Product: DeliverED Home

Team: DeliverED

Abstract

Our product aims to eliminate the need for waiting at home for a delivery – we will provide a secure and convenient personal mailbox for contactless deliveries, controlled by an Android app. In this demo, we have mainly focused on reconfiguring our App UI to be more inclusive, developing the Webots simulation with an infrared sensor, hardware testing, and handling notifications with regards to device-app connectivity. We have also implemented the necessary code for our hardware components, such as the forklift and Photon, ready to be executed when possible. There has also been progress with developing our website, and market research. We have thoroughly considered the feedback we received from the Usability workshop, into both our App UI and our plans for the User Guide.

1. Project plan update

Our Demo 3 goals are as follows:

- Update website - Achieved
- Webots simulation (infrared sensor) - Achieved
- Update App UI with accessibility & usability considerations – Achieved
- Photon / App communication and testing - Achieved
- Gather sensor readings from hardware (forklift, locking door) - Achieved
- Available forklift in Appleton Tower - Partially Achieved

We are fully on track to where we thought we would be at this stage. In the usability workshop this week, we thought of more app UI developments, which we have considered on top of the alterations we initially planned to do after the accessibility work last demo. The hardware side is extremely reliant on Garry and technical time, though we are happy with our progress. The only small deviation is with respect to the forklift, as it is still being commissioned, however we do currently have a Lego alternative. There have been some setbacks regarding App / Device connectivity with the components we have in Appleton Tower, which are explained in the hardware section.

The hardware side of things was taken care of by Harry, Hrichika and Hallelujah. The hardware team was in constant contact with experts regarding our commissioned forklift, as well as the Arduino. The focus in terms of hardware for this demo was testing, since we are taking a modular approach and each component must be seen interacting in response to the app. Neo developed the Webots simulation to include an infrared sensor. Another focus was getting app-device connectivity with regards to notifications, which is something that was worked on by Yizhuo and Harry. The alterations made to the app UI with regards to the accessibility feedback in demo 2 was discussed by Chris, Amy and Huacheng, and then implemented by Huacheng. The developments to the website were completed by Chris and Amy. Hrichika and Chris attended the Usability workshop, which led to further app UI discussion by the design team, as well as preliminary ideas being laid out for the user guide by Amy and Chris.

During this demo we have continued managing our tasks using Trello, and have set up GitHub repositories for the individual sections of the project, for example the app and the website. These repos are within a Github organisation, which we have set up for our company DeliverEd. We have kept in constant contact through Microsoft Teams. The software and hardware teams respectively kept in contact internally almost daily, with the two teams coming together for a general update once a week. Lots of communication with experts has occurred this demo, with huge focus on hardware testing, usability, and data ethics.

In terms of budget, there have been only minor changes. We maintain that our product must include an installation cost, and the physical components we have commissioned and are using (photon, forklift, electromagnetic lock, motors, etc.) are the main sources of cost.

Everything seems on track for our demo 4 goals, and there are no significant alterations to note. Obviously, our initial demo 4 milestone of having a complete hardware device will not happen due to the modular approach we are taking. However, we will have all of the individual sections, as well as evidence of them communicating.

2. Technical details

2.1. Webots Simulation

In this week, Neo has implement additional features and embedded them into current controller.With such features, the mailbox is able to detect if a parcel has been delivered and perform relevant actions and interact better with users.

As annotated in the capture above, there are 3 major changes. At the very top of the mailbox there is a cylinder annotated with 1. This is the place to store actual controller, Photon. Placing it at top with an hump ensures a

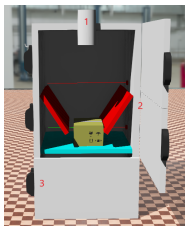

Figure 1. IR Sensor and Other Features

better capture of WiFi signal, and makes operations (e.g. maintenance, reboot...) more convenient since it's easy to reach.

Secondly, at label 2, two infrared sensors are installed, one above the red one-way-door, and one above the blue platform. The sensor above the one-way-door is to detect the existence of new parcels. If this infrared sensor recognizes a shorter distance reading (beam is blocked by parcel), the one-way-door opens and convey the parcel to the next blue platform until the beam is no longer blocked. The sensor above the blue platform performs a similar function, it indicates if the platform need to descend more to create more space for the next parcel by detecting if the beam is blocked by a parcel.

Thirdly as proposed in the last demo meeting by Steve, Neo has modified a few 3d model structure and textures. At label 3, a lock model is made and attached to each level of compartments. However, it does not have any practical usage for now since all actions are triggered via controller, there is no need to have an additional locking feature, and it's indistinguishable if it's locked or unlocked unless indicator lights are introduced. Additionally, the physical magnetic lock has already been commissioned by hardware teams and technicians. Thus, there is no need for duplicated work. The original yellow rubber duck which act as a parcel has been changed with a real 3d modeled parcel.

2.2. Hardware

One of the main goals of this demo for the hardware and software teams collaboratively was to establish communication between the app and the Photon device over the internet. With some troubleshooting, we managed to get this working on Harry's personal Photon, which is being used for testing purposes (demonstrated in our demo video). Naturally, we next tried to extend this to a Photon that was ordered into Appleton to begin the process of connecting our modular elements, but this is where we ran into a problem. The SDP-robots Wi-Fi connection cannot talk to the outside, and that includes the Particle cloud. This was a huge issue, and the only solution was to have the device communicate through Eduroam. The problem with this however is that would require a team member to send their login details to the technicians at Appleton, something we decided was not a valid solution.

We tried multiple ways to go around this issue, installing the Particle Command Line Interface in an attempt to claim the device remotely, however finding that this too required a connection to the Particle services. Our final solution in this case is to have the technicians claim the device instead and provide us with the access token and device id needed to talk with it over the internet. They can then connect it up to Eduroam securely themselves, and the only downside is that we will need to manually send over code to flash onto the device. This can be somewhat mitigated by testing features on the test Photon, then sending that tested code to Appleton. These issues took up a large amount of our time in this demo cycle, since they were so important.

In the meantime, we have been hard at work gathering readings from the various sensors we have access to in Appleton, as well as controlling motors for the lift and the solenoid for the locked door, as is discussed in section 3.1. Some of this testing process can be seen below in our quantitative analysis. The forklift design mentioned in the last demo is currently being built, and an initial Lego version has been completed. Finally, something we missed

Figure 2. First build of the forklift

in the last demo was the topic of hardware accessibility. Through meetings, it was suggested that we could have the device mounted at different heights or have a sliding bar to adjust it for less able users, as well as other suggestions such as ensuring the doors are easy to open or adding legs if it sits on the ground. We fully intend to make the device as usable as possible, but there is a trade-off here between accessibility and security. For example, the sliding bar would harm security since it would be much easier to detach the mailbox from it compared to if the box were directly attached to the wall. We believe that since avoiding package theft is our main goal, the priority with our design should be that attempts to steal the entire device or break into it should not be easy. In reality, there would be a paid installation, and during this the customer would be able to request the ideal height for them. We will be paying attention to things like this in both our hardware and app design.

2.3. User interface

For this demo we completed our revision of the new mobile app UI and we gave just begun implementation. The priorities in this redesign were to simplify user navigation and to be more inclusive of those with visual and physical disabilities. We also took on board the feedback collected at the 'Usability Workshop' and conducted a full evaluation (see Appendix) based on a recommended checklist for usability[6], by Nielsen Norman Group, for UI design

usability.

To simplify navigation, we have moved the core functions of the app to the home screen (the device manager) of the app. Now, at a glance, the user can see important information regarding each of their devices.

For inclusivity, we have opted to go for a **very** light blue background so that coloured elements will be easier to identify (our primary colour being blue). We have also redesigned our icons so that they clearly convey their action (regardless of colour). The most crucial redesign was for important information/errors - a circled exclamation point will now be present beside the text.

To summarise our evaluation on usability, we believe that our application's UI will successfully accomplish the heuristics listed in the NNG article. Some parts to highlight where we feel we have accomplished great usability are...

- The fact the user can very easily see the status of their devices with minimal interaction. It is also clearly indicated when a status is healthy or is important.
- The vocabulary used in the app is simple and a novice user will know the purpose of each screen and action.
- Human error prevention is ensured at any point the user can modify their account or device with save and discard buttons.
- Help is very easy to find, with a static 'Help' button on every screen. The help screen provides links to common scenarios and errors.

See Figure 3 for wireframes of the home, device information and notifications screens plus 2 screens our current implementation. View our Figma demo to navigate a wireframe of our app.

We have also added animations in our app's implementation to improve the UX. We have added a loading animation while signing in and transitions between screens.

Our next step is to carry out user research, and we can compare our evaluation to the data gathered from users outside of development. Details of this testing will be discussed in the Evaluation section.

For the website, we have started filling in content for the 'System' and 'Team' section, as well as improving the mobile view. See our progress on our website here.

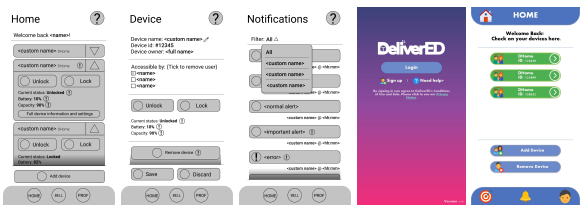

Figure 3. App UI Redesign Wireframes | Current Implementation

2.4. Software

2.4.1. Registration

QR code is chosen for registration. User can add devices to account by scanning QR code on the mailbox. This QR code contains unique identification information. Mailbox ID will be processed by device control and notification system.
From a security perspective, mailbox is expected to be only connected with authenticated accounts. The first account which successfully added device will be automatically registered as an "owner". Owner have right to prevent other accounts from pairing with the mailbox. This is especially useful when QR code is lost and found by strangers somehow. If the stranger tries to pair with the mailbox, it will send a message to the owner. The owner must authorize the stranger's account. Without authentication, a new account cannot pair with a mailbox which have had a owner.

2.4.2. Notification

Notification system experienced a reconstruction. We have made it more user-friendly by adding "delete", "swipe" and "clear". User can delete a notification by drag it to left, swipe messages by long press then drag it upward/downward and click "clear" button to clear all notifications.
A notification has life cycle now. It can be created from an HTTP message. A HTTP client was implemented for notification system. We decided to use Polling to connect application with mailbox. The App periodically sends http requests to the mailbox and waiting for response. A secret authentication header was used in HTTP the POST to protect privacy. The mailbox can also send messages to the app if the status of mailbox meet a certain threshold. Notifications are generated based on both Polling response and auto-generated message.

2.4.3. Settings

User can change settings now. Settings are separated to 5 parts: General, Account, Mailbox, App UI Preference and User Agreement.

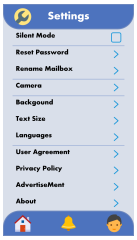

Figure 4. Settings

3. Evaluation

3.1. Hardware Testing

With respect to the the electromagnetic lock, we are able to operate it from our microcontroller, as has been discussed in the hardware section. We are also able to read sensor values to check if our door latch is pulled in (=1) or pushed out (=0), or if our solenoid off (=0) or on (=1) and if it is pulled in (=1) or pushed out (=0).

Notification	Success	Errors
locked	5	0
unlocked	5	0
receivedParcel	5	0
full	5	0
lowCharge	5	0
noCharge	5	0
locationChanged	5	0
connect	5	0
disconnect	5	0
paired	5	0
removed	5	0
readyUpdate	5	0
resetPassword	5	0
renameDevice	5	0

Table 1. Notification | Developer Testing

Items	♯	Cost(£)
Metal Work [Lock]	1	10.00
Standard Deadbolt Lock	1	20.00
Solenoid	1	25.00
Electronics And Wiring [Lock]	-	5.00
Motor [Fork Lift]	1	35.00
Lego Set [Fork Lift]	1	15.00
Motor Board [Fork Lift]	1	10.00
Basic Sensors	-	10.00
Advanced Sensors	-	30.00
Encoder Board [Fork Lift]	1	10.00
Miscellaneous [Fork Lift]	-	10.00
Particle Photon	1	18.67
Raspberry Pi	1	30.00
Technician time (3 hrs) [Fork Lift]	-	-
Total		228.67

Table 2. Budget

We tested the open/closed state of our solenoid by both operating it with our microcontroller and manipulating it manually, then checking the sensor readings for whether they reflect it being pulled in or pushed out. On average, the readings were accurate 4/5 times. We suspect that the sensor readings are not entirely accurate due to some noise present in them. We ran a similar test after manipulating our latch and the latch sensor returned expected values between 3/5 and 4/5 times, with some noise accounting for 1 or 2 unexpected readings. For each case, we changed states (i.e., between 0 and 1) thrice.

Garry was present at the lab to confirm that our solenoid was working as expected at all points, which means that it was pulled in at our "on" command and pushed out at our "off" command (as was implemented using the microcontroller).

We also tested a setup of starting a motor depending on the infrared sensor reading, where the motor starts if an object is at less than x cm from the sensor and stops otherwise. We expect to refine this and implement this concept while operating our forklift, where it lowers the platform only until the parcel is at the infrared sensor's level inside the secure compartment, and thus, is an obstacle in front of the sensor. We also wish to use this idea when opening the collapsible door once the parcel has entered the drop-off area, is an obstacle in front of the sensor in that area, and the access door to the drop-off area has been closed.

3.2. Software Testing

3.2.1. Developer Testing

Notification system has been tested. A success in testing represents a notification can be successfully generated by a given HTTP message and then display in alert page. Testing result is shown in Table 1.

3.2.2. Usability Testing

We will conduct video meetings with testing participants on Microsoft Teams. The testing will entail a user being given scenarios to achieve (such as adding a device); free range over the app's navigation; and a questionnaire enquiring usability and inclusivity. The user will interact with an online demo on Figma. The benefit to using this is that they will be able to share their screen so we can see their actions, and it does not require them owing an Android device.

4. Budget

Table 2 summarises our expenditures till date. While this is the budget for the hardware prototype of our system, the manufacturing cost of the actual mailbox will differ. Although this is over £200, this does not reflect the real manufacturing cost, which we intend on keeping within that limit. Among other things, we expect that purchasing material in batches would mean lower cost per item. We will summarise the estimated cost of building the real system and its intended market price in demo 4.

A few considerations for the real system would be:

1. **Handymen for installation:** We suggest 2 people for 2 hours for installing the smart mailbox according to the customers' preference. The average hourly rate of a handymen in Edinburgh is £11.71 [2]. So the total installation fee will be £46.84. (Note that this is excluding the bracket for installing the mailbox onto the wall.)
2. **UV Sterilizer LED:** We expect there to be 2 such LEDs (110 V, US) on 2 opposite walls (the ones adjacent to the access doors). The total comes to £24.02 (£12.01 each) [1].

5. Video

Watch our Demo video here (SharePoint).

Watch our full animated introduction video here (SharePoint) (a snippit will be played in the Demo video).

References

[1] "50W UV LED Ultraviolet Sterilizer Lamp Germicidal Disinfection Mite Removal - 110V - US". In: (2021). URL: https://uk.banggood.com/50W-UV-LED-Ultraviolet-Sterilizer-Lamp-Germicidal-Disinfection-Mite-Removal-p-1685241.html?utm_source=googleshopping&utm_medium=cpc_organic&gmcCountry=GB&utm_content=minha&utm_campaign=minha-gbg-en-pc&currency=GBP&cur_warehouse=CN&createTmp=1&utm_source=googleshopping&utm_medium=cpc_bgs&utm_content=sxxx&utm_campaign=sxxx-ssc-gbg-all-newcustom-ncv80-0928&ad_id=468370263002&gclid=CjwKCAiAgc-ABhA7EiwAjev-j24qHFNyqYHWovR-0uR6sUqkLh0BnBq5BSeRZAGhFRHUdECgRBSD2BoCOLMQAvD_BwE&ID=566615.

[2] "Average Handyman Hourly Pay in United Kingdom". In: (2021). URL: https://www.payscale.com/research/UK/Job=Handyman/Hourly_Rate.

[3] "INTRODUCING The Yale Smart Delivery Box". In: (2021). URL: https://august.com/pages/smart-delivery-box.

[4] "Locker". In: (2021). URL: https://www.amazon.co.uk/Click-and-Collect-with-Amazon-Lockers/b?ie=UTF8&node=2594544031.

[5] "myRENZbox for RESIDENTIAL LIVING". In: (2021). URL: https://safetyletterbox.com/parcel-boxes/myrenzbox-for-residential-living/.

[6] Jakob Nielsen. "10 Usability Heuristics for User Interface Design". In: (2020). URL: https://www.nngroup.com/articles/ten-usability-heuristics/.

Appendix

UI Evaluation

1. **Visibility of system status:** On the home screen, the user can see at a glance if their devices are in a healthy status or not by the circles exclamation point indicator next to them. The user can click the dropdown button on a device to see the 3 most important statuses: locked/unlocked, battery life and capacity.

2. **Match between system and the real world:** The vocabulary used in the app is simple and should not take a tech-savvy user to understand what information is being presented or what action a button will take.

3. **User control and freedom:** At any opportunity the user has to edit anything, there are confirmation and cancel buttons either on the same screen or in a pop up modal. Important actions will have an exclamation point beside them.

4. **Consistency and standards:** There is no ambiguity in the language used for actions throughout the app. Actions can only be performed from their designated page, e.g. editing your device name can only be done by navigating to that device's information.

5. **Error prevention:** Errors from the physical device will come through on the notifications screen. They are easy to identify by the big exclamation point icon and universal exclamation point beside the text. In addition, they will be highlighted red. All notifications will come through the phone's native notifications as well. Errors within the app will appear as a toast notification at the bottom of the screen. All errors will have meaningful descriptions.

6. **Recognition rather than recall:** With our simple design that has few screens to navigate the user should not have to recall where to find information or actions.

7. **Flexibility and efficiency of use:** We have all our core information and functionality on the home screen so that a novice user can easily perform tasks. On the home screen, each device has a button that takes the user to a new screen where they can perform more actions. Here they can edit the device name, if they are the owner, control access to the device or remove the device.

8. **Aesthetic and minimalist design:** Our interface is simple, with large text and buttons. There is no clutter and we make the most of the screen's real estate.

9. **Help users recognize, diagnose, and recover from errors:** As mentioned earlier, error messages are identifiable from the large exclamation point and red highlighting, with a meaningful description that will also suggest looking at the help screen.

10. **Help and documentation:** As mentioned above, there is a help screen with quick links to how to complete common tasks and solve common errors. It also has buttons to see the full user guide online or download it as a PDF, as well as an 'Email us!' button.

Market Competitors

Here we look at some of the existing market solutions for safeguarding delivered packages when the customer is unable to collect it firsthand. We discuss below how our product stands out from these identified competitors.

1. **myRenzbox:** [5] It is targeted at commercial and residential building owners or office owners and is not meant for personal use. Even for apartment dwellers, building owners need to provide it for its tenants, since it is not designed to be purchased by an individual. DeliverEd Home, on the other hand, has been designed keeping home owners in mind, which means that it is perfectly suitable for individual purchase and use. This also means that our design is much more compact as compared to myRenzBox and can be installed into smaller spaces. The sanitation feature of our product is another differentiating factor, which is not included in myRenzbox.

2. **Yale Smart Delivery Box:** [3] A key difference between our products is that there is no secure compartment in their design. The parcel drop-off and collection area are the same. This makes already dropped off parcels susceptible to theft by delivery people with ill-intentions. This was a key consideration in our design for DeliverEd Home, where we have a secure compartment for parcel storage, which is not accessible from the drop-off compartment. The Yale box is unlocked using a keypad attached to it, which is an optional feature. In contrast, DeliverEd Home can be unlocked remotely, using our app, while having a backup lock and key mechanism. This means that you can give your trusted ones access to you parcels in the secure compartment without having to be present or share a pass-code. This is useful in situations such as where you may have perishables delivered that you can't collect in time. The Yale box remains unlocked until a delivery is made and then auto-locks. On the other hand, the drop off area for DeliverEd Home is secured using an electromagnetic lock which we plan on operating using our app. We are considering being able to keep it open for the duration that you expect a delivery. Securing it was considered so that no random person can pull a prank and drop off unexpected items into you mailbox. Yale's design also does not consider accessibility issues. For example, in most cases, one would have to bend down to retrieve parcels. DeliverEd Homes takes this into consideration and allows you to customise its installation. The sanitation option is, again, a differentiating factor between our product and Yale's, where they don't provide one, unlike us.

3. **Amazon Lockers:** [4] This is a "go and collect" service which may or may not be available in an area close to you, whereas with DeliverEd Home, you are able to store your parcels at your own address. So you don't have to worry about the inconvenience of having to travel to collect your parcels. With Amazon Lockers, you have 3 calendar days to pick up your parcels, but with DeliverEd Home, you don't have to worry about how long you are away. You can store your parcels in our mailbox for as long as you need to. Another important consideration is that only items dispatched by Amazon are eligible for delivery to and collection from a Locker. With DeliverEd Home you can opt for any courier service of your choice. And lastly, just as with the above two, you have an option to sanitise your parcels inside our mailbox, which you don't with the Amazon Locker.