

Appendix E

User Manuals

This software is a prototype system to manage the online safety of someone who has an elected supervisor to help them decide if they should click on an email link. Although a two-person system, this version runs completely from one machine for demonstration purposes. This guide is for anyone wishing to run the system on their personal machine to interact with the system themselves. Please note this prototype has only been configured for Windows (see E.1.1 below). Google Gemini (2025a) was used to inform what to include in this user manual.

WARNING. Please note, the system remains in a development stage so may exhibit unexpected or unintended behaviour and is not fit for actual deployment. **This is not an executable. This installation is for demonstration and development purposes only.** The browser extension performs online activity monitoring but does not export this data outside of the system, although security and privacy of communications within the system is yet to be audited and reinforced. **The software could have unintentional effects, install at your own risk.**

E.1 Installation Guide

E.1.1 Minimum System Requirements

Operating System: Windows 10 / Windows 11

Storage Available: 500MB

RAM: 4GB

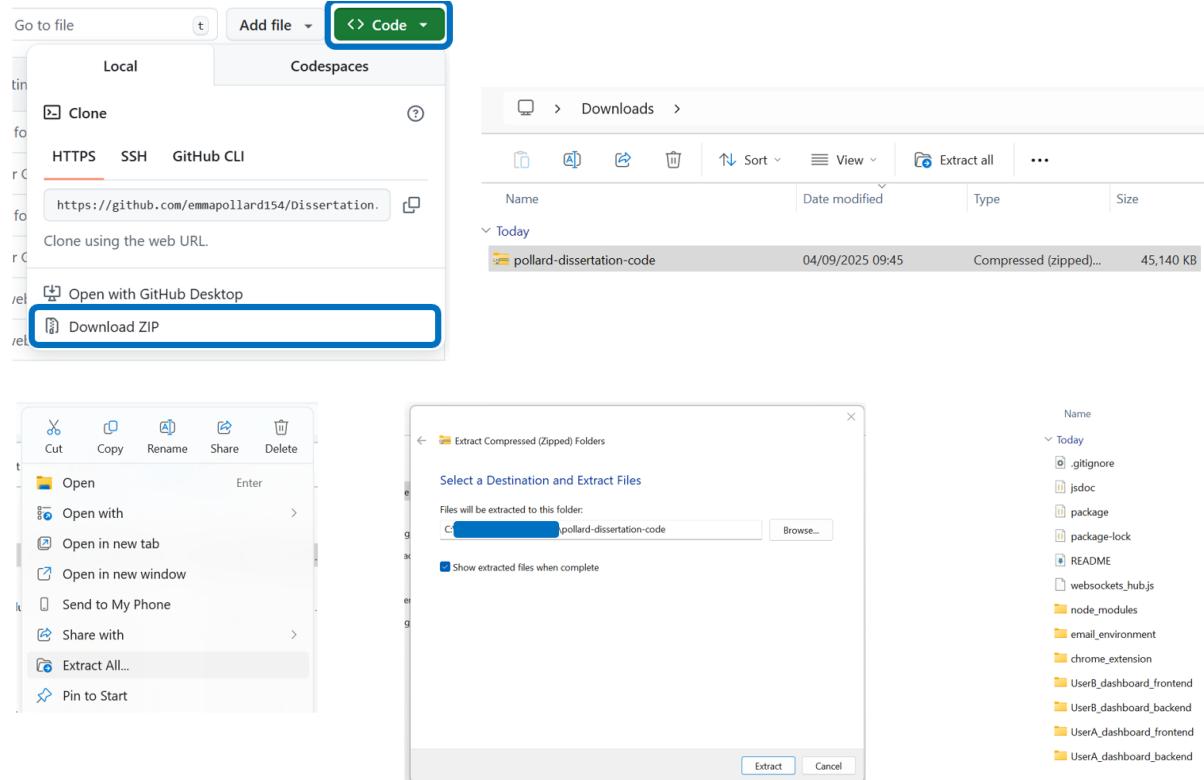
Processor: Intel Core i5

Prerequisite software:

- Visual Studio Code (version 1.80 or above, with preferred JavaScript, React, and HTML debuggers) - <https://code.visualstudio.com/>
- Google Chrome (latest version) - https://www.google.com/intl/en_uk/chrome/
- Node.js (v22.7 or above) - <https://nodejs.org/en>

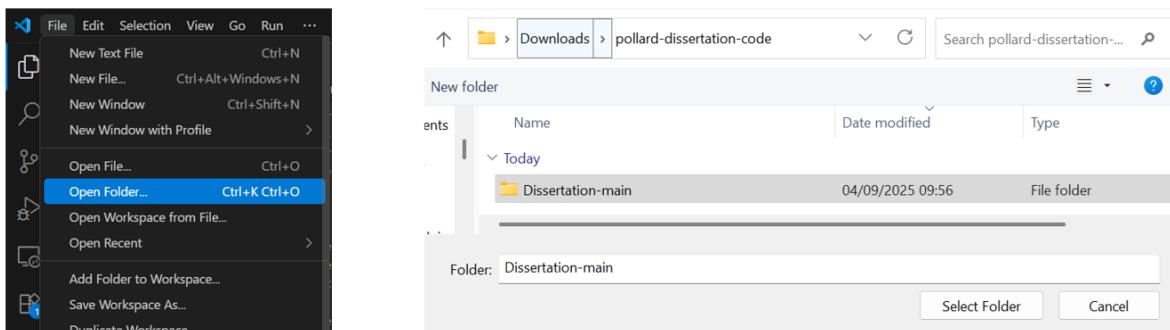
E.1.2 Download

Download the zipped project submitted alongside this. Alternatively, download the zip from <https://github.com/emmapollard154/Dissertation.git> (main branch). Unzip the folder to a directory of your choice and you should be able to see the code files and directories as shown below. **Note: directory names and file paths may appear differently.**



E.1.3 Configuration

Open VSCode. Select 'File' -> 'Open Folder'. Select the unzipped folder you just downloaded.



The port numbers are already configured but, should issues arise, see Section E.2.5.

OPTIONAL: If you would like to set up email notifications for User B, you will need to create an EmailJS account <https://www.emailjs.com/> and create an appropriate email template. To activate the email system, fill in the public key, service ID, template ID, and destination email address you have configured in EmailJS (these will be visible on the EmailJS dashboard)

and uncomment the sendAlertEmail() function calls in App.jsx (User A).

```

App.jsx M X
UserA_dashboard_frontend > src > App.jsx > ...
40  * Temporary email address for User B notifications.
41  * @global
42  * @type {String}
43  * @deprecated since version 1.0. Must be updated.
44  */
45  const TEMP_EMAIL = '<FILL IN EMAIL ADDRESS>'
46 /**
47  * Public key for EmailJS.
48  * @global
49  * @type {String}
50  * @deprecated since version 1.0. Must be updated.
51  */

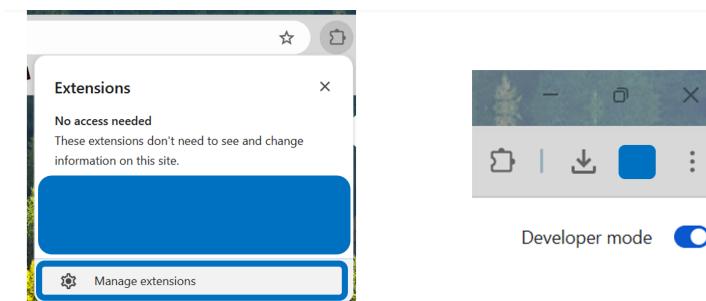
52 const PUBLIC_KEY = '<FILL IN PUBLIC KEY>';
53 /**
54  * Service ID for EmailJS.
55  * @global
56  * @type {String}
57  * @deprecated since version 1.0. Must be updated.
58  */
59 const SERVICE_ID = '<FILL IN SERVICE ID>';
60 /**
61  * Template ID for EmailJS.
62  * @global
63  * @type {String}
64  * @deprecated since version 1.0. Must be updated.
65  */
66 const TEMPLATE_ID = '<FILL IN TEMPLATE ID>';
67

socket.on('a_choice', (data) => {
  console.log('App.jsx (A): User A made choice: ', data);
  // sendAlertEmail(); // remove comment if EmailJS account has been created and configured
  fetchActionData();
});

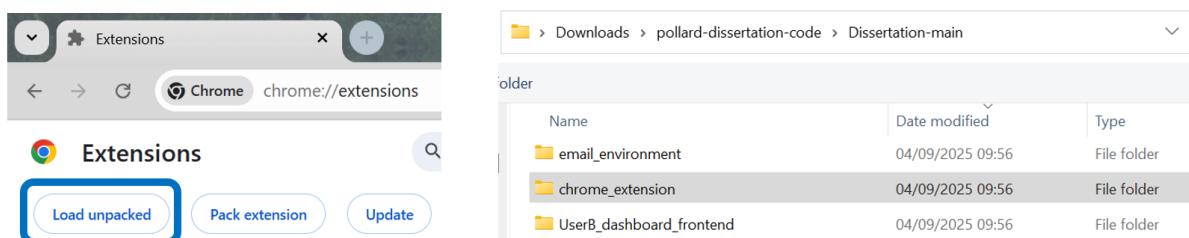
socket.on('a_message', (data) => {
  console.log('App.jsx (A): User A sent message: ', data);
  // sendAlertEmail(); // remove comment if EmailJS account has been created and configured
  fetchMessageData();
});

```

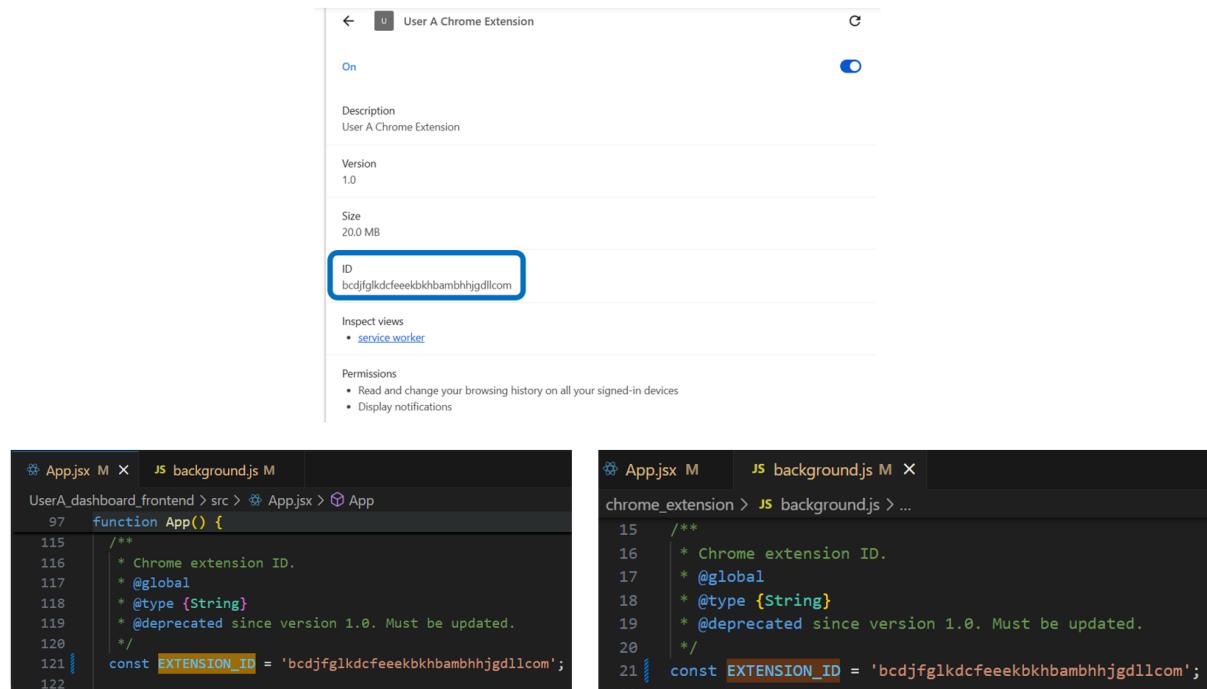
Open a Chrome browser and click the extensions icon -> 'Manage extensions'. Ensure that developer mode is enabled.



Then click 'Load unpacked' and select the 'chrome_extension' directory from the project folder.



Find the extension ID of the browser extension and update this value in App.jsx (User A) and background.js.

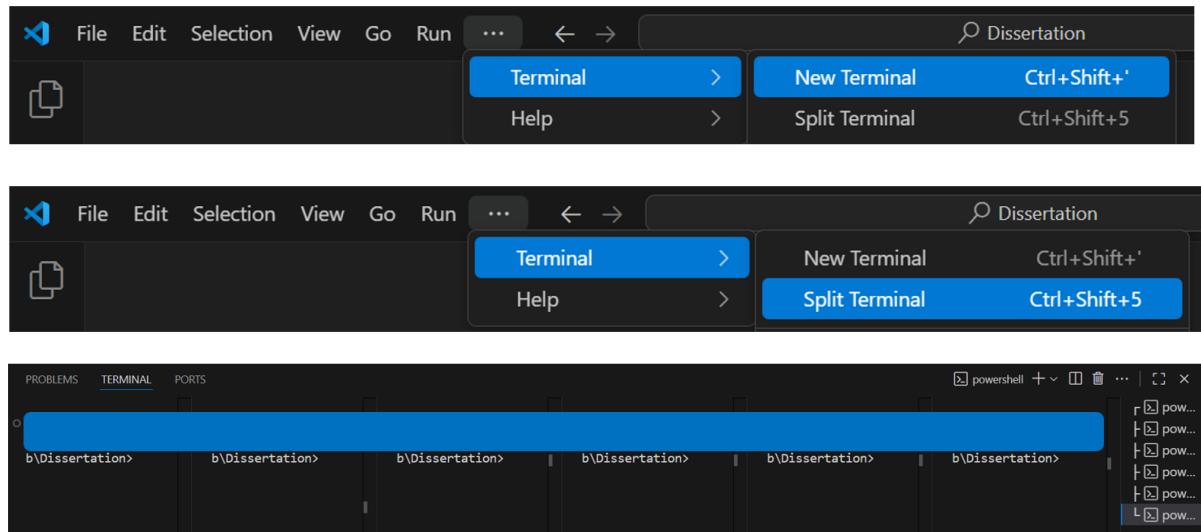


Optional: pin the Chrome extension to the toolbar.

E.2 Operational Manual

E.2.1 Launch

In VSCode (with the project directory open as in Section E.1.3). Select '...' -> 'Terminal' -> 'New Terminal'. Then select '...' -> 'Terminal' -> 'Split Terminal' and repeat this last step until you have six terminals at the root of the project directory.



Navigate to the necessary directories in the terminals by running the following commands:

1. *Do nothing to remain at root of the project directory*
2. `cd .\UserA_dashboard_backend\`

3. cd .\UserB_dashboard_backend\
4. cd .\UserA_dashboard_frontend\
5. cd .\UserB_dashboard_frontend\
6. cd .\email_environment\



Run the following commands from each terminal respectively to launch the system:

1. node .\websockets_hub.js
2. node .\server.js
3. node .\server.js
4. npm run dev
5. npm run dev
6. npm run dev

```

PS C:\Dissertation>UserB_dashboard_backend> node .\server.js
Backend server (B) running on http://localhost:8080
server.js (B): connected to the database.
server.js (B): connected to websockets hub.

PS C:\Dissertation>UserA_dashboard_backend> node .\server.js
Backend server (A) running on http://localhost:5000
server.js (A): connected to the database.
server.js (A): browsingHistory table created / already exists.
server.js (A): requests table created / already exists.
server.js (A): trusted table created / already exists.
server.js (A): settings table created / already exists.
server.js (A): message table created / already exists.
server.js (A): action table created / already exists.
server.js (A): connected to websockets hub.

PS C:\Dissertation> node .\websockets_hub.js
websockets_hub.js: hub running on http://localhost:9000
websockets_hub.js: new backend connected - 62n9w5cpclsCFdVLeAAAB
websockets_hub.js: 62n9w5cpclsCFdVLeAAAB registered as server_A
websockets_hub.js: new backend connected - vzw-zfXq8MV4ewKmAAAD
websockets_hub.js: vzw-zfXq8MV4ewKmAAAD registered as server_b

PS C:\Dissertation>UserA_dashboard_frontend> npm run dev
> usera-dashboard-frontend@0.0.0 dev
> vite

VITE v6.3.5 ready in 261 ms
→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help

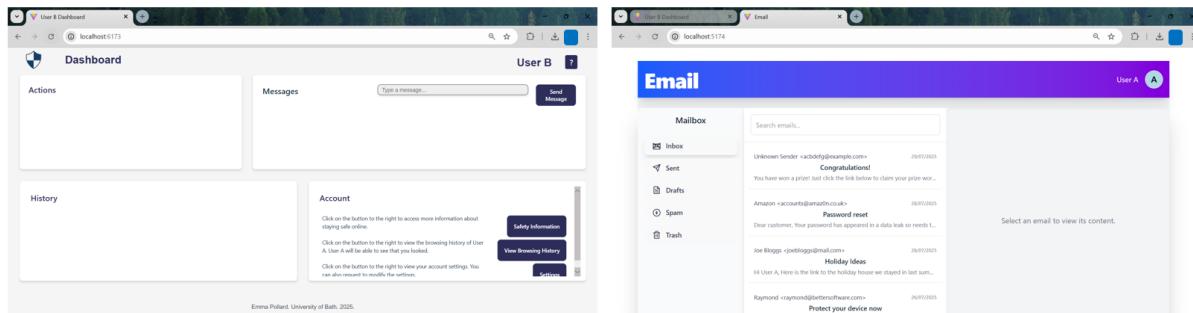
PS C:\Dissertation> npm run dev
> email_environment@0.0.0 dev
> vite

VITE v6.3.5 ready in 248 ms
→ Local: http://localhost:5174/
→ Network: use --host to expose
→ press h + enter to show help

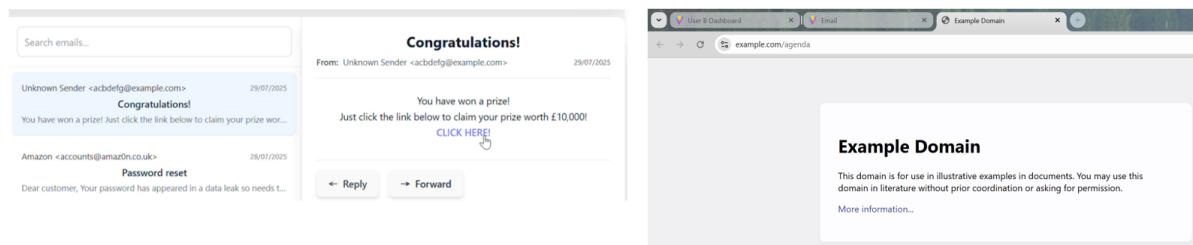
PS C:\Dissertation>UserB_dashboard_frontend> npm run dev
> userb-dashboard-frontend@0.0.0 dev
> vite

VITE v6.3.5 ready in 234 ms
→ Local: http://localhost:6173/
→ Network: use --host to expose
→ press h + enter to show help
  
```

In a Chrome browser, navigate to localhost:6173 and localhost:5174 in separate tabs to open the User B dashboard and email webpage respectively (or whichever port numbers you have assigned to B_FRONTEND and EMAIL_PORT).



The email links should open an example website.



Open the browser extension installed in Section E.1.3 to open the User A dashboard in a new tab.



E.2.2 Functionality

Tip: move the User B dashboard to a separate window.

When opening the extension, you will be taken to the User A dashboard welcome screen (if the system is unused, as on initial download). Configure the settings you want to allow and add any trusted contacts. It is recommended to add 'joebloggs@mail.com' as a trusted contact to demonstrate the functionality.

Welcome

Select the options you want to allow (email).

- Continue (no interference).
- Record action for User B too see later. Continue with action.
- Ask User B for advice (accept / reject) regarding this action. Pause action.
- Ask User B for advice (accept / reject) regarding this action. Pause action and permanently disable link if rejected.
- Block this action yourself (disable link). User B will not be informed.

Intervention will be activated when you click any link contained in an email. If you choose to involve User B, they will be able to see the time of your request and the link you want to access.

Save

Add trusted contacts

Add

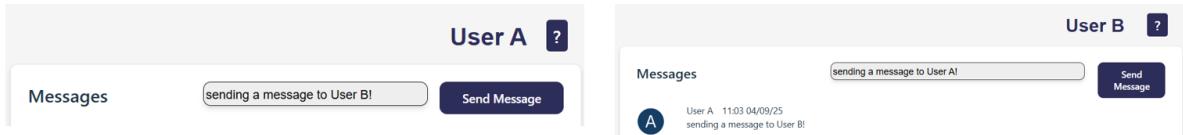
joebloggs@mail.com **Remove**

Okay

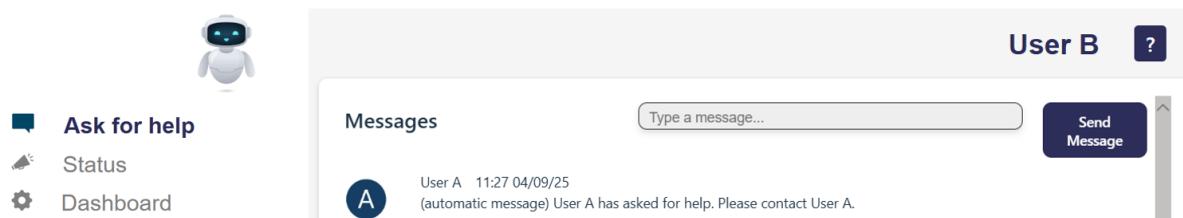
The following content illustrates a range of possible actions to demonstrate the functionality of the system.

Dashboard Actions

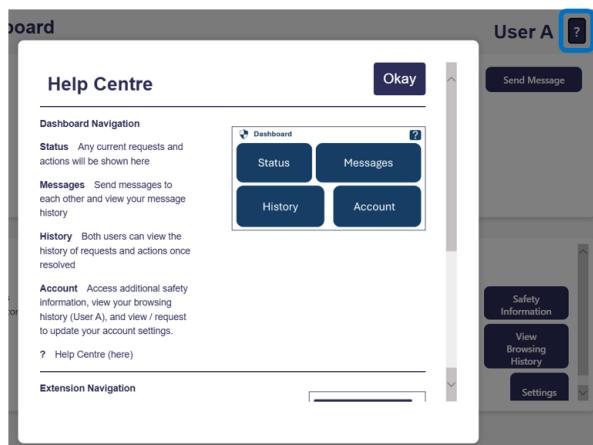
Send a custom message. Each user can type a message and send it to the other user.



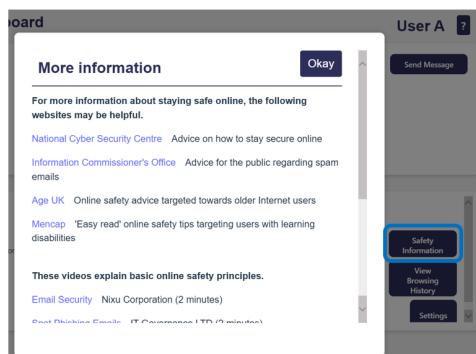
Send an automatic message. User A can send an automatic message asking for help to User B.



Visit the help centre. Each user can get information about the system by clicking the '?' in the top right corner of the dashboard.



See additional online safety resources. Each user has access to additional online safety information.



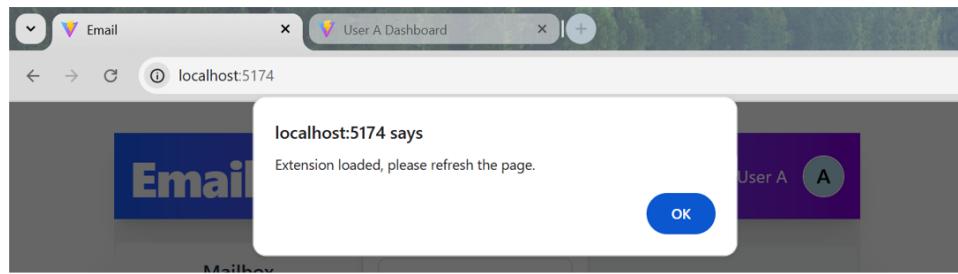
View User A browsing history. Each user can view the browsing history of User A, but when User B has a look this is added to the action history visible to both users.

URL	Date
https://www.bbc.co.uk	14:17 21/08/25
https://www.google.com	14:17 21/08/25
https://www.ecosia.org	14:16 21/08/25
https://example.com	14:16 21/08/25
https://www.dictionary.com	14:14 21/08/25
https://www.thesaurus.com	14:14 21/08/25

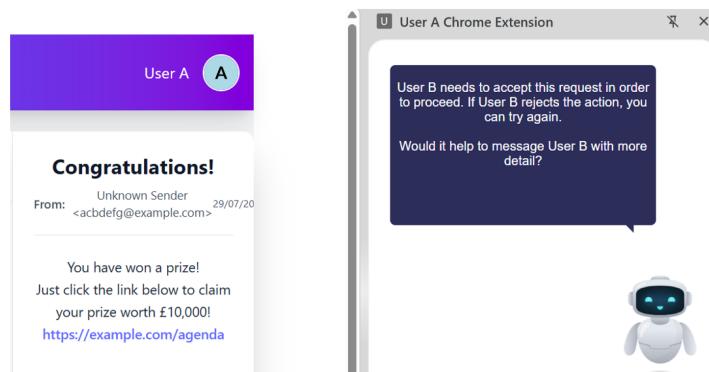
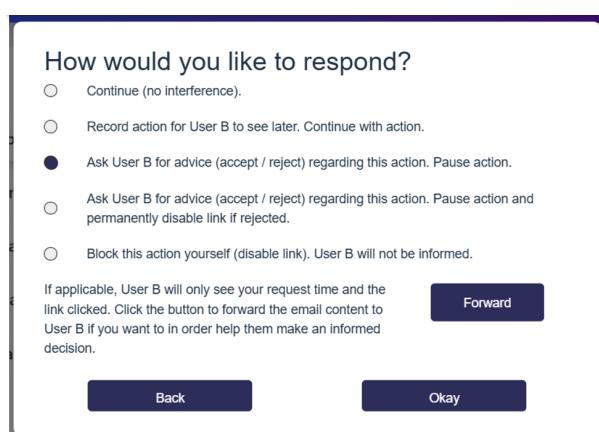
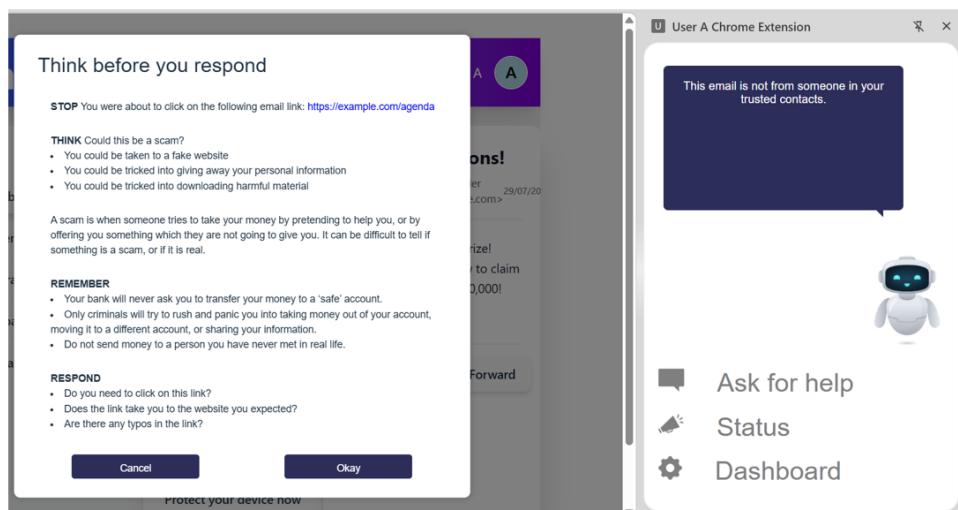
Update settings. Each user can request to update settings. It is intended for this to be performed when both users are together and discuss the options available. Only from the User A dashboard can the settings be updated.

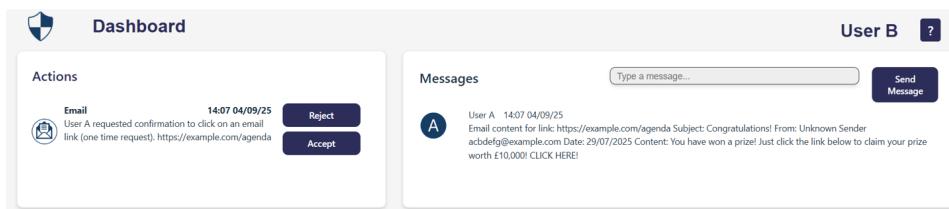
Update trusted contacts. Each user can request to update trusted contacts. It is intended for this to be performed when both users are together and discuss the options available. Only from the User A dashboard can the settings be updated.

Email Actions



Click on an untrusted link. User A can choose to include User B in the decision making process and choose to forward the email content.





User A can also choose to resolve the action independently.

How would you like to respond?

- Continue (no interference).
- Record action for User B to see later. Continue with action.
- Ask User B for advice (accept / reject) regarding this action. Pause action.
- Ask User B for advice (accept / reject) regarding this action. Pause action and permanently disable link if rejected.
- Block this action yourself (disable link). User B will not be informed.

If applicable, User B will only see your request time and the link clicked. Click the button to forward the email content to User B if you want to in order help them make an informed decision.

Forward **Back** **Okay**

Congratulations!

From: Unknown Sender <acbdefg@example.com> 29/07/20

You have won a prize!
Just click the link below to claim your prize worth £10,000!
<https://example.com/agenda>

<- Reply **→ Forward**

You chose to block this action independently.



Click on a trusted link. If an email is from a trusted contact, any links will open as usual and the system will not intervene.

User A **A**

Holiday Ideas

From: Joe Bloggs <joebloggs@mail.com> 28/07/2025

Hi User A,
Here is the link to the holiday house we stayed in last summer.
holidayhomes.co.uk.
Joe

<- Reply **→ Forward**

User A Chrome Extension

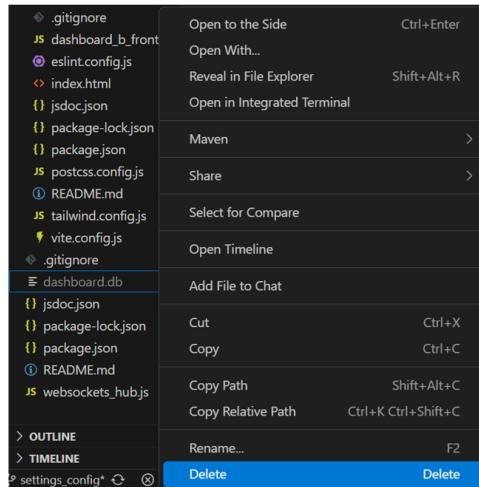
This email is from someone in your trusted contacts.



Ask for help

E.2.3 Restart and Shutdown Procedures

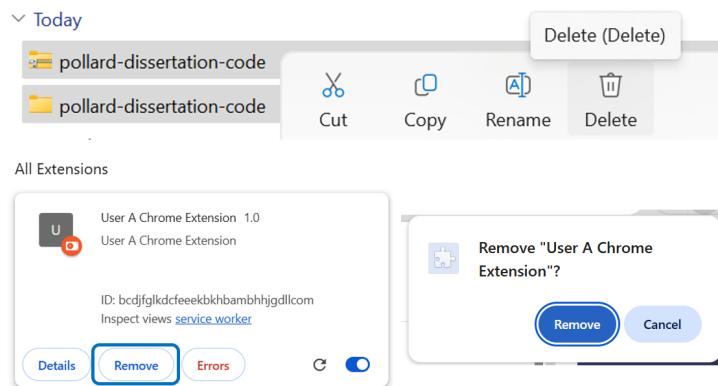
To restart the system, close the side panel and kill (Ctrl + C) terminals 1 - 3 (server terminals). Delete dashboard.db (the database). Restart terminals 1 - 3 using the commands in Section E.2.1 (ie. node <filename>). Refresh each of the webpages in the browser and reopen the browser extension.



To shutdown, close the side panel and all tabs. Kill (Ctrl + C) all terminals. Delete the database dashboard.db if you want delete system history. Otherwise, you can return to the configured system by rerunning the commands in Section E.2.1 and the history will be restored.

E.2.4 Uninstallation

Follow the instructions in Section E.2.3 to shutdown the system then simply delete the project directory from your device and remove the extension from the Chrome extension manager webpage.



E.2.5 Troubleshooting

Port Numbers

It is possible that (some of) the default port numbers for the dashboards, servers, and email webpage are already in use by your machine. If, when trying to run the system, you get an error message saying that a specific port is already in use, you can reconfigure the system. Choose a large number that is not already reserved (https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers). Reassign this value to each instance

of the corresponding variable in the code (Ctrl + Shift + F, then search for the variable name). The self-descriptive variable names are as follows: A_FRONTEND, B_FRONTEND, A_BACKEND, B_BACKEND, HUB_PORT, and EMAIL_PORT.

Simultaneous Requests

The current system can not triage multiple request resolutions simultaneously. For example, if User B accepts two email links in close succession, User A may only get the opportunity to access the most recently resolved via the side panel. This is a flaw in the system design. You may have to resubmit a request as User A and resolve it as User B to access a link should this occur. When multiple requests are pending, the email webpage may cease loading the appropriate HTML (another bug in the current prototype). It is recommended to refresh the email webpage at this point to clear the history, but this does leave the issue of the system action history not synchronising with the email webpage.

Packages, Imports, and Dependencies

The system should contain the correctly configured dependencies when downloaded (for a Windows operating system), contained in each of the node_modules directories in each project subdirectory. If you get an import or dependency issue, it is best practice to delete the node_modules directory and package-lock.json file. Then run npm install to reinstall the dependencies from the affected project subdirectory. Running npm update may also resolve issues with outdated packages.

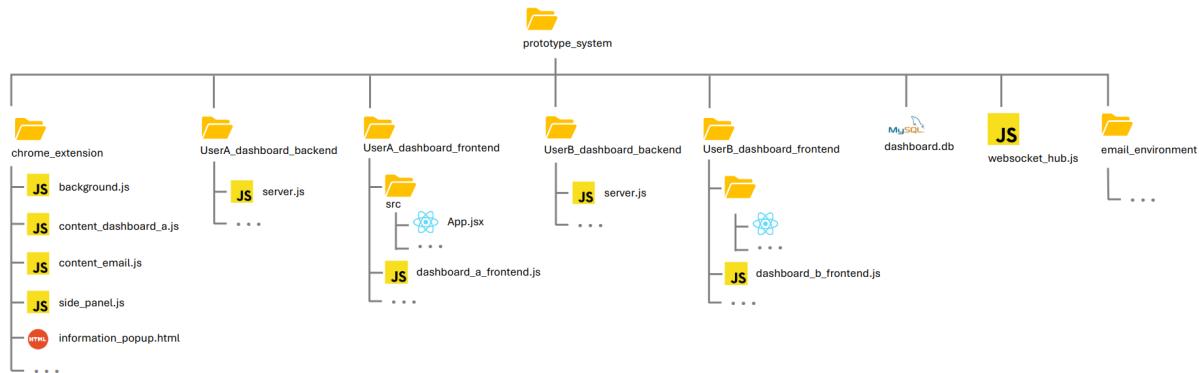
System Logs

The best way to identify and resolve problems with system behaviour is using the log files. The server terminals (1-3) should log any communications and database accesses. Webpage and Chrome extension logs can be viewed in the browser by right clicking on the chosen page/element -> 'Inspect' -> 'Console'.

E.3 Maintenance Guide

E.3.1 Codebase

Each component of the system has its own directory: the chrome extension, backends for each user, frontends for each user, and the email environment. The central WebSocket hub is in the main project directory.



E.3.2 Dependencies

The dependencies of each project subdirectory are detailed in the package.json files under 'dependencies' and 'devDependencies'.

E.3.3 Update Procedure

It is recommended to use version control software (such as GitHub) if making any modifications to the system. You can use the GitHub template available from <https://github.com/emmapollard154/Dissertation.git>. Update the documentation accordingly. You are not able to contribute to the original GitHub repository.

E.3.4 Recommended Updates

The recommended updates for the system include, but are not limited to:

- Transfer to a distributed system
- Improve the UI
- Improve stability and handling of multiple requests
- Implement alternative communication and notification systems
- Add more online contexts where safety is critical

as discussed in Section 8.