QualOpt

A Web App to Support Qualitative Research

Dinal Malaka Wanniarachchi and Fraser Lewis-Smith

Supervised by Dr. Kelly Blincoe and Dr. Rashina Hoda

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

Intro

This part IV project seminar is for QualOpt: A web app to support qualitative research.

QualOpt started off as a summer project at the end of 2016 and had an initial prototype had been developed.

Background - GitHub

- **GitHub** is a web development platform which supports Version Control for software development.
- GitHub has many users (24 Million+ users).
- Large potential for software research.
- Researchers are interested in obtaining participants from GitHub for their studies and surveys.
- Retrieving data is limited by the GitHub API Request limit.

Background - GHTorrent

- The **GHTorrent** project collects massive amounts of data from GitHub (E.g. Repository information, Commits etc.).
- GHTorrent creates publicly available data sets of this GitHub data and updates it daily.
- GHTorrent bypasses the limit by using multiple API keys in parallel
- Provides huge amounts of valuable data for researchers.

The Problem

The Problem - Email Spam and Privacy

- The GHTorrent data sets also contained personal information
  - (E.g. Email Addresses and Real names).
- Researchers used the email addresses and GitHub users complained
- GHTorrent allowed frustrated GitHub users to join a “Do-Not-Survey-List”
- This information was not easily found, and only had a small amount of users.
- Personal data was later excluded from the public data sets, but researchers are still able to obtain this data through an application.

What QualOpt Will Provide

- Allow Researchers to still use the collection of emails, with limitations.
- Filters to find more suitable candidates for research studies and minimise the total number of emails being sent.
- Emails sent through QualOpt will allow the recipients to immediately Opt-Out from future emails.

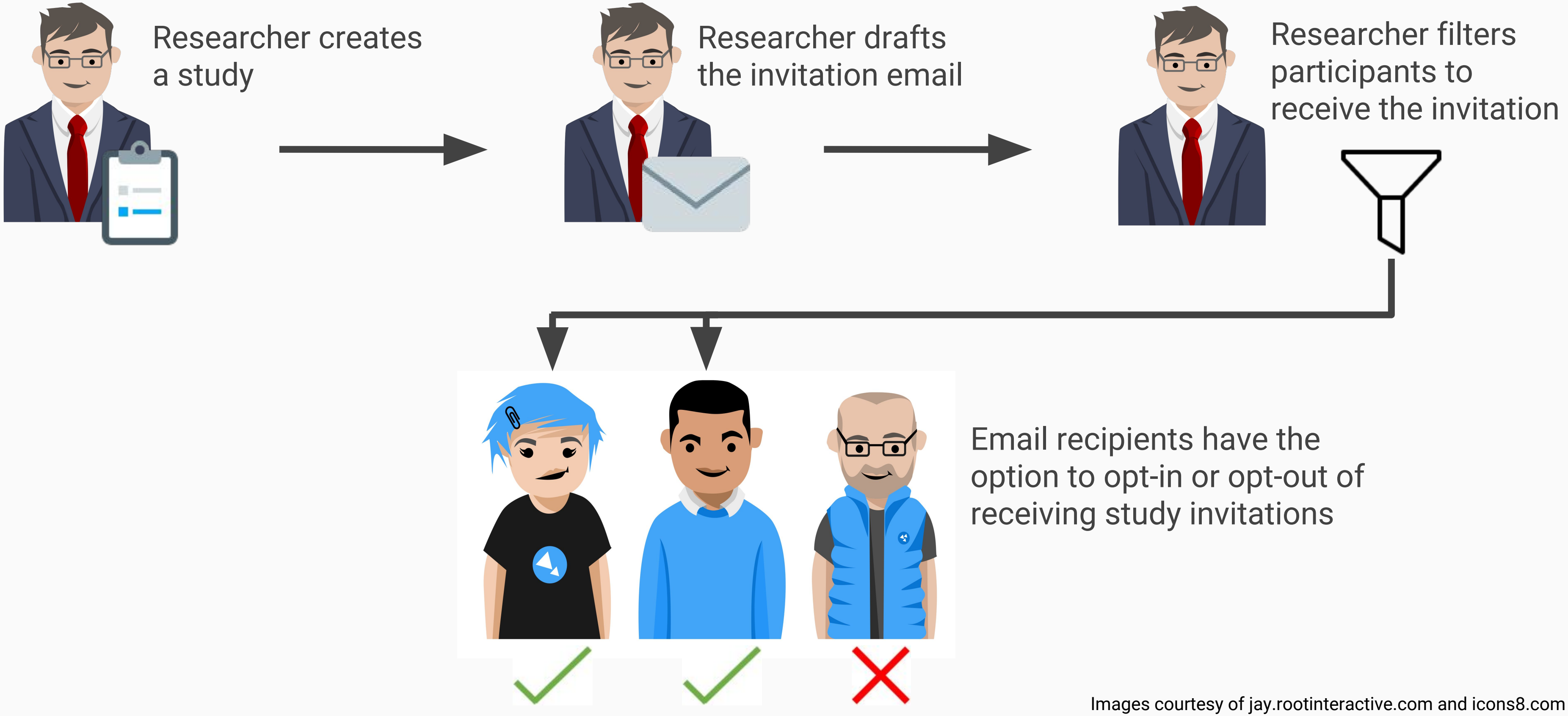
Initial Existing Implementation

- The initial prototype was separated into two parts:
  - Front-End User Interface (HTML, CSS and JavaScript)
  - Back-End Web Service (JAX-RS and MySQL Database)
- Both parts were developed on separate repositories on GitHub.
- We decided to simplify the development process by combining both parts in one repository.

Initial Existing Implementation

- The User Interface was also split into two parts:
  - Researcher side
  - GitHub User side
- Only the interface for the researcher side of the application had been developed.
- User information and data was not encrypted.
- Additional filters were needed for the researcher side.

User Story



Researcher creates a study

Researcher drafts the invitation email

Researcher filters participants to receive the invitation

Email recipients have the option to opt-in or opt-out of receiving study invitations

Images courtesy of jay.rootinteractive.com and icons8.com

Methodology

Complete review of the existing user interface of the web application.

Create a Lo-Fi prototype to follow the revisions and new features outlined in the review

Create a RESTful API to serve the data to the web application

Create a web application to consume the API and provide the user interface

Deploy the application to a web platform, along with an SQL database provider.

Perform enhancements and bugfixes in iterations, with a deployment at the end of each iteration.

Our Solution

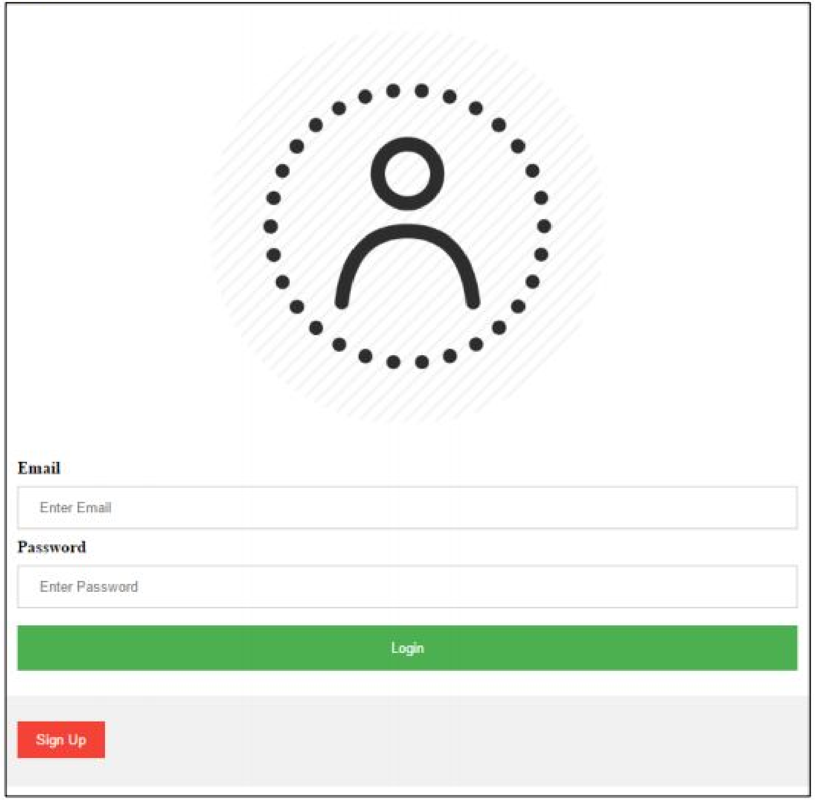
UI Review

We took the web application from the summer project and reviewed the user interface

We found that there was a lack of detail and a confusing workflow

Proposed enhancements and solutions to problems are presented in the notes under each screen

Login



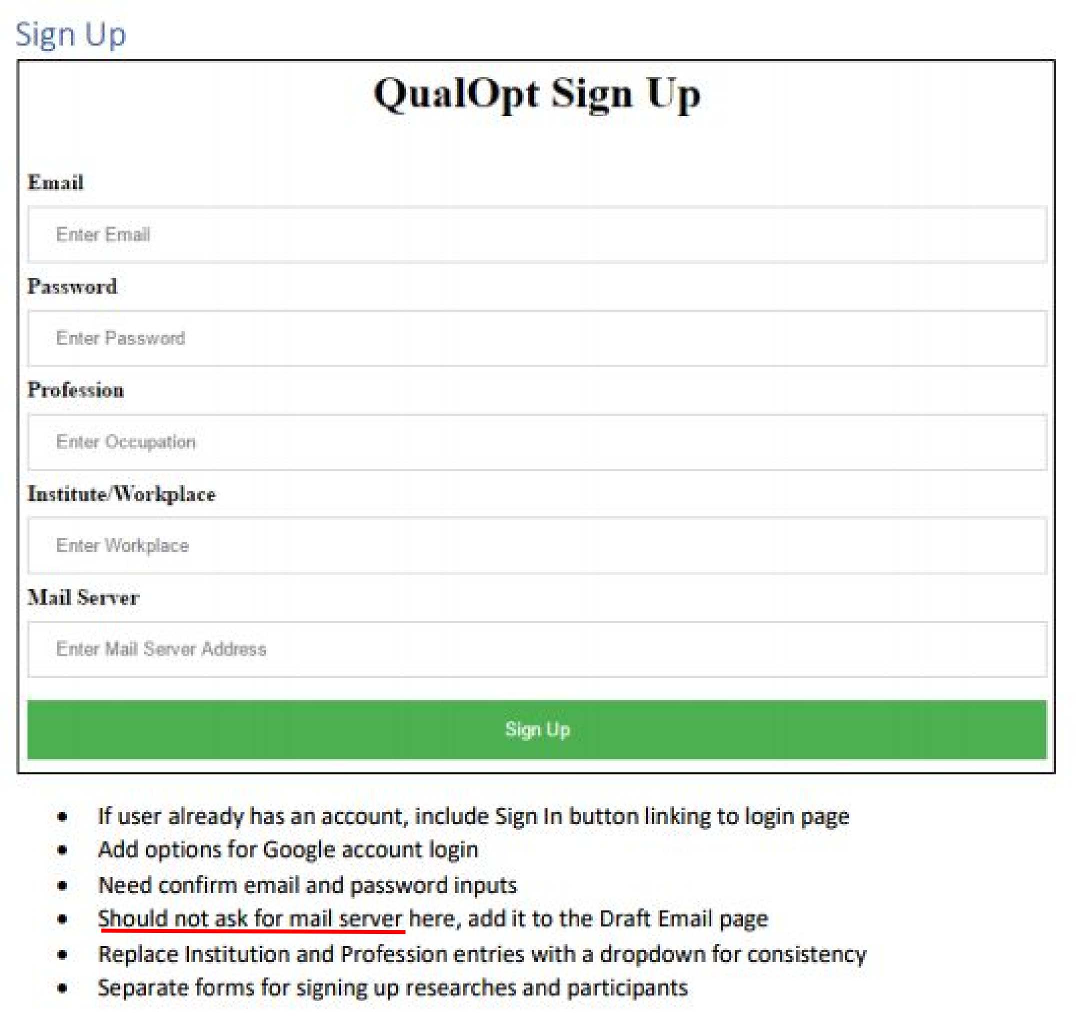
- Should include title
- Add a remember login details button
- Add a forgot password button (and subsequent security screens)
- Does there need to be unique login pages for researches and participants?
  - If they are separate, we would need another page like a portal for any user-type and then allow them to sign-up and register as either a Researcher or Participant

UI Review

We took the web application from the summer project and reviewed the user interface

We found that there was a lack of detail and a confusing workflow

We proposed enhancements and solutions to problems in the notes under each screen



Sign Up

QualOpt Sign Up

Email

Enter Email

Password

Enter Password

Profession

Enter Occupation

Institute/Workplace

Enter Workplace

Mail Server

Enter Mail Server Address

Sign Up

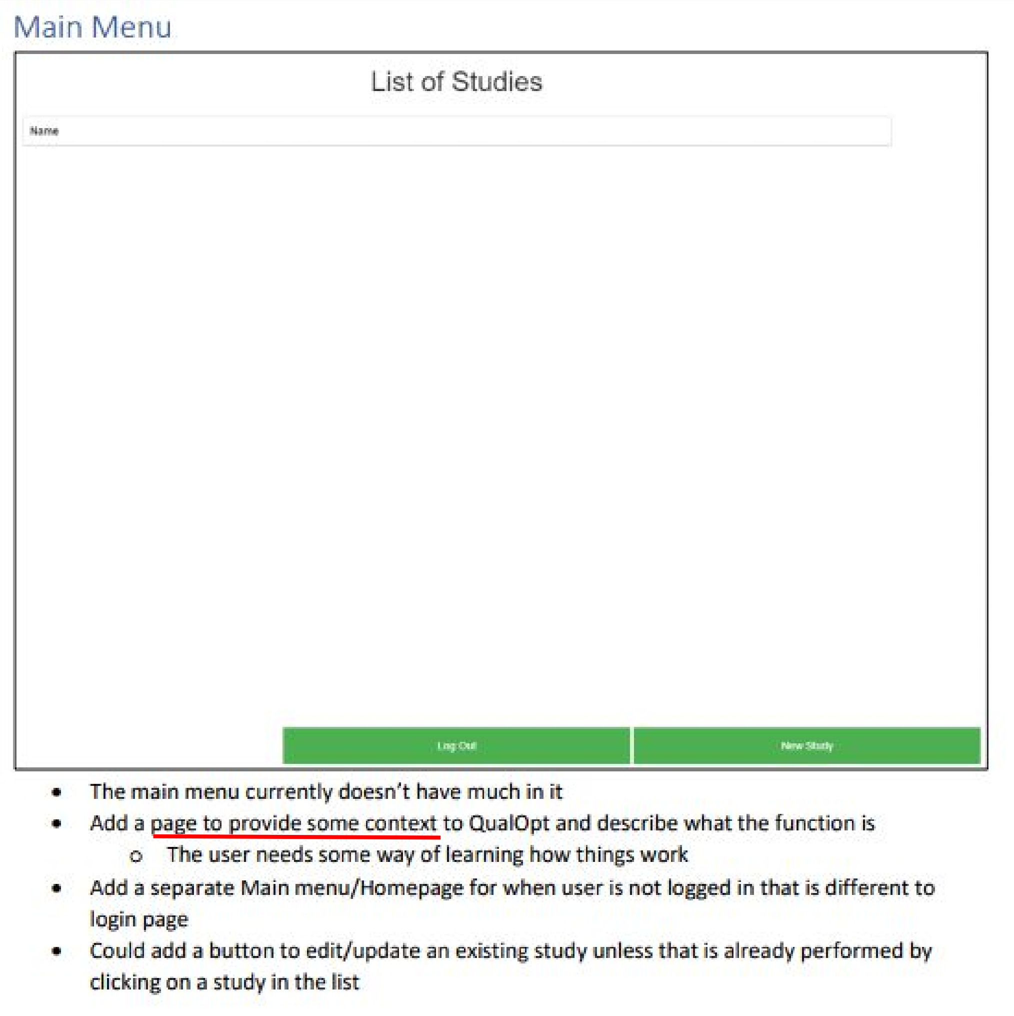
- If user already has an account, include Sign In button linking to login page
- Add options for Google account login
- Need confirm email and password inputs
- Should not ask for mail server here, add it to the Draft Email page
- Replace Institution and Profession entries with a dropdown for consistency
- Separate forms for signing up researches and participants

UI Review

We took the web application from the summer project and reviewed the user interface

We found that there was a lack of detail and a confusing workflow

We proposed enhancements and solutions to problems in the notes under each screen



Main Menu

- The main menu currently doesn't have much in it
- Add a page to provide some context to QualOpt and describe what the function is
  - The user needs some way of learning how things work
- Add a separate Main menu/Homepage for when user is not logged in that is different to login page
- Could add a button to edit/update an existing study unless that is already performed by clicking on a study in the list

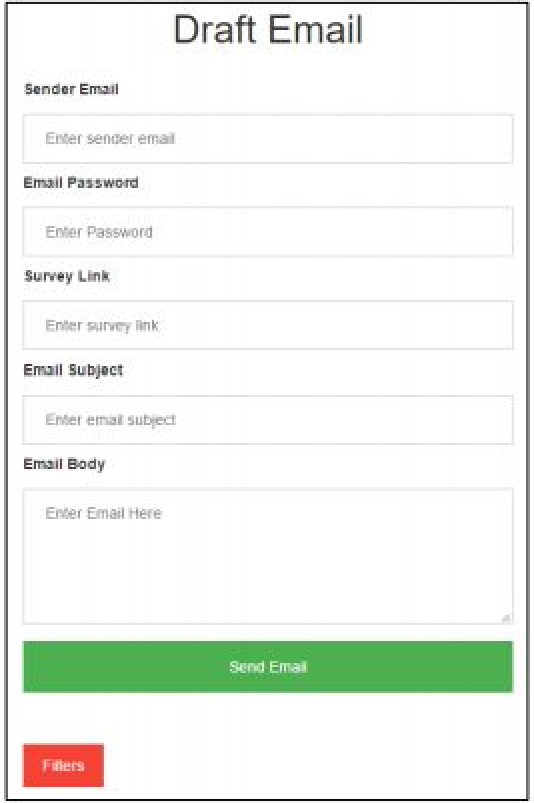
UI Review

We took the web application from the summer project and reviewed the user interface

We found that there was a lack of detail and a confusing workflow

We proposed enhancements and solutions to problems in the notes under each screen

Email Draft



Draft Email

Sender Email

Enter sender email

Email Password

Enter Password

Survey Link

Enter survey link

Email Subject

Enter email subject

Email Body

Enter Email Here

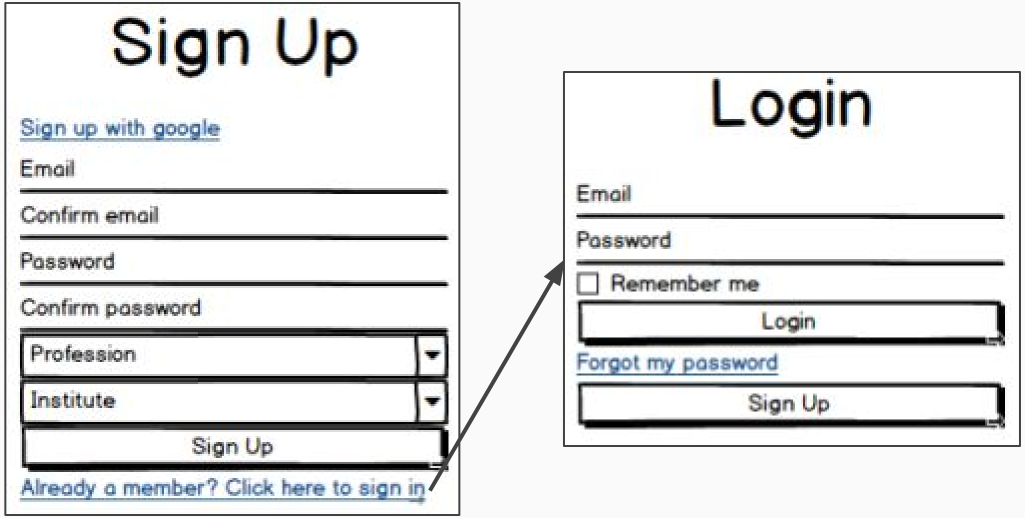
- Link to page only on Study Summary after researcher has created the study and has selected study summary
- Save repetitive input like mail server, so user does not have to retype it every time.
- Remove email password
  - QualOpt should be sending the email without using the user's personal email address.

Lo-Fi Prototype

From the feedback we gathered in the UI review we constructed a Lo-Fi prototype for the application

We used Balsamiq, a popular mockup tool to create the prototype

The goal of this prototype was to cover all the changes proposed in the review so we can implement them in the development stages

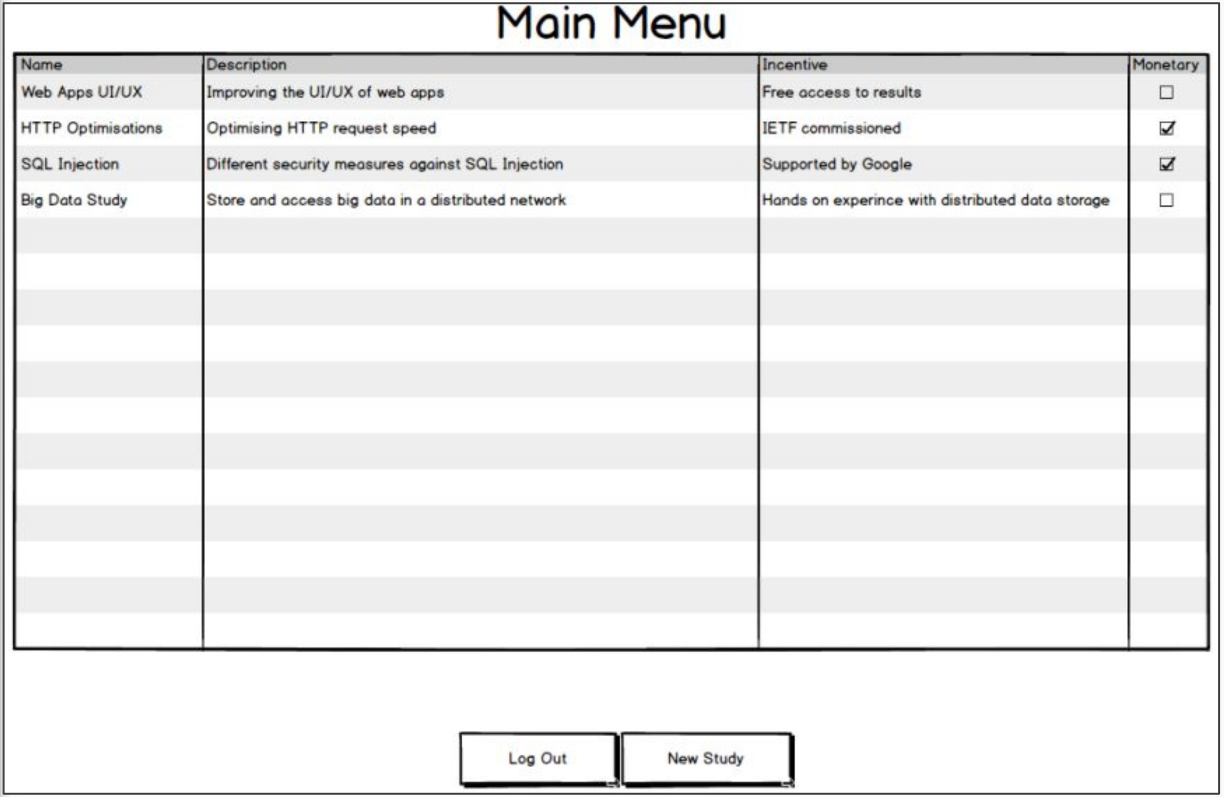


Lo-Fi Prototype

From the feedback we gathered in the UI review we constructed a Lo-Fi prototype for the application

We used Balsamiq, a popular mockup tool to create the prototype

The goal of this prototype was to cover all the changes proposed in the review so we can implement them in the development stages



Main Menu

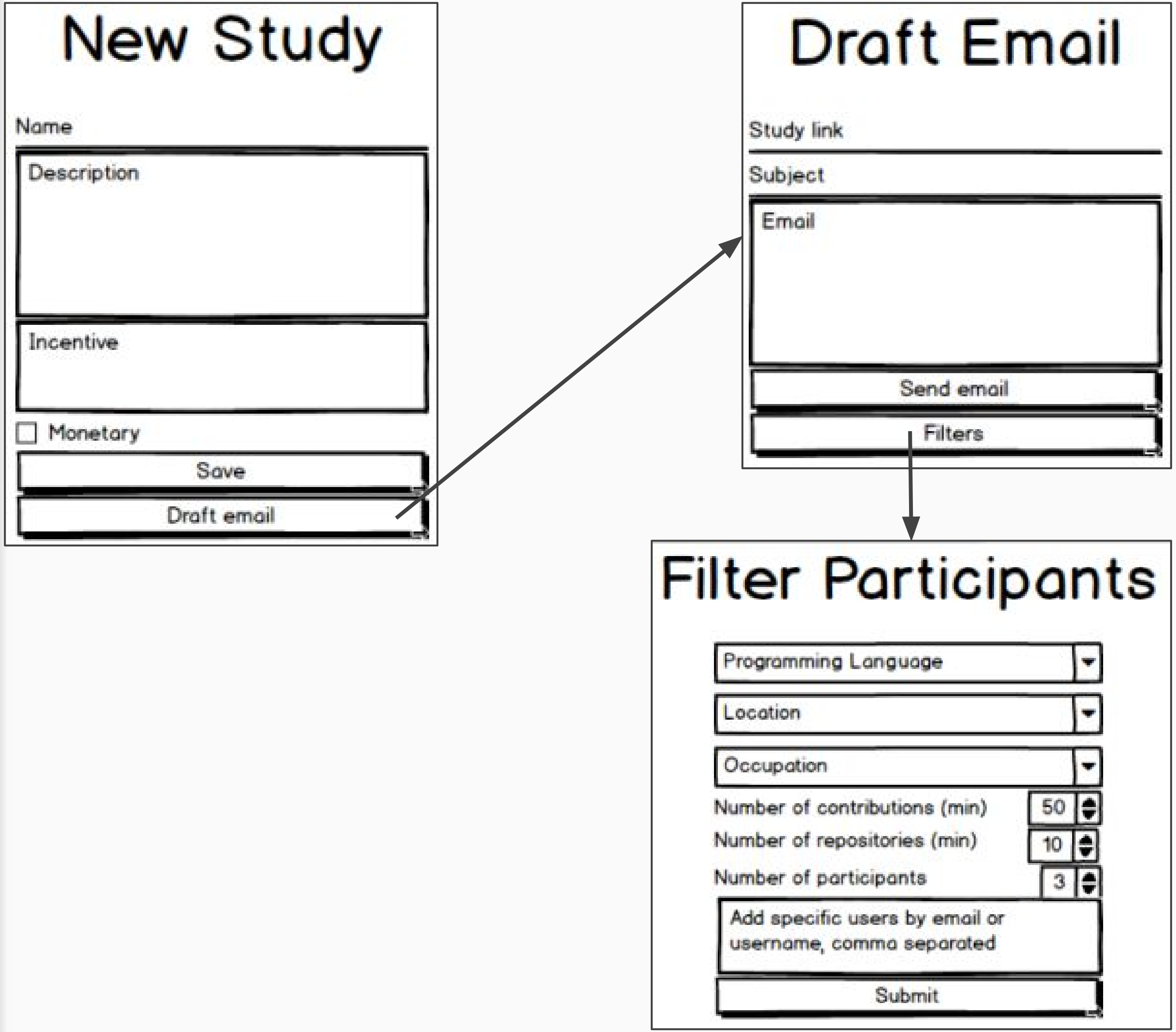
Name	Description	Incentive	Monetary
Web Apps UI/UX	Improving the UI/UX of web apps	Free access to results	□
HTTP Optimisations	Optimising HTTP request speed	IETF commissioned	☑
SQL Injection	Different security measures against SQL Injection	Supported by Google	☑
Big Data Study	Store and access big data in a distributed network	Hands on experince with distributed data storage	□

Lo-Fi Prototype

From the feedback we gathered in the UI review we constructed a Lo-Fi prototype for the application

We used Balsamiq, a popular mockup tool to create the prototype

The goal of this prototype was to cover all the changes proposed in the review so we can implement them in the development stages



New Study

Name

Description

Incentive

□ Monetary

Save

Draft email

Draft Email

Study link

Subject

Email

Send email

Filters

Filter Participants

Programming Language

Location

Occupation

Number of contributions (min) 50

Number of repositories (min) 10

Number of participants 3

Add specific users by email or username, comma separated

Submit

JHipster

JHipster is a development platform to generate, develop and deploy Spring Boot + Angular Web applications

We used JHipster as an initial scaffold for our project, allowing rapid development through generators

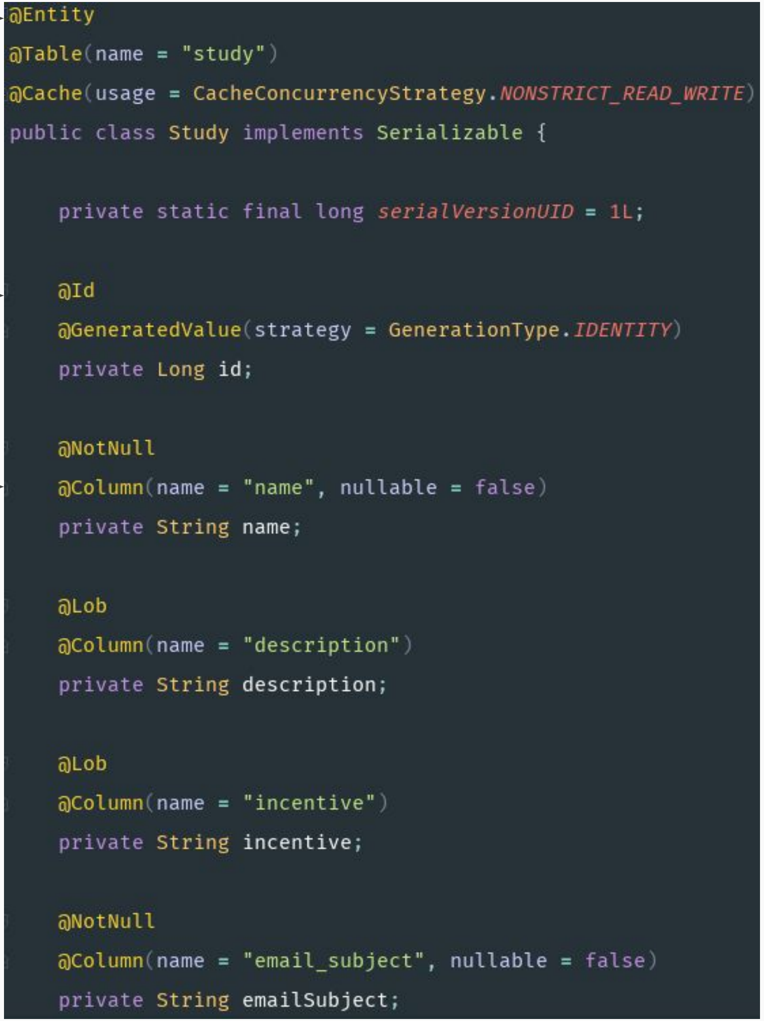
This scaffold was suitable for our application, allowing us to focus on the usability

Back-End

Spring is a modular application framework for Java based around the principle of dependency injection, an inversion of control pattern which improves independence and modifiability

We use Spring Boot which simplifies the bootstrapping and development of a Spring application

Using Spring improves the integration of our modules for **data persistence**, web services and security



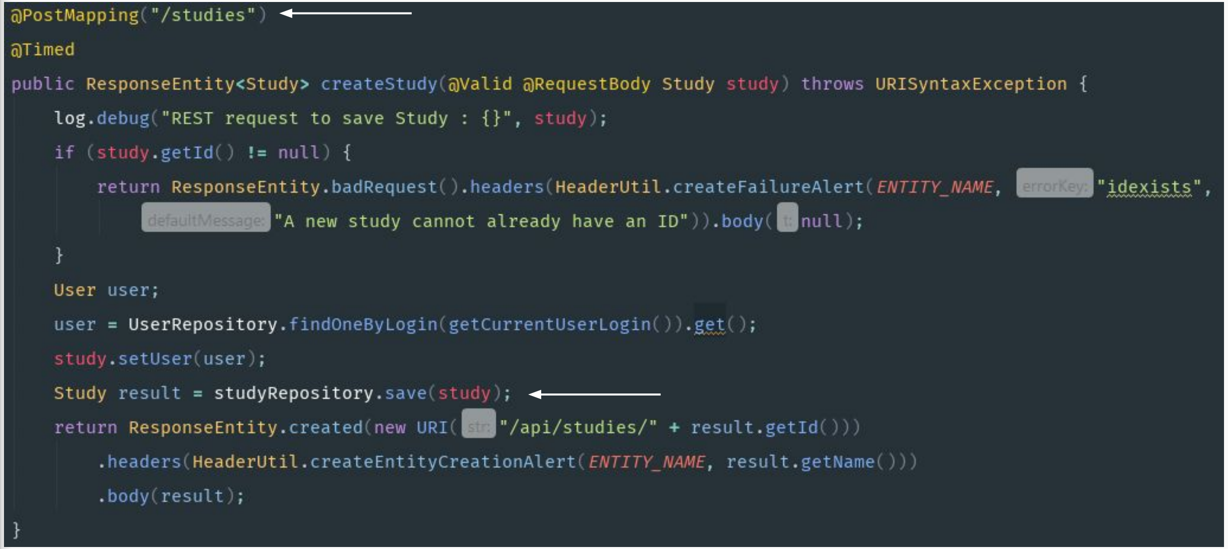
```
@Entity
@Table(name = "study")
@Cache(usage = CacheConcurrencyStrategy.NONSTRICT_READ_WRITE)
public class Study implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @NotNull
    @Column(name = "name", nullable = false)
    private String name;

    @Lob
    @Column(name = "description")
    private String description;

    @Lob
    @Column(name = "incentive")
    private String incentive;

    @NotNull
    @Column(name = "email_subject", nullable = false)
    private String emailSubject;
```

Back-End

Spring is a modular application framework for Java based around the principle of dependency injection, an inversion of control pattern which improves independence and modifiability

We use Spring Boot which simplifies the bootstrapping and development of a Spring application

Using Spring improves the integration of our modules for data persistence, **web services** and security



```
@PostMapping("/studies")
@Timed
public ResponseEntity<Study> createStudy(@Valid @RequestBody Study study) throws URISyntaxException {
    log.debug("REST request to save Study : {}", study);
    if (study.getId() != null) {
        return ResponseEntity.badRequest().headers(HeaderUtil.createFailureAlert(ENTITY_NAME, errorKey: "idexists",
            defaultMessage: "A new study cannot already have an ID")).body( t: null);
    }
    User user;
    user = UserRepository.findOneByLogin(getCurrentUserLogin()).get();
    study.setUser(user);
    Study result = studyRepository.save(study);
    return ResponseEntity.created(new URI( str: "/api/studies/" + result.getId()))
        .headers(HeaderUtil.createEntityCreationAlert(ENTITY_NAME, result.getName()))
        .body(result);
}
```

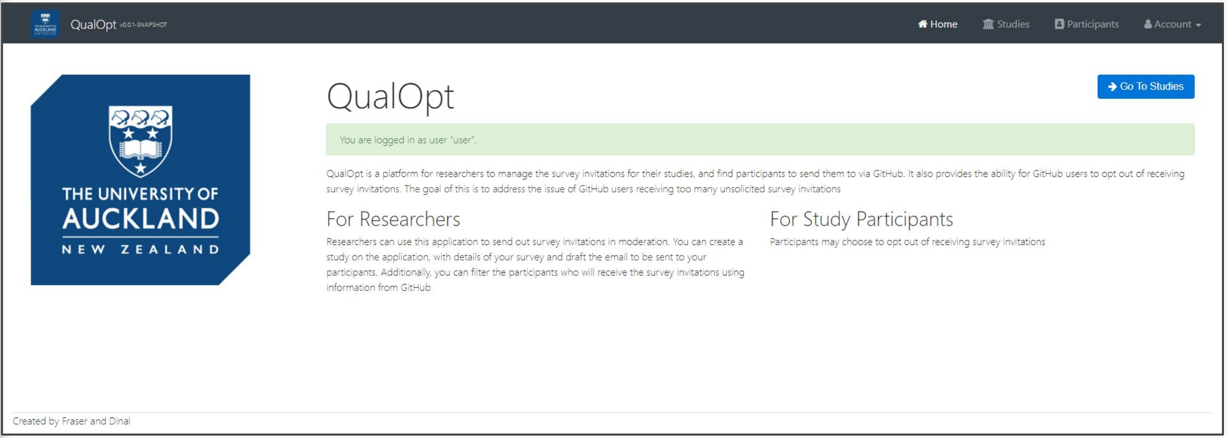
Front-End

Angular is a component-based framework created by Google for building client applications in HTML and TypeScript

We use Angular for our client side code which consumes the REST API

Bootstrap is a front-end component CSS library created by Twitter

We use Bootstrap to handle the responsive design of our front-end



QualOpt

You are logged in as user "user".

QualOpt is a platform for researchers to manage the survey invitations for their studies, and find participants to send them to via GitHub. It also provides the ability for GitHub users to opt out of receiving survey invitations. The goal of this is to address the issue of GitHub users receiving too many unsolicited survey invitations

For Researchers

Researchers can use this application to send out survey invitations in moderation. You can create a study on the application, with details of your survey and draft the email to be sent to your participants. Additionally, you can filter the participants who will receive the survey invitations using information from GitHub

For Study Participants

Participants may choose to opt out of receiving survey invitations

Created by Fraser and Dinal

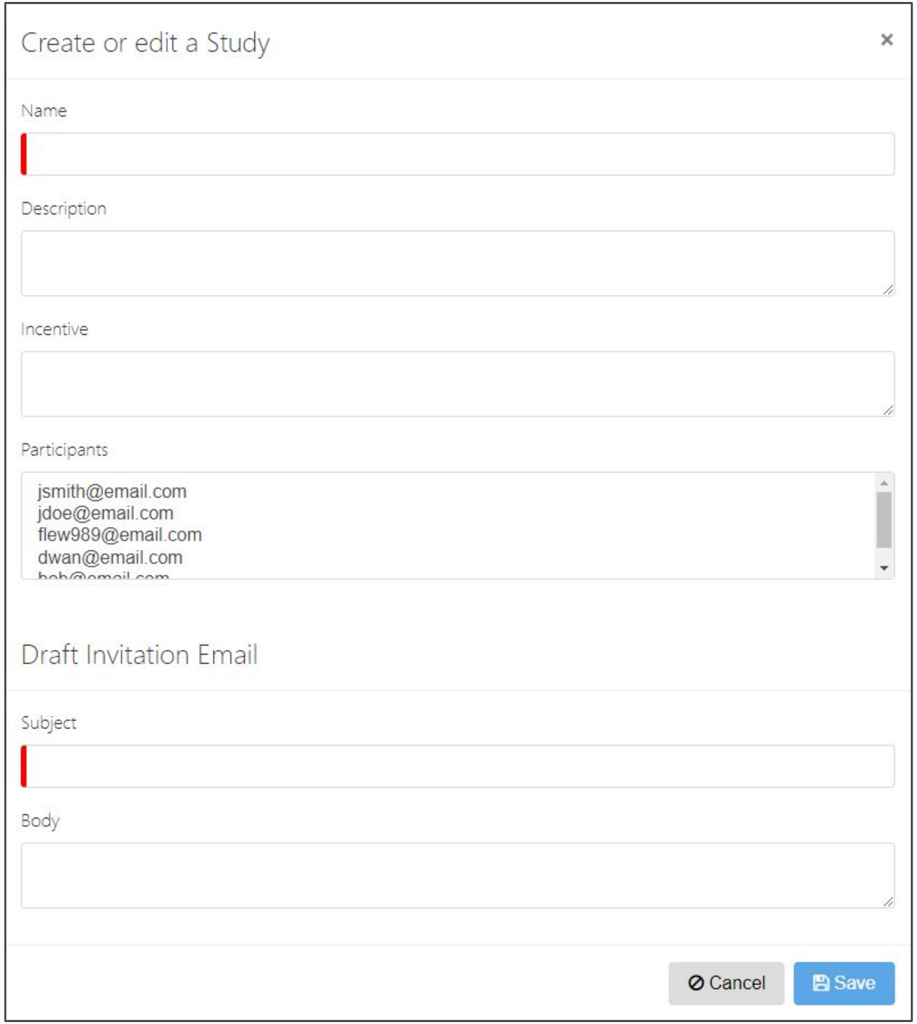
Front-End

Angular is a component-based framework created by Google for building client applications in HTML and TypeScript

We use Angular for our client side code which consumes the REST API

Bootstrap is a front-end component CSS library created by Twitter

We use Bootstrap to handle the responsive design of our front-end



Create or edit a Study

Name

Description

Incentive

Participants

jsmith@email.com
jdoe@email.com
flew989@email.com
dwan@email.com
bob@email.com

Draft Invitation Email

Subject

Body

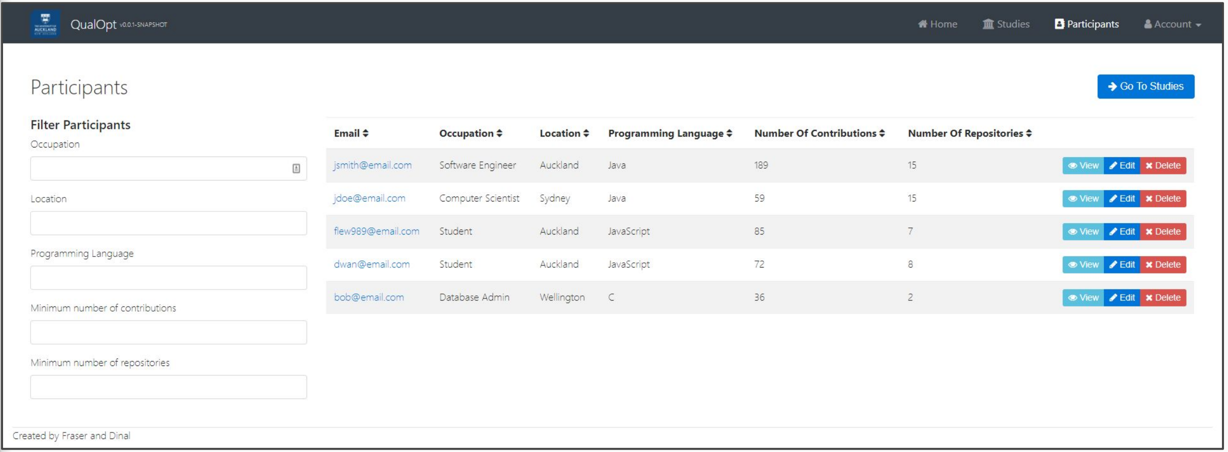
Front-End

Angular is a component-based framework created by Google for building client applications in HTML and TypeScript

We use Angular for our client side code which consumes the REST API

Bootstrap is a front-end component CSS library created by Twitter

We use Bootstrap to handle the responsive design of our front-end



Participants

Filter Participants

Occupation

Location

Programming Language

Minimum number of contributions

Minimum number of repositories

Email	Occupation	Location	Programming Language	Number Of Contributions	Number Of Repositories
jsmith@email.com	Software Engineer	Auckland	Java	189	15
jdoe@email.com	Computer Scientist	Sydney	Java	59	15
flew989@email.com	Student	Auckland	JavaScript	85	7
dwan@email.com	Student	Auckland	JavaScript	72	8
bob@email.com	Database Admin	Wellington	C	36	2

Created by Fraser and Dinal

Security

HTTP session authentication is a stateful mechanism storing the session_id in a cookie, the server performs all the authentication and authorisation. The cookie is mirrored in the database to prevent cookie theft

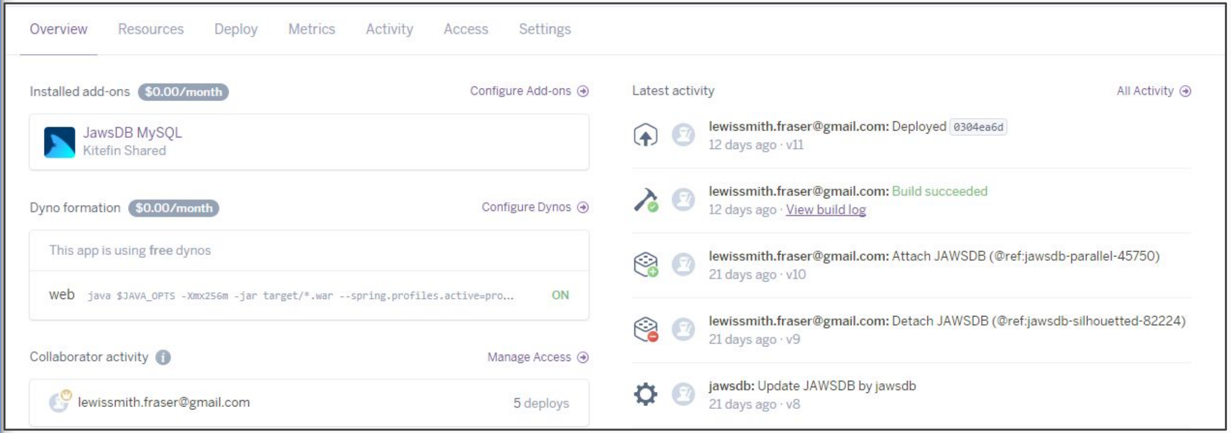
Cross-site scripting (XSS) is the injection of malicious client-side code. Angular prevents values entering the domain object model (DOM) by treating all values as untrusted and sanitizing and escaping them

Deployment

We chose to use Heroku as our web hosting service because it has the best support for the development tools we used

It was important for us to have the simplest pathway to hosting the application as possible

Jaws DB is an addon for Heroku which provides an SQL database, storing our server-side data



Installed add-ons $0.00/month

JawsDB MySQL
Kitefin Shared

Dyno formation $0.00/month

This app is using free dynos

web `java $JAVA_OPTS -Xmx256m -jar target/*.war --spring.profiles.active=pro...` ON

Collaborator activity

lewissmith.fraser@gmail.com 5 deploys

Latest activity

lewissmith.fraser@gmail.com: Deployed `0304ea6d`
12 days ago · v11

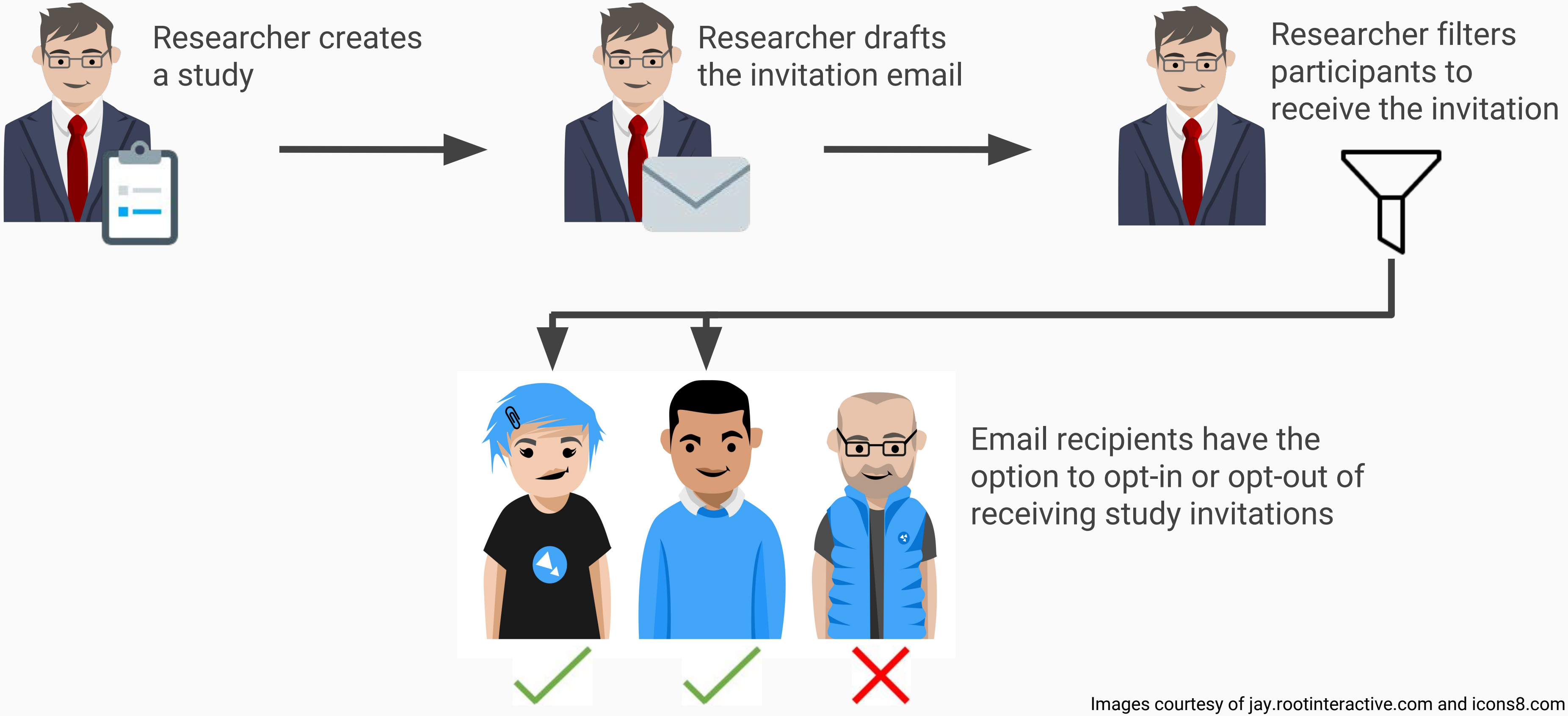
lewissmith.fraser@gmail.com: Build succeeded
12 days ago · View build log

lewissmith.fraser@gmail.com: Attach JAWSDB (@ref:jawsdb-parallel-45750)
21 days ago · v10

lewissmith.fraser@gmail.com: Detach JAWSDB (@ref:jawsdb-silhouetted-82224)
21 days ago · v9

jawsdb: Update JAWSDB by jawsdb
21 days ago · v8

Usability Study



Researcher creates a study

Researcher drafts the invitation email

Researcher filters participants to receive the invitation

Email recipients have the option to opt-in or opt-out of receiving study invitations

Images courtesy of jay.rootinteractive.com and icons8.com

Results

- Successful redesign of the client-side user interface
- Full-stack web application developed using cutting-edge technology and modern frameworks
- Created an online platform for researchers to manage their studies and study participants
- Potential for open-source collaboration in the future

Thanks!

Questions?

Dinal Malaka Wanniarachchi
dwan133@aucklanduni.ac.nz

Fraser Lewis-Smith
flew989@aucklanduni.ac.nz

Department of Electrical and Computer Engineering

https://qual-opt.herokuapp.com/

Image courtesy of negativespace.co