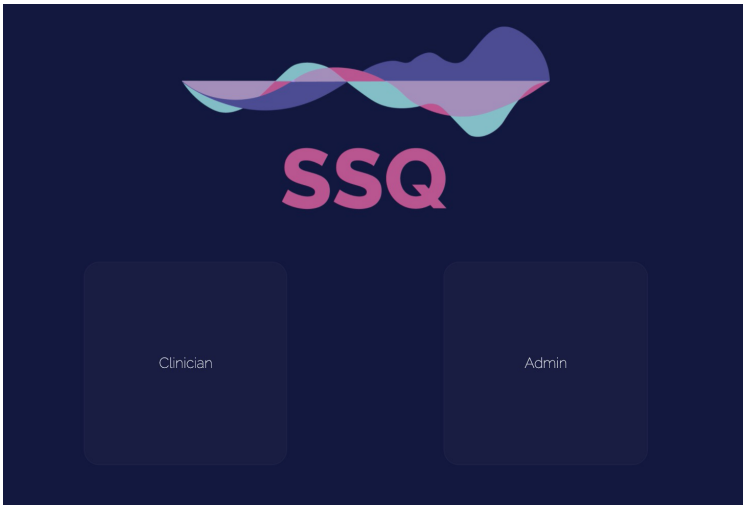
Successfully run the software locally.

Features developed so far

Please refer to Product Backlog.

User Manual

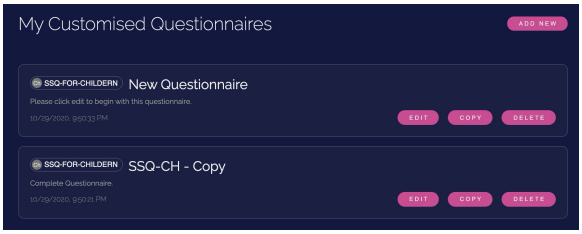
Go to https://ssq-client.herokuapp.com



Clinician

1. Log in or register
2. Clinicians can tailor their questionnaires by clicking 'Copy' which creates a copy of the standard questionnaire created by admin. Or add a new one and work from scratch

3. From the 'My Customised Questionnaires' area, clinicians and edit, copy (duplicate) or delete questionnaires.

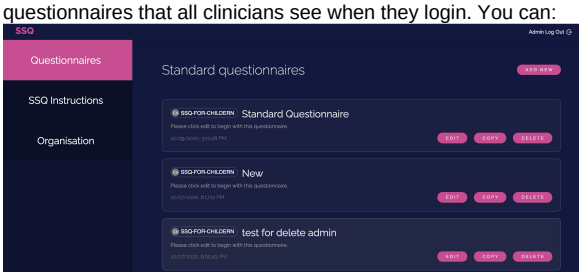


4. See steps (3) and (4) of the admin instructions to the right for edit questionnaire instructions.

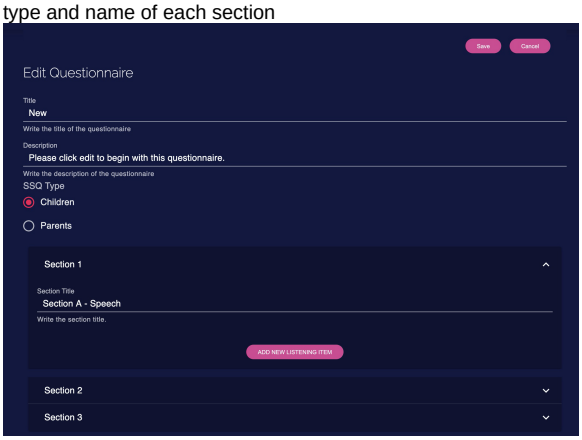
5. Conduction questionnaires - Clinicians can facilitate their client to complete the questionnaire remotely. They click 'Share a Questionnaire' and choose which one to share.
    a. A dialogue pops up to enter the client's email, a message, which sections, a checkbox for recipient to

Admin

1. Login with admin credentials
2. As admin, you are the caretaker of the standardised questionnaires that all clinicians see when they login. You can:

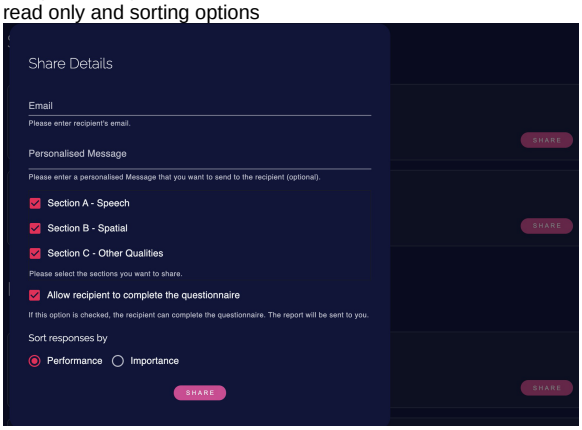


    a. Add new one and create from scratch
    b. edit an existing
    c. Copy (duplicate) one and edit from there
    d. Delete
3. When editing you can set the questionnaire name, description type and name of each section



4. You create the scenario, the labels on either end of the range slider and the possible answers in multi-choice questions

complete the questionnaire or leave unchecked for read only and sorting options

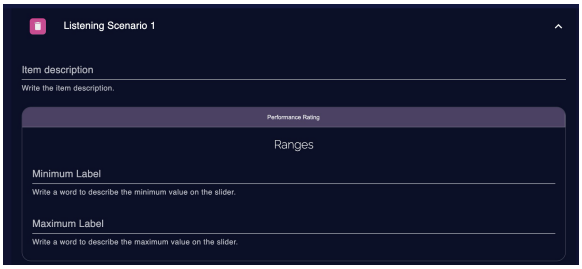


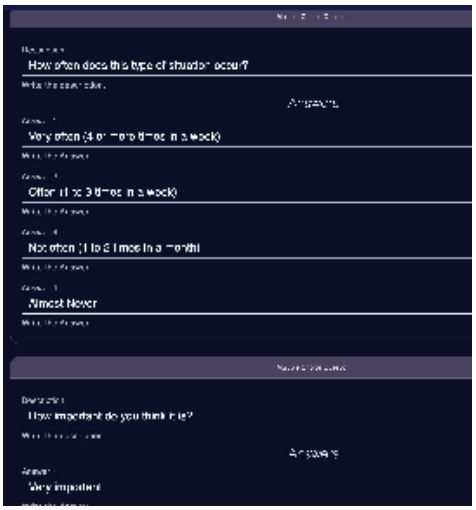
b. The recipient will then receive it in their email, complete it and a PDF report and CSV data is emailed back to the clinicians email address.

6. Conducting questionnaires - Clinicians can conduct the questionnaire themselves (either in person or over the phone) by clicking 'Start a Questionnaire' in the left pane.

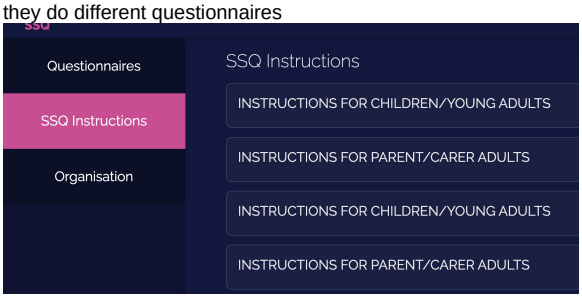
   Note: If a clinician wants to conduct a questionnaire over the phone, they may still want their client to follow along. The clinician can then go to 'Share a Questionnaire' and will have the option of emailing a read-only version.

   a. After submission, the PDF report and CSV data is emailed back to the clinician





5. You can edit the instructions that different users will see when they do different questionnaires



6. You can also see the various organisations that clinicians that have registered are affiliated with.