Test strategy

This document will summarise the group testing strategy at a high level.

Product overview:

Citrus-app is a Customer Relationship Management tool which has been designed for use predominately as a personal tool for a specific client. It is a web application which has been built using Javascript and, more specifically, using Node.js/Express on the backend and React on the frontend. Given this is a student project with the developers having fairly limited web development experience and limited expertise in secure app development and UI/UX design, the main focus while testing will be ensuring that most of the critical requirements of the project have been met.

Overall test strategy:

- Passfinders will use several test modalities to ensure relatively comprehensive testing at both the software and user level; as described in the documents describing Integration Testing, User Acceptance Testing, and End-to-End Testing, noting that some Unit Testing has also been undertaken.
- All software tests can be found within the “/test” directory of our Github repo.

Test environment

- Testing to be primarily undertaken using devices of the team and the client, which introduces limitations as to determining the functionality of the application on different operating systems given we were mostly using MacOS.
- Given more time, we could also have planned for undertaking testing across a greater range of devices and OSs.

Testing tools:

- Mocha and Chai frameworks will be used to support some of our software testing.
- Tests run automatically when pushing code to main to ensure that no code added to main will break the build and to facilitate this we used a Github Actions pipeline.
- For UAT and E2E Testing, we have constructed our own sets of criteria and forms based loosely on template references online.

Risk analysis:

As above due to the inexperience and lack of specialisation of the team, there is considerable risk associated with both subtle bugs in the code and of issues to do with the security of the our application, which might not be apparent at the time that the product is handed over to the client. To an extent, we have mitigated this risk, by making use of very popular frameworks and tools in our tech stack which have been thoroughly road tested by large numbers of developers.

Record of testing:

Date	Testing completed	Notes
19/10/2021	Software (unit and integration tests run)	Successful build after tests automatically run via Github Actions pipeline.
18/10/2021	E2E testing	Most critical tests passed, with a couple of minor patches required.
18/10/2021	UAT testing	Most critical tests passed, with a couple of minor patches required. Limitations noted in terms of only one UAT assessment performed, although this did help us to make note of what needed patching. Most of the absolutely critical features at time of UAT were functional and since then any deficiencies have largely been patched except for generate notifications about imminent events functionality.
21/10/2021	E2E	On repeat, all E2E test criteria now successful.