July 12th, 2020

Dr. Craig Scratchley
School of Engineering Science
Simon Fraser University
V5A 1S6

RE: ENSC 405/440 Design Specifications for **DPL** by **Parcel Solutions Inc.**

Dear Dr. Scratchley,

The attached document, Parcel Solutions' design specifications for DPL, provides a summary of our capstone project. Our goal is to design a cost effective and secure delivery parcel locker to improve the online shopping experience.

The purpose of this proposal is to outline the design of the DPL. This includes system overview, overall device design, engineering standards, as well as User Interface and supporting test plans. The specifications outlined in this document will serve as a guide throughout the design of the DPL.

Parcel Solutions consists of 6 driven and talented senior engineering students ranging in concentrations from Computer Engineering and Electronic Engineering: Arian Vafadar Moghaddam, Boey Leung, Chaoqun Ding, Huron Lee, Linqi Cheng, and Sina Ahmadian Behrouz .

We appreciate your time in reviewing our functional specifications for DPL. If you have any questions or concerns regarding our proposal, please do not hesitate to contact our Chief Communication Officer, Sina Ahmadian Behrouz, by phone at (604) 771-1996 or by email at ahmadian@sfu.ca.

Sincerely,

Arian Vafadar Moghaddam
Chief Executive Officer
Parcel Solutions Inc.

Design Specification for *DPL*

Project Team - Group 4:
Arian Vafadar Moghaddam
Boey Leung
Candice Ding
Huron Lee
Lynch Cheng
Sina Ahmadian Behrouz

Contact Person:
Sina Ahmadian Behrouz
ahmadian@sfu.ca

Submitted to:
Dr. Andrew Rawicz – ENSC 440
Craig Scratchley – ENSC 405W
School of Engineering Science
Simon Fraser University

Issue Date:
July 12th, 2020

Table of Content

List of Figures

List of Tables

Glossary

- **2PL**: Two-Phase locking concurrency control which guarantees serializability.
- **Bcrypt**: bcrypt is a password-hashing function based on the Blowfish cipher
- **CSRF**: Cross-site request forgery
- **FW**: Firmware
- **MVCC**: Multi-Version Concurrency Control
- **ORM**: Object Relational Model
- **REST API**: Representational State Transfer protocols for API design
- **SHA-256**: is a set of cryptographic hash functions.

Abstract

This document describes the design specifications and outlines the details of DPL. It contains a detailed look at each specific engineering standard and a complete analysis of the problems it targets to solve. The goal is to give the reader an explanation of the design choices and justifications. Preliminary future design goals are also included.

The DPL consists of 2 components at its highest level: a website and a parcel locker aimed at improving security and efficiency of online shopping. The website will display UI and the frontend; however, the website will be controlled by the underlying server which will have its own components. Other than filling backend needs, the server will control the DBMS, DPL management, and a connection to the courier server for information retrieval. The parcel locker will consist of its own intuitive software controls along with an easy to use UI. The website and the parcel locker will seamlessly integrate with the firmware and hardware, resulting in a product that creates the most efficient and secure online shopping experience.

This document focuses on the technical details of hardware, firmware/software, physical/ operational and mechanical components along with the justification of the chosen parts and design decisions that were made. The engineering standards that are being followed are also included in this document as well as a high-level test plan. Specifically, a high-level test plan is presented in order to comprehensively test all the different components of the DPL system and to ensure full functionality. A fully functional proof-of-concept system is to be delivered by August 20st, 2020.

1. Introduction

This document intends to outline the technical design specifications for Delivery Parcel Locker DPL. These specifications will describe the various design choices that were made in accordance to the function requirements for the product, as outlined in the DPL Functional Requirements Document [1].

DPL is broken into two separate communicating components: the DPL station and the web application. The DPL station consists of the parcel lockers, 7-inch display, main console unit and Raspberry Pi4 camera. The web application will consist of a backend server and a DBMS. Each major component and sub-component will provide detailed justifications for these design decisions, and the specific design processes on how the corresponding systems shall be implemented and interact with one another. Furthermore, the design specifications of DPL provides details pertaining to the specific computations, physical, mechanical and electrical, safety, and environmental requirements necessary to develop the proof-of-concept and the prototype of the DPL.

The design specification document will outline the following:

- The DPL product and its subsystems
- An overview of the proof of concept model and its product features
- Technical details of system design to support the functional requirements
- Test plans to examine proper functionality
- User Interface Design

1.1 Background

A common issue many online shoppers have is efficiency and accessibility to one's package. At the moment, couriers around North America are instructed to leave a note on apartment buildings if the customer is not available to receive their package. The customer is then expected to take the note left at the door by the courier and go pick up their package from the nearest postal service office. This solution is extremely inefficient and inconvenient for many customers. With DPL, online shoppers will no longer need to go to a postal service office to pick up their package. Parcel Solutions will provide online shoppers with a secure and remote method for customers to have their packages delivered.

A major issue that has surfaced across North America in the past several years with the fast rise of online shopping is package theft. Online shoppers around North America have packages delivered to their homes and apartments all the time, however, occasionally they come home

from work to pick up their package from their doorsteps just to realize that their package has been stolen. A major issue we Parcel Solutions is targeting is security. DPL by Parcel Solutions will improve package security all around North America and ensure that customers will never have to worry about package theft again.

Currently, there is no product like this on the market, only Amazon offers something with a similar functionality. However, the lockers that Amazon offers aren't widely accessible across North America and are only available to online shoppers that directly order through Amazon. This is where we differentiate ourselves from the competition.

1.2 Scope

This report clarifies the functional specifications and requirements of DPL, which must be met by Parcel Solution Inc. This document will give a thorough description of the proof of concept model and all the different requirements necessary for production. Additionally, it seeks to clarify the high-level functionality of the product, how it would work and how it would be utilized.

1.3 Intended Audience

This document is intended to be used by the members of Parcel Solutions Inc. for development of DPL as well as the teacher assistants' marking this document, and senior engineers who are keen on this project. It can be used throughout the research and product development stages as a reference to provide the overall view of the product. Furthermore, future revisions of DPL will draw from the framework outlined and detailed in this document.

1.4 Design Specification Classification

In order to make it easier to reference and prioritize design specifications, the following convention will be used throughout this document:

[Des A.B.C. D – XX] A Design Specification

Des is an abbreviation for design. The letters **A**, **B**, **C** and **D** correspond to integers values that symbolize the hierarchical order of the design specifications. The letter **A** will reference the root section, **B** represents the subsection of **A**, **C** corresponds to the subsection of **B** and finally **D** corresponds to the subsection of **C.** The **XX** specifies the priority of the requirement.
There are three priority levels:

PC – The requirement refers to a proof-of-concept (Alpha phase)
PT – The requirement refers to a prototype (Beta Phase)
FP – The requirement refers to a final product (Production)

2. System Overview

This section will identify the system overview of our product. DPL incorporates two subsystems that communicate with one another to form the overall system. The first system is the DPL station which consists of the parcel lockers. The second system is the web application and the server module. The general requirements of DPL are shown in Table 1. The following sections will cover the justification for the use of each component, a technical description and how they integrate into the DPL system.

Des 2.1 - FP	The system will be intended for outdoor usage
Des 2.2 - FP	DPL owner and customers should be able to cancel any expected deliveries
Des 2.3 - FP	The embedded system along with its needed hardware must cost less than $500 (not including the cost for the skeleton body of the mailboxes)
Des 2.4 - FP	The device shall be easy to assemble and disassemble
Des 2.5 - FP	Strong documentation practices and software version control management systems will be implemented
Des 2.6 - FP	Physical user manuals should be printed and shipped with the product. Languages used in the manual will be appropriate for their destination
Des 2.7 - FP	The device can be operated and initialized by a single person without a need for prior technical knowledge

Table 1: General Design Requirements

2.1 Product Design

DPL is a modern parcel locker that consists of a backend/frontend website and a locker to improve the online shopping experience. The website will display UI and the frontend; however, the website will be controlled by the underlying server which will have its own components. Other than filling backend needs, the server will control the DBMS, DPL management, and a

connection to the courier server for information retrieval. The interactions between the server, DBMS, DPL management and website can be seen in Figure 1.

The parcel locker will consist of its own intuitive software controls along with an easy to use UI. The UI will be presented to the customer and courier in the form of a digital screen. The courier and customer interaction with the parcel locker UI is shown in Figure 1.

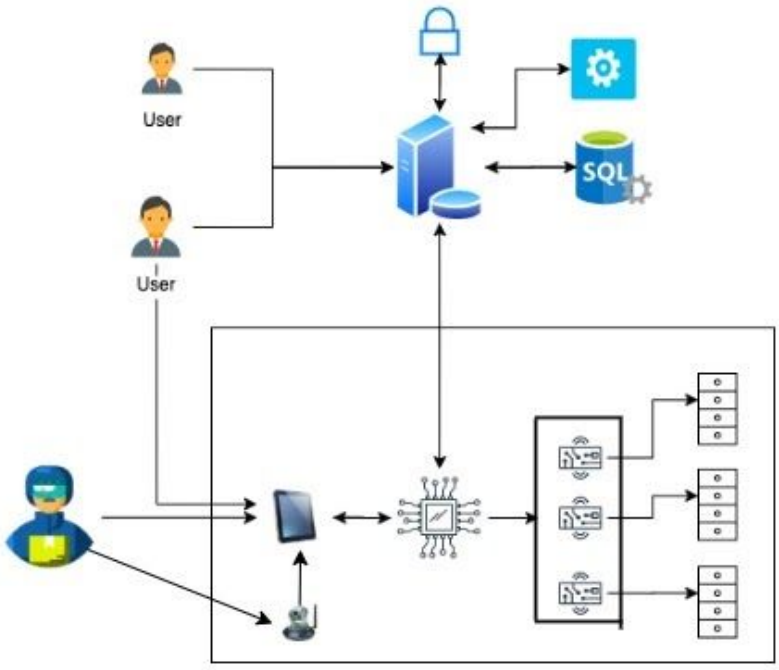

Figure 1: Overview interaction system diagram

2.1.1 Website and Server Module

The website component of DPL will consist of a backend server and a DBMS that will coordinate the customer and DPL interactions. The customer will be required to create an account with DPL on our website in order to be able to use the DPL services. The customer will be required to log in onto our website and make a DPL delivery request. DPL will generate a set of instructions for the courier company to access the DPL lockers. Once the customer receives their tracking number, they will be required to input the tracking number into the DPL website. This allows DPL to check the database and match the delivery date of the package with an empty locker. Once the package tracking number has been put into the database, DPL will generate an access code for the customer to access the parcel locker. The tracking number and customer information will be stored in the database for future reference.

2.1.2 Parcel Locker Module

The parcel locker will consist of its own intuitive software controls and a digital touch screen to present the UI. The courier and the customer will interact with the DPL lockers through the digital touch screen. The courier company will have specific instructions given to them by DPL to access the parcel lockers and place the customer package inside. The instructions provided by DPL will include two options for the courier to access the parcel lockers. The courier can either scan the barcode of the package, or alternatively, the courier can enter the provided passcode by DPL. The customer will use an access code provided by DPL to pick up their package from the parcel locker. Once the customer has picked up their package, DPL will update their database and the locker availability.

3. Design Specification

This section will identify the different hardware, software, and physical components that will be used in our product. The microcontroller and sensors used in the system are manufactured by third party libraries and will be configured and integrated to enable DPL backend functionalities. The following sections will cover the justification for the use of each component, a technical description and how they integrate into the DPL system.

3.1 Physical

The physical design of DPL consists of a DPL station design and the parcel lockers design. The physical requirements of DPL are shown in Table 2. The design choices for these requirements will be explained in section 3.1.1 and 3.1.2.

Des 3.1.1 - FP	DPL's exterior body shall be made out of aluminum
Des 3.1.2 - FP	DPL's exterior is water-proof
Des 3.1.3 - FP	The controller's box must allow for air to transfer out, this will be used along with fans for cooling mechanism
Des 3.1.4 - FP	The height of DPL must not exceed 160cm
Des 3.1.5 - FP	All the lockers and main console box must be attached together
Des 3.1.6 - FP	Each locker must be with the following dimensions, (L:W:H), 80x70x18 cm

Des 3.1.7 - FP	All inputs, outputs, and controls must be clearly labeled
Des 3.1.8 - FP	The Display monitor must be within 60-80 cm from the ground, which is in compliance with the average adult waist measurement of 70 cm in North-America.
Des 3.1.9 - FP	DPL will require a minimum of 180 cm of flat base and 80 cm of clearance space.
Des 3.1.10 - FP	DPL must allow for circulation of fresh air in the main controller locker.

Table 2: Physical Design Requirements

3.1.1 DPL Station Design

The DPL station design will resemble the diagram shown below in Figure 2. The DPL station will be composed of a display screen similar to the one shown in Figure 2. The station will also consist of several parcel lockers that can be accessed by both the customer and delivery parcel company. The delivery parcel courier and the customer will interact with the DPL station using the display interface and barcode scanner. Furthermore, each parcel locker will be assigned a

number such that the customer and courier company can easily distinguish between the lockers.

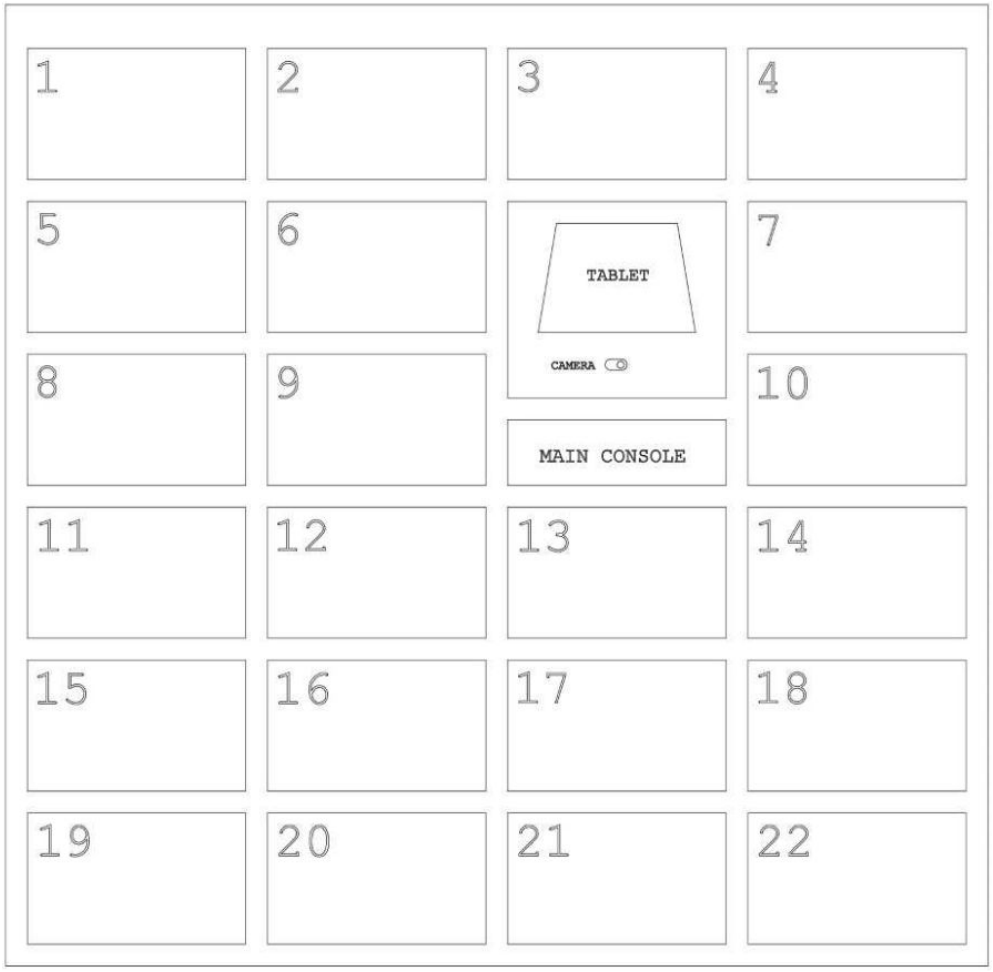

Figure 2: Proposed DPL station for final product

The DPL station will be designed and assembled as such so that the maximum height of the station does not exceed 160cm. The station will also require a minimum of a 180cm flat base and 80cm of clearance space to ensure that the station can be deployed. All the lockers and the main console box will be attached together to form the overall DPL station. The main console that will contain the Raspberry Pi's and other wiring will be protected and hidden using aluminum sheets. A small section of the console box will be cut in the shape of a rectangle to display the Raspberry Pi screen. The Raspberry Pi Screen and display specifications are shown below in Figure 3.

Screen specifications:

- 7" Touch Display
- Dimensions: 194 x 110 x 20 mm

- Actual screen size: 155 x 86 mm
- Resolution: 800 x 480 px
- 10 finger capacitive touch

Figure 3: Official 7" Touchscreen for Raspberry Pi

3.1.2 Locker Design

The design requirements for the parcel locker are shown in Table 3. This section will provide the justification of requirements and the design choices for the parcel lockers.

Des 3.4.5.1 - FP	The lock shall be made from aluminum metal sheets
Des 3.4.5.2 - FP	The lock shall have surface mounting either vertically and horizontally offers easy and flexible installation
Des 3.4.5.3 - PC	The lock shall have a minimum lifespan of 1 million clicks
Des 3.4.5.4 - FP	The lock shall have a dimension no larger than (LxWxH) 80x70x18mm.
Des 3.4.5.5 - FP	The lock shall be applicable for 0.5-4 Kg weight doors
Des 3.4.5.6 - FP	The work environment of the lock shall be between -30 to 50 Celsius
Des 3.4.5.7 - PT	The cabinet lock shall not have a handle, upon opening the lock will be unlocked and the door will be pushed open

Table 3: Parcel Locker Design Requirements

The DPL lockers will be designed similar to the diagram shown below in Figure 4. The lockers will all be attached to each other as shown in Figure 2. The lockers will all be the same size, specifically, the lockers shall will have the following dimensions:

Length = 80cm Width = 70cm Height = 18cm

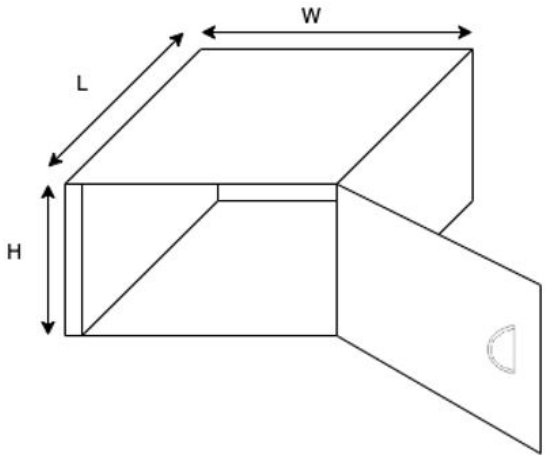

Figure 4: Locker Cabinet Diagram

The lockers shall not have a handle, instead the lockers will be opened using the display interface. The customer and the courier will have to close the door by pushing it in.

The lockers will be made using aluminum sheets. The aluminum sheets will be cut and shaped using a Water Jet machine. Parcel Solutions Inc will be working with ABCNC Industries Ltd in order to create and design the lockers. The aluminum material will ensure that the lockers will not be damaged by water and ensure that the system is waterproof. This will protect the electronic components and wirings inside.

3.2 Hardware

The Hardware segment of DPL is the bridge that allows couriers and users to physically interact with DPL. DPL will supply an empty box for a courier to store the parcel. The courier will verify themselves and the master controller will receive the signal from the touch interface then give the commands to the slave controller to open the door automatically. After the courier stores the parcel and closes the door, the locked Electronic Cabinet Lock sends the message to the master controller that this box is employed. Upon delivery, the main controller will be responsible for communicating with the web servers to notify customers to alert them about the status of their

package.

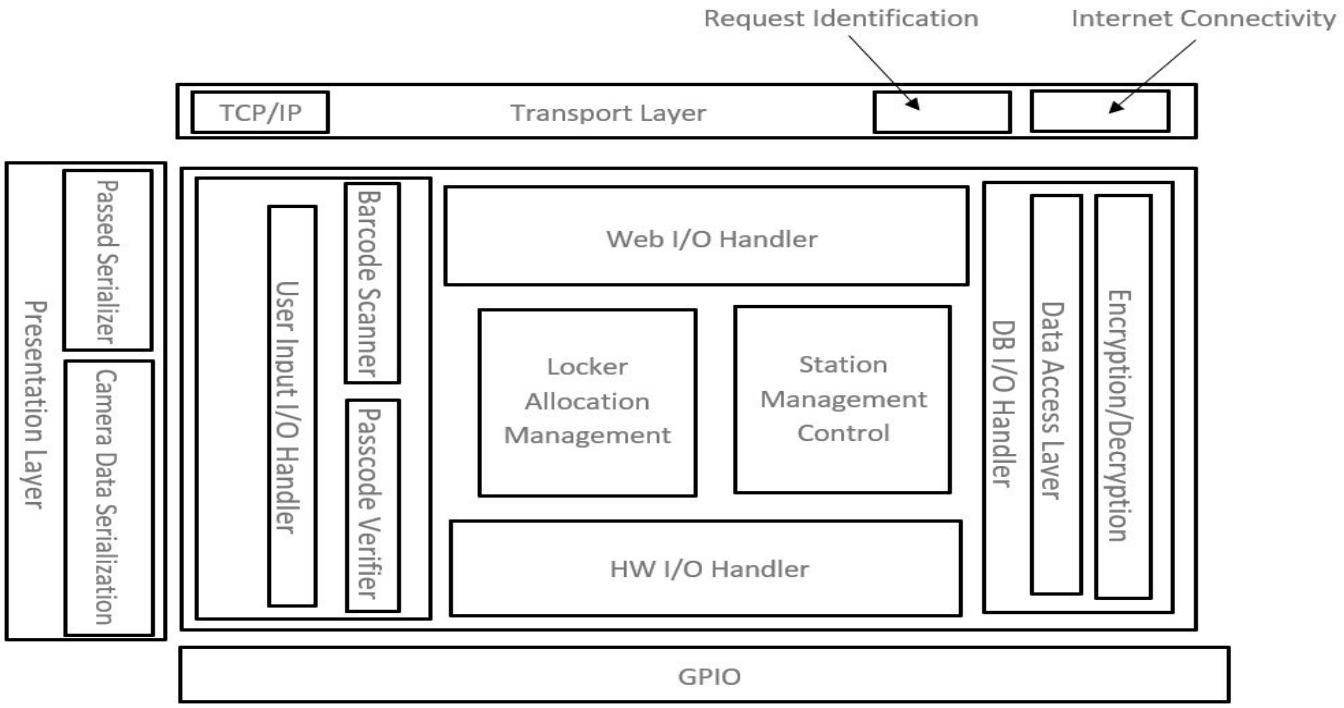

Figure 5: Modular System Diagram of FW within DPL station

The modular diagram in figure 5 will display the different components of the Raspberry Pi that enable the functionality of the station.

1. Presentation Layer
    a. The presentation layer is where GUI resides, this layer will interact as an interface between the user and the system. It addiotionally processes users inputs.
2. User input I/O Handler:
    a. Will process the inputs provided by the user using core functionalities.
3. HW I/O handler:
    a. Responsible for interacting with the slave controller and reading the state of locks.
4. Web I/O Handler
    a. Will interact with the web server to receiver/send information regarding state of deliveries.
5. DB I/O Handler:
    a. Will control access to the database. Additionally, it encrypts/decrypts sensitive data upon access.

3.2.1. Master Controller - Raspberry Pi

Raspberry Pi Model will be the microcontroller for the output module of DPL in the proof of concept. Raspberry Pi's ability to execute multiple concurrent programs, along with powerful

processing ability to handle image processing while performing simultaneous RTOS tasks, makes it an ideal microcontroller for DPL station. We can separate the input and output into different modules as we want the process to be parallel. Another reason would be the compilation with other components. Raspberry Pi can be easily integrated with the slave controller and LCD screen without any additional drivers or extensions.

Figure 6: Raspberry Pi Model [2]

Des 3.2.1.1 - PC	The secondary storage is 8 GB.
Des 3.2.1.2 - PT	The main memory is 2 GB
Des 3.2.1.3 - PC	The main controller supports Gigabit Ethernet connection
Des 3.2.1.4 - FP	The main controller supports bluetooth module
Des 3.2.1.5 - PC	The master controller has general-purpose input/output (GPIO) and Display to the serial (DSI) ports to connect the display interface
Des 3.2.1.6 - PC	The master controller has USB ports to connect the slave controller
Des 3.2.1.7	The main controller operates within 3.3 to 5 DC-Volts

Table 4: Raspberry Pi Model with satisfied requirements

3.2.1.1 Slave Controller - Electronic-Salon DPDT Signal Relay Module

We will use Signal Relay Module as our slave controller in the early stage of development as the turn-around time for our final slave controller is longer than expected. We are using the relay

system to act as a switch for each locker. Signal relay module is responsible for providing relay output to the locks. This system converts digital signals to electrical currents from different voltage levels to ensure the equipment receives the right amount of power at the right time. When the relay is off, the COMMON port is connected to the Normally Closed port. The Normally Open port connection of the relay is not connected until the relay turns on. When the relay turns on, the COMMON port moves from Normally Closed port to Normally Open port. Our choice of slave controller is Electronic-Salon DPDT Signal Relay Module. The main reasons for this choice are

- The small size of the board. It is compact to install in the locker.
- Compatible with the master controller.
- Low operating voltage usage.

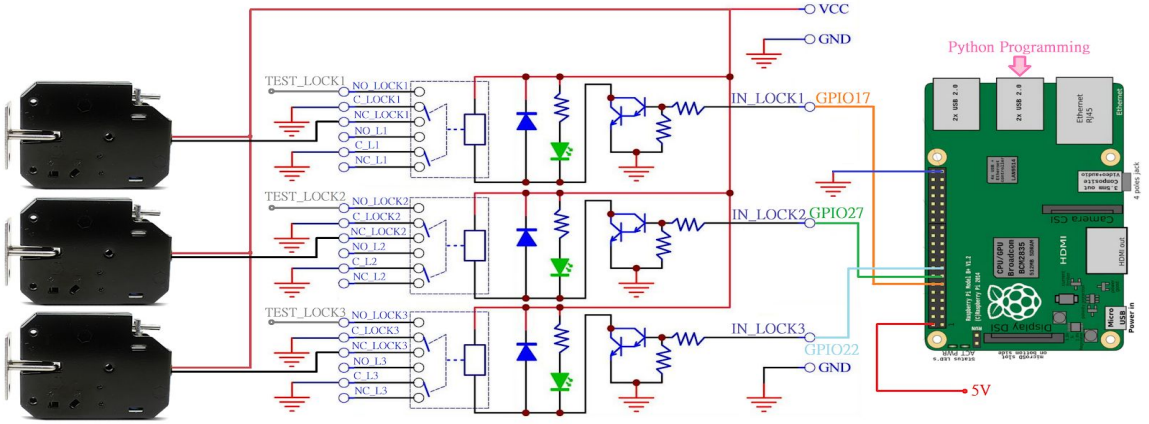

Figure 7: Overview of raspberry, lock and relay system integration.

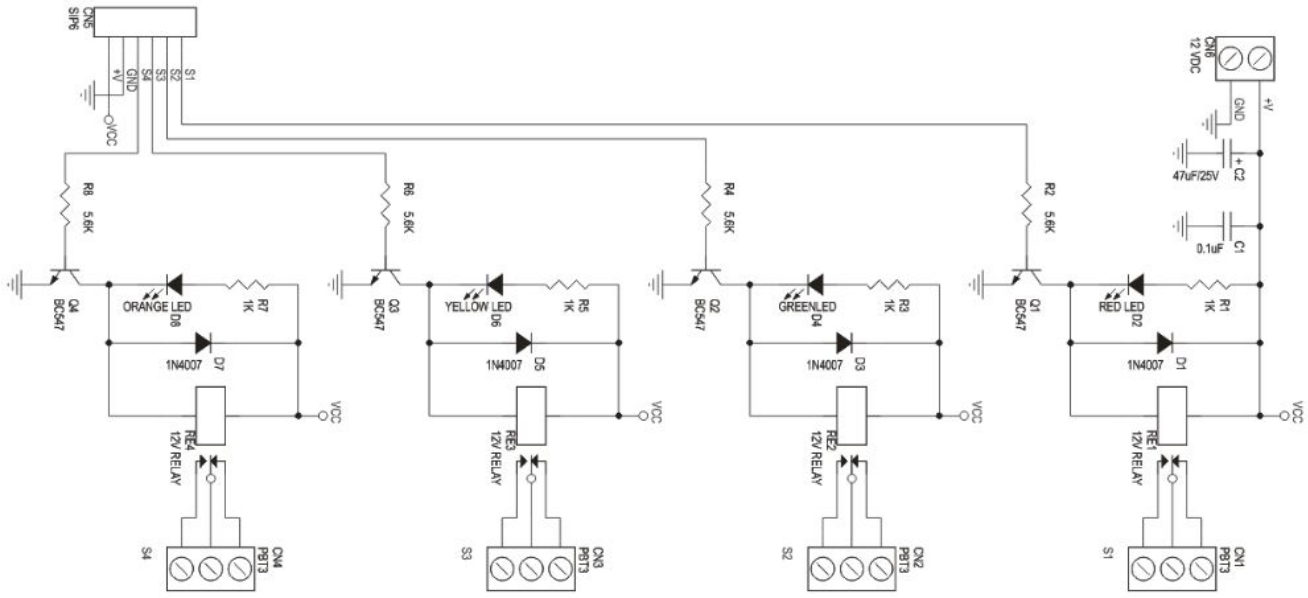

Figure 8: Circuit Schematic of Electronic-Salon DPDT Signal Relay Module [3]

Des 3.2.1.1.1 - PC	The signal relay module has GPIO header
Des 3.2.1.1.2 - PC	The product dimension is L9.9*W9.9*H5.3mm
Des 3.2.1.1.3 - PC	The product weight is 40.8g
Des 3.2.1.1.4 - PC	The operating current is within 0.5 to 1 mA
Des 3.2.1.1.5 - FP	The signal relay module supports 5 electronic cabinet locks

Table 5: Electronic-Salon DPDT Signal Relay Module with satisfied requirements

3.2.1.2 Slave Controller - AL 2448

We will use AL 2448 over the Relay controller once we receive it and integrate the system together. DPL will use a slave controller as the interface between the main controller and the electronic locks. It is responsible to read the status of each lock and report the door status to Raspberry Pi. We chose AL 2448 for the slave controller as AL 2448 can report up to 24 locks' status and the system is able to report 984 locks with extension board by RS485 which provides a great opportunity to add more locks for our final prototype. The board additionally coheres with our engineering standards and requirements.

Figure 9: AL 2448 Slave Controller[4]

Des 3.2.1.2.1 - PC	AL 2448 can report 24 locks' status to Raspberry Pi
Des 3.2.1.2.2 - PC	AL 2448 has digital input for door status reporting
Des 3.2.1.2.3 - PC	AL 2448 has short circuit protection function

Des 3.2.1.2.4 - PC	The product dimension is L88*W214*H17mm
Des 3.2.1.2.5 - PC	AL 2448 can report up to 24 locks' status
Des 3.2.1.2.6 - FP	AL 2448 can report 984 locks' status with extension board by RS485

Table 6: AL 2448 with satisfied requirements

3.2.2. Touch Display - Raspberry Pi LCD Display

We will be using a Raspberry Pi LCD display. The touch display satisfies the dimension and resolution metrics in place for our LCD display. It is a 7", diagonal, multitouch 800 x 480 LCD screen shown in the Figure 10. It provides accurate readings of touch interactions and is compatible with our main controller without the need for calibration or any external drivers.

Figure 10: 7" Multi Touch LCD screen[5]

Des 3.2.2.1 - PC	The screen is a touch interface
Des 3.2.2.2 - PC	The power supply of display interface provides 500 mA
Des 3.2.2.3 - PC	The display screen is 7 inches diagonally
Des 3.2.2.4 - PC	The display interface has brightness of 900 Cd/m^2 to be visible even on bright days
Des 3.2.2.5 - PC	The display interface has a pixel resolution 800x480
Des 3.2.2.6 - PC	The display interface has 40 pins interface capable to buffer and refresh the display interface panel at 60 Hz
Des 3.2.2.7 - PC	The display interface has general-purpose input/output (GPIO) and Display to the serial (DSI) ports to connect to the master controller

Table 7: Raspberry Pi LCD display with satisfied requirements

3.2.3. DSCK6656 Electric Magnetic Lock

The electronic cabinet lock is the high security access control to the lockers. Our team chose DSCK 6656 Electric Magnetic Lock for DPL. The reasons are as follows:

- The chrome-plated steel housing provides a high reliability and protection to the lockers
- This lock can accept access control signals from the signal relay module. As such Raspberry Pi is available to indicate whether the mechanism is locked or unlocked
- The size of the lock is easy to install for cabinet locking application
- The low operation voltage of the lock is suitable for DPL

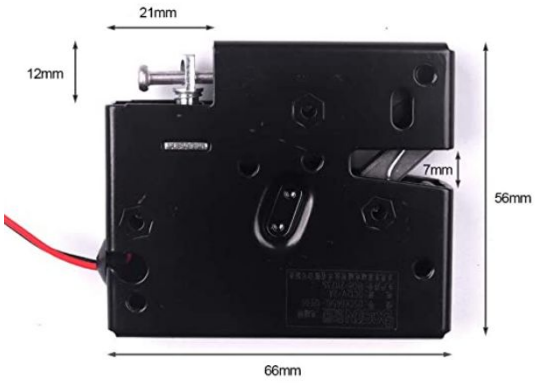

Figure 11: DSCK6656 Electric Magnetic Lock [5]

Des 3.2.3.1 - FP	The lock is made by steel plate material
Des 3.2.3.2 - FP	The lock has surface mounting vertically and horizontally offers easy and flexible installation
Des 3.2.3.3 - PC	The lock has a minimum lifespan of 1 million clicks
Des 3.2.3.4 - FP	The product dimension is L80xW70xH18 mm
Des 3.2.3.5 - FP	The lock is applicable for 0.5-4 Kg weight doors
Des 3.2.3.6 - FP	The work environment of the lock is between -30 to 50 Celsius

Table 8: DSCK6656 Electric Magnetic Lock with satisfied requirements

3.2.4 Camera - Raspberry Pi Camera Module V2

The camera is responsible for scanning the Barcode from the courier personnel and reading QR code from the user's mobile device. Our choice of camera was Raspberry Pi camera module v2

because it has an 8-megapixel sensor which can be used to take high-definition stills photographs. It also contains a CSI port which can directly connect to the Raspberry Pi without extra hardware components. The size of the camera has reached our requirements which is compact and easily be installed in the control panel. It is also competitive with its low cost.

Figure 12: Raspberry Pi Camera Module V2[6]

Des 3.2.4.1 - PT	The camera is able to scan the QR code
Des 3.2.4.2 - PT	The camera has a Camera Serial Interface (CSI) port to connect the master controller
Des 3.2.4.3 - PT	Product size is L40xW40x20 mm
Des 3.2.4.4 - PT	The camera is capable of 3280 x 2464 pixel static images
Des 3.2.4.5 - FP	The weight of the camera is 3g
Des 3.2.4.6 - PT	The camera supports 640x480 video
Des 3.2.4.7 - PT	The camera has a native resolution of 8 megapixel

Table 9: Raspberry Pi Camera Module V2 with satisfied requirements

3.3 Web Application

DPL software will mainly comprise of the web-application and the backend servers that will be the spine of dealing with the sum of interactions. DPL will gather sensitive data containing client information and data. It is fundamental to forestall information spillage and keep hidden information secure. Also, the modules must be intended to represent future redesigns and extensions. Our web-application will be based upon a layer of framework to deal with connection functionalities in the most efficient and secure way and allow us to focus on the logic of the code.

The website component of DPL will consist of a backend server and a DBMS that will coordinate the customer and DPL interactions. The customer will be required to log in onto our website and make a DPL delivery request. DPL will generate a set of instructions for the courier company to access the DPL lockers. Once the customer receives their tracking number, they will be required to input the tracking number into the DPL website. This allows DPL to check the database and match the delivery date of the package with an empty locker. Once the package tracking number has been put into the database, the web application will generate a passcode for the customer to access the parcel locker. The tracking number and customer information will be stored in the database for future reference.

3.3.1 Frameworks

Flask will be the backend framework for our web application. It is a lightweight micro-framework that will give the developers the flexibility to design the application as they desire. Flask is the best choice for our application, as we will need to define the structure of the interaction as opposed to relying on a default structure defined by the framework.

DPL will deploy it's frontend framework with the help of Vue.js framework. Vue.js is an open-source model–view–viewmodel JavaScript framework for building user interfaces and single-page applications. Vue Js is lightweight, small in size, flexible and well documented. Further on, it has an easier learning curve than alternative options such as React and Angular Js. Therefore, Vue.js is the best option for our application.

Des 3.1.1 - FP	Backend Framework is based on the MVC model.
Des 3.1.2 - FP	Backend Framework must be flexible and allow the engineer to design the interaction between different protocols.
Des 3.1.3 - FP	Backend Framework must be well documented.
Des 3.1.4 - FP	Frontend Framework is based on the MVC model.

Table 10: Framework Requirements

3.3.2 Database Design

The entirety of our data will be put away utilizing a DBMS server. DPL is an information driven venture; in this way, there can not be any irregularities with the database. It is crucial for the DBMS to be able to handle as many as concurrent real-time queries. It is additionally significant for the information to be steady and not be changed in an unfortunate manner.

Des 3.3.2.1- PC	DBMS will follow ACID properties.
Des 3.3.2.2 - PC	Database tables are normalized to prevent interdependencies.
Des 3.3.2.3 - PC	DBMS is relational as the data is carefully structured.
Des 3.3.2.4 - PC	DBMS will use MVCC concurrency control protocol.
Des 3.3.2.5 - FP	Postgres will be the DBMS server used for the web application.

Table 11: Database Requirements

3.3.2.1 Database Server

To enable concurrency and stability of the database, we will be using MVCC, Multi-version concurrency control, protocol as opposed to using a 2PL. MVCC allows for both reader and writer threads to work without blocking each other. Issuing a join operation between tables is a common operation within our backend logic. Locking a database to perform write operations will delay all the read operations. This will add a significant overhead to our queries. Given the magnitude of our application, MVCC is the optimal choice of concurrency control.

We will be using Postgres DBMS servers to handle the connections and queries of our relational DB. After looking into our APIs and view-controller design, we've noticed that about 55% of our queries will require read-only transactions. Postgres will handle concurrency and read-only operations, using MVCC, more efficiently than its competitors.

3.3.2.2 Database Schemas

There will be three database tables to perform delivery requests as displayed in Figure 13. It is also good to mention that there will be more tables used for book-keeping operations behind the scene, such as keeping track of cookies deployed to users, requests and logs. However, for core operations we will be focusing on the following three tables.

1. Users: will contain basic information regarding users
2. DPLstations: contains information each dpl station available
3. Deliveries: contains information regarding a package delivery

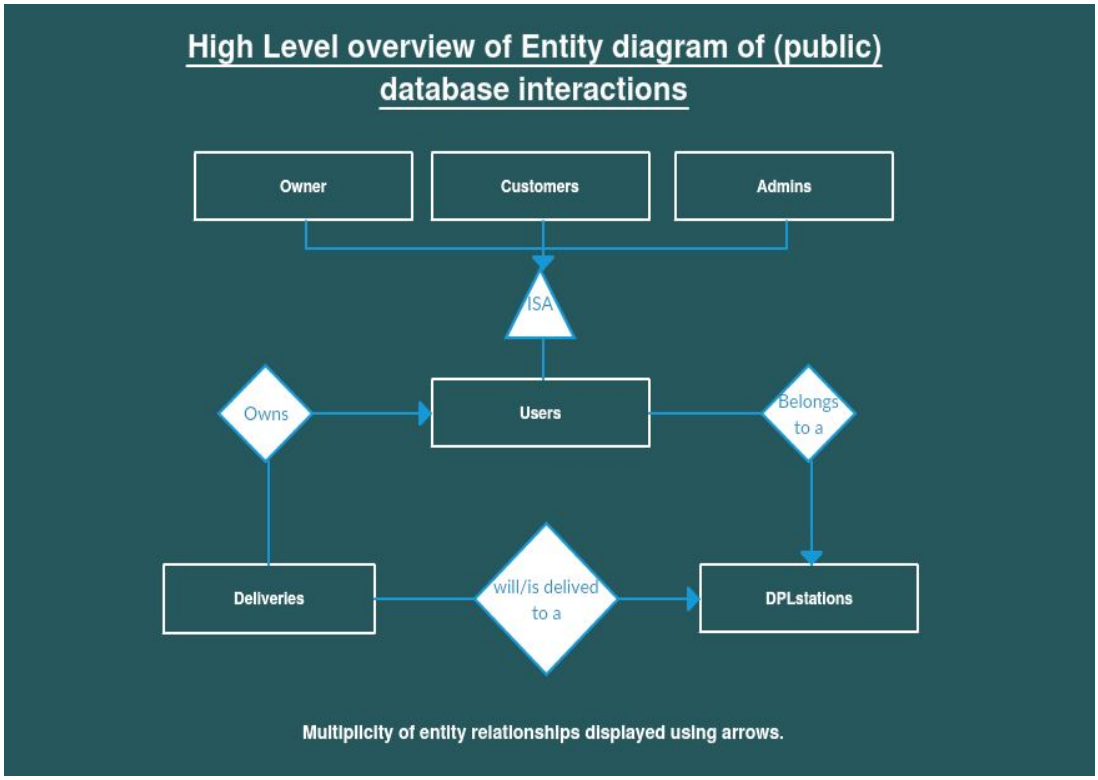

Figure 13: Entity diagram of the database.

Field Name	**Type**	**Constraints**
uid	Integer	primary Key
email	string(120)	Unique
first_name	string(20)	NOT NULL
last_name	string(20)	NOT NULL
password	string(60)	NOT NULL
phone_number	string(16)	N/A
city	string(20)	NOT NULL
address	string(60)	NOT NULL
postal_code	string(6)	NOT NULL

role	Enum [customer, owner, admin]	Default: consumer
dpl_station	integer	Foreign Key, default: 0

Table 12: User Table fields and constraints

Field Name	**Type**	**Constraints**
delivery_id	Integer	primary Key
tracking_number	string(40)	NOT NULL
expected_date	datetime	NOT NULL
registered_date:	datetime	NOT NULL
passcode	string(10)	Unique - NOT NULL
status	Enum [delivered, in_progress, canceled, delayed, picked_up]	Default: in_progress
to_customer	Integer	Foreign Key

Table 13: Deliveries Table fields and constraints

Field Name	**Type**	**Constraints**
station_Id	Integer	primary Key
city	string(20)	NOT NULL
address	string(60)	NOT NULL
total_lockers	Integer	NOT NULL
lockers_available	Integer	NOT NULL
postal_code	string(6)	NOT NULL
owner_id	integer	Foreign Key

Table 14: DPLstations Table fields and constraints

3.3.3. APIs

Our connection APIs, based on the REST API model, will act as a bridge between our web servers and local controllers within the DPL stations. Web Servers and DPL controllers will exchange data regarding package deliveries, using a client-server architecture. Information regarding all customers and packages will be stored in our DB-servers; however, specific information regarding that DPL station will also reside in its local database. This will allow the DPL station to handle services in an efficient manner as the majority of the overhead is within dense queries and sending packets. More importantly, this will enable the local DPL stations to be usable in the case that internet connection cuts off or poor connectivity.

The connection between web-application and DPL station will be a real-time bi-directional event based connection. This bridge of telecommunication between client and server will be enabled using HTTP WebSocket transports [7].

- The following information will be sent to the web-application APIs.
    - UPDATE requests
        - Updating delivery in db servers information upon receiving the package.
- The following information will be sent to the DPL station.
    - UPDATE requests
        - Update customer information
        - Update delivery information (such as tracking number)
    - DELETE requests
        - Delete expected delivery in local database in case of cancelation/expiration
    - POST requests
        - Add new customer to local database
        - Add new delivery to local database

Des 3.3.3.1 - PT	All the system interactions and requests are done in real time.
Des 3.3.3.2 - PT	DPL controller and host server will communicate using Client-server architecture.
Des 3.3.3.3 - PC	APIs are based on the REST API model.
Des 3.3.3.4 - PT	HTTP WebSocket transport is used to send requests between DPL controller and host server.

Table 15: API Requirements

3.3.4 Security

Our goal is to create a secure application and system where information regarding deliveries and personal information are secure. We will be designing our web-application to combat against security threats and vulnerabilities. Furthermore,

Des 3.3.4.1 - FP	The web-application will be safe from SQL-Injection attacks.
Des 3.3.4.2 - FP	The web-application will be safe from CSRF attacks.
Des 3.3.4.3 - FP	The web-application will be safe from cross-site scripting attacks.
Des 3.3.4.4 - FP	User's are authenticated prior to performing any requests or viewing personal and/or sensitive data.
Des 3.3.4.5 - FP	Sensitive information will be encrypted prior to storage in the database.
Des 3.3.4.6 - FP	SHA256 hashing algorithm is used for encryption in the DPL station.
Des 3.3.4.7 - FP	Bcrypt library will be used for encryption in DPL web-application.

Table 16: Security requirements table

3.3.4.1 Web Security Protocols

Majority of security threads are prevented through utilization of industry standard security protocols. We will be covering details on how we will handle numerous of most commonly used and applicable threats to our application.

Since we are using Flask, we're less vulnerable to attacks, as we have less 'surface area' of code exposed. Flask does not contain a pre-defined structure which gives the developers the flexibility to define their layers and structures. As so this gives the engineers more control over handling threads.

We will be using an ORM relational database to prevent sql-injections. ORM packages will detect malicious input by the user, through escaping the input and preventing user input from impacting the behaviour of the query. Cross-site scripting attacks will be handled in a similar manner where user input is escaped using forums, WTForms.

To combat against CSRF, cross-site request forgery, we will be generating CSRF tokens to authenticate requests and make sure they're generated from client-side code. CSRF token will be generated using the secret key for the application [8]. Further on, we have designed our GET requests such that they will be side-effect free.

Security protocols will, also, consist of authentication of users and adding restrictions to protect sensitive information upon specific requests.

3.3.4.2 Encryption

It is very important for DPL to protect its user's information. Since we will be using an external DBMS we must encrypt sensitive information, such as passwords, prior to storing it in DB. This is all due to the fact that we're using a third-party DB server and we're unaware of what might happen behind the scene.

For our local DPL station we will also be encryption user information to protect it against break-ins. We will be using SHA256 hashing algorithm to encrypt data. SHA256 is a relatively slow hashing algorithm to decrypt which will make it ideal for local DB to reduce chances of brute forcing the algorithm.

For the server's side we will be using a different algorithm, specifically Bcrypt. Bcrypt will use the given field and a 'salt' to encrypt information. Additionally, it will also require both user information and salt to decrypt information. In a case of leakage of user information, one would also need the server's secret salt to decrypt information. This makes it ideal for server's side encryption.

3.3.5. Scanner Module

Our Scanner Module makes use of 'pyzbar' open source library. We expand on the pre-existing code to tailor it for our application, to read and detect both QR code and barcode.

3.3.6. Access Code

DPL will generate a ten digit access code for both the courier company and the customer in order for them to access the parcel lockers. This code is decoded into two distinct sections, being a four digit DPL Station ID and a Randomly Generated six digit number.

Figure 14: Access code decoded

The DPL Station ID will be associated with the Country, State/Province, city and a location ID that the DPL station is located. The location ID will help us distinguish DPL stations in cities that have multiple DPL stations. Finally, the Randomly Generated # will be used to further distinguish the access code and enhance the security of the parcel lockers. The number of combinations for a six digit number is shown below in Equation 1.

$$10^6 = 1,000,000$$

Equation 1: The number of combinations for a six digit number

4.Conclusion

This document outlines all the design specifications of DPL and it is intended to be used as a design reference for Parcel Solutions Inc. This document provides a detailed insight on the hardware and software designs required for the development of our product. Specifically, this document focuses on the design of the parcel lockers and the web application.

The parcel lockers consist of all the hardware components in which it will be stored in the Main Console unit as shown in Figure 2. The physical measurements, constraints and design of the DPL station is also illustrated in this document which can be seen in Figure 4 and Figure 2. The DPL station will consist of its own intuitive UI which is separate from the web application. The UI will be presented in the form of a 7-inch display which will be located above the Main Console unit on the DPL station.

The web application portion of DPL will demonstrate our developers programming abilities and skill set. The Web application will be deployed using frontend and backend frameworks in order to institute our website to handle functionalities in the most optimized manner and prevent data leakage. Combining with the server and DBMS, they can handle the concurrent real-time queries and store customers' information adequately. The entity diagram of the DBMS is shown in Figure 13. Additionally, the web application will consist of APIs to bridge a connection between the web application and the DPL station.

Furthermore, this document provides a test plan that will be used to assess the success of DPL against the specified functional deliverables. The test plan will consist of a series of test cases that we will complete to ensure the requirements and design specifications of DPL are justified as proposed.

The design specification will serve as a reference document for Parcel Solutions' team as we begin developing the prototype for DPL in the upcoming months. These designs will serve as a guideline but may change throughout the development cycle for the prototype.

References:

[1] "Requirements Specification for DPL," Parcel Solutions Inc, 2-July-2020

[2] "Buy a Raspberry Pi Touch Display," Raspberry Pi Foundation, [Online]. Available: **https://www.raspberrypi.org/products/raspberry-pi-touch-display/** [Accessed: 7-July-2020]

[3] "4 Channel Relay Board," Electronic Hubs, [Online]. Available: **https://www.electronics-lab.com/project/4-channel-relay-board/** [Accessed: 7-July-2020]

[4] "Specification of Slave Controller AL 2448," Aliexpress, [Online]. Available: **https://www.aliexpress.com/item/32651314309.html?spm=a2g0o.detail.1000023.9.2086654eHbMGPq** [Accessed: 7-July-2020]

[5] “Atoplee 1pc DSCK6656 Holding Force Electric Magnetic Lock for Door Access Control System Electromagnet Fail-Safe,” [Online]. Available:

https://www.amazon.ca/DSCK6656-Electric-Magnetic-Electromagnet-Fail-Safe/dp/B01IB3OIVO/ref=sr_1_1_sspa?dchild=1&keywords=dsck6656+electric+magnetic+lock&qid=1594110572&s=hi&sr=1-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEzVExNWEhKME1CUENWJmVuY3J5cHRlZElkPUEwNDMzMDIwMTE2R05KMTJTUlkzSyZlbmNyeXB0ZWRBZElkPUEwOTQ3OTQ5MTBCRFFNUERZS01FSiZ3aWRnZXROYW1lPXNwX2F0ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU= [Accessed: 7-July-2020]

[6] “Buy a Camera Module V2,” Raspberry Pi Foundation, [Online]. Available: https://www.raspberrypi.org/products/camera-module-v2/ [Accessed: 7-July-2020]

[7] D.Kumar,“Advantages of building applications with Vue.js,” GitConnected, 3-Jan-2019 [Online]. Available:

https://levelup.gitconnected.com/advantages-of-building-applications-with-vue-js-b2a23e4199c8?gi=edf6e3ee7648 [Accessed: 9-July-2020]

[8] “CSRF Protection,” FlaskWTF [Online]. Available:

https://flask-wtf.readthedocs.io/en/stable/csrf.html [Accessed: 9-July-2020]

Appendix A: User Interface and Design Appearance

A.1.Introduction

This section will cover the user interface and design appearance of DPL and provide justification for our products design choices.

A.1.1. Purpose

The purpose of this document is to define and analyze User Interface design of the DPL and to provide an overview of the software and hardware parameters in the design of DPL.

A.1.2. Scope

Primarily, this document focuses on the proof-of-concept and the prototype UI. It contains the required user knowledge, and restrictions as well as the usability testing approaches with designers and potential end users.

A.2. User Analysis

DPL is designed in a way that makes it very simple for the user to utilize the product. The web application portion will have an intuitive UI such that any online shopper can maneuver. The web application will offer a standard log in system similar to other web applications such as Facebook, Instagram, Gmail and etc. Our web application will have a clear and detailed instruction page that will enable the user to use DPL's services.

The parcel locker will have a separate intuitive UI in which the customer and the courier will use to interact with the lockers. The parcel lockers will have a main console box which will contain a 7-inch display unit. The display will be similar to a standard smartphone such that the courier and the customer can easily use to interact with the system.

Some helpful experience which the user may have previously acquired could be experience using an android phone, Iphone or a tablet. Another useful experience could be the use of similar parcel lockers such as Amazon's own Amazon-Lockers.

The intended audience of DPL are people who have the ability to use a computer to purchase items online and are not visually impaired as well. Using DPL will not require any specific physical constraints that the users must meet, as long as the user can go to the lobby of their building and collect their mail, they can use DPL's services.

A.3. Graphical Representation

Figure 17 is the Homepage of DPL's web application. As shown in this Figure, a description of DPL's services is provided for interested customers wanting to learn about DPL. The two options on the top right corner allow the customer to log-in or sign-up and initiate their DPL request.

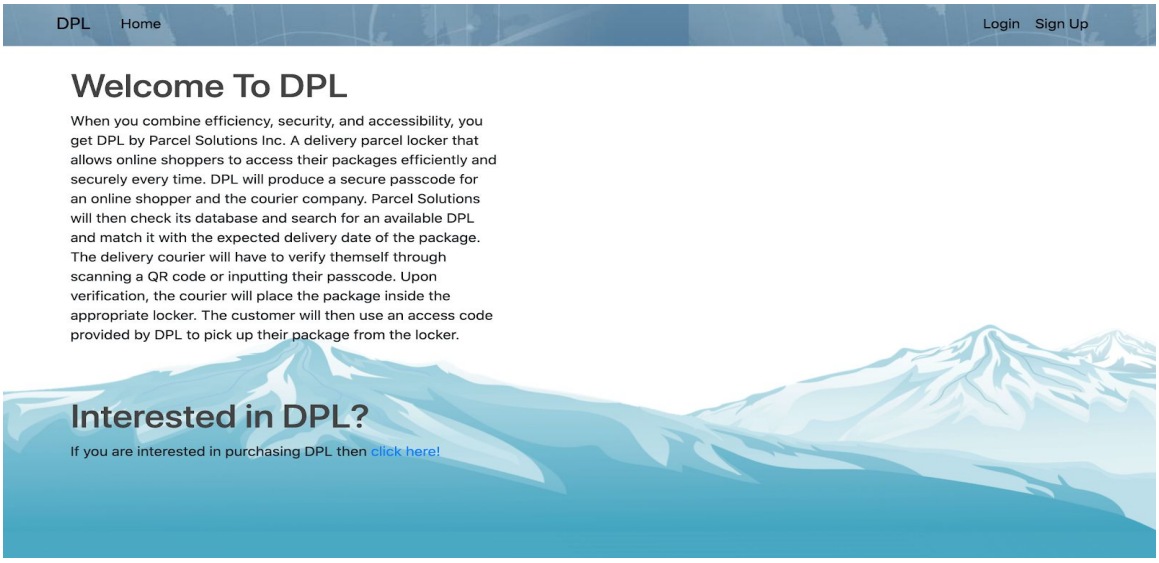

Figure 15: Web Application Home page

Figure 18 shows the log-in page for our web application. As shown below, it is a standard log-in page with a typical forgot password feature and sign-up option.

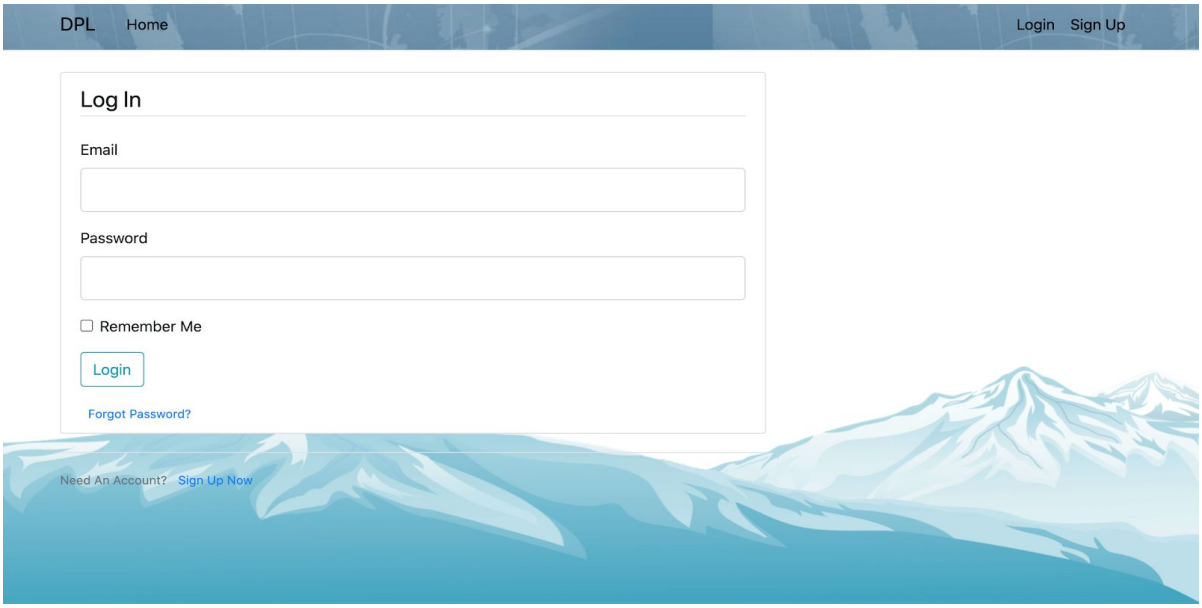

Figure 16: Web Application log-in page

Figure 2 is the prototype design of a final version of the DPL station that will be deployed in apartment lobbies. As shown, it will consist of four components, the lockers, the camera, the display unit, and the main console unit. The number of lockers will be dependent on the number of units in the apartment building. The camera will be used by the courier to scan the customer packages in order to place them in a parcel locker. The display unit can be used by the customer and courier to interact with the DPL station. Finally, the main console unit will contain the hardware components and the majority of the wiring.

The user interface of the display unit can be seen in Figure 19. The courier and customer can enter the provided access code to gain access to the parcel lockers.

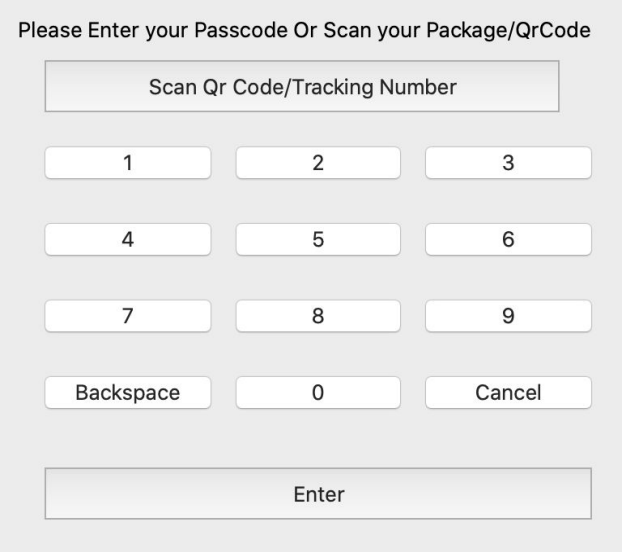

Figure 17: 7-inch display UI of the DPL Station

A.4. Technical Analysis

This section analyzes the user interface and shows how "Seven UI Interaction Elements" outlined in the text of Don Norman have been realized.

A.4.1. Discoverability

Discoverability is an important factor for an intuitive UI. Given the vast variety of layout choices, we must prioritize which elements need to be emphasized in the design. To achieve this, we will be prioritizing the most common functionalities that a user would need. Our focus is to allow users and courier personnel to efficiently use our touch displays GUI to handle deliveries. To focus more on the perspective of users, we have created a list of typical tasks needed to accomplish this design.

- **User needs to add delivery requests**
- **Courier Personal needs to place packages inside parcel lockers:** A barcode scanner has been placed on the main console box for the courier to scan as shown in Figure 2. Alternatively, a display unit has been placed on the main console box for the courier to enter an access code as shown in Figure 2.
- **DPL Owner needs to plug in the station:** a long plug-in cable has been placed on the back side of the station.

A.4.2. Feedback

Feedback is used to engage and explain, and can improve user satisfaction. In order to enhance user experience we have tried to provide feedback at every step.

- When a user registers a delivery their delivery is processed and the output is displayed to them. In the case where the delivery is accepted, a confirmation email will be sent.
- Users will be notified about the status of their delivery via email.
- DPL station's wifi connectivity will be displayed within the GUI.
- Once a user enters a passcode through GUI, the output will be displayed on the screen.
- Once a user/courier-personal closes the door of a locker, we will also confirm that the locker is now closed by displaying the state of the lock on the GUI screen.

A.4.3. Conceptual Models

Conceptual models are used to allow users to have a similar experience where possible. Conceptual modelling can be done at an early point in the design cycle so that there is a fundamental comprehension of how users consider undertakings. A well designed UI communicates all the necessary information to create a conceptual model of the system to allow for a better understanding and feeling of control. In the case of DPL, prior experience with websites and web interactions will be useful in order to create an initial conceptual model. Since the UI for DPL will be intuitive and consistent with current UI design, for both web-application and display GUI, the user would quickly grasp and have a good conceptual model of how it works.

A.4.4. Affordance

An affordance is a quality of an object or an environment that enables an individual to perform a specific action or ability. DPL will enable the user to improve their online shopping experience. This is done by eliminating the need for the user to leave their building to pick up their package from a parcel post office. DPL will have two distinct interfaces, being the 7-inch display on the

main console box and the web application. The 7-inch display will consist of a single page in which the courier and the customer can enter the access code to gain access to the lockers. The web application contains several different web pages, each consisting of a limited number of buttons that are well-labeled. Any graphical display on the web pages will display only critical information.

A.4.5. Signifiers

Signifiers are used to provide extra information on how to use the product's main elements. DPL uses Touch Display and notification emails as signifiers. Within the touch display there are many icons describing the state of the machine whether as in whether it is connected to the internet, is in sync with servers and other useful information that are displayed to the user.

A.4.6. Mapping

The relationship between controls and their actions follows the principles of good mapping, enhanced as much as possible through spatial layout and temporal contiguity. Since our goal is to have a minimalistic and efficient UI, the general usage of it will be simple. Our user interface will avoid unnecessary elements and are clear in the language they use on labels and in messaging.

Our touch displays UI will contain only a few buttons which will have a well defined label. Our goal is to allow both users and courier-personal to interact with the DPL station in a very quick manner through conceptual manner. Our web-application will follow a similar manner. It will only consist of a few, less than 10, routes. Additionally it will also be minimalistic to allow visitors see the core elements of the interface and make their journey intuitive and memorable.

Further on, in terms of physical placements, all ports are placed on the rear end of the station so that the wires are out of the way. Touch Display is placed in the front so the user can interact with it.

A.4.7. Constraints

Constraints are limitations that may restrict a user's interaction with DPL. The DPL lockers in apartment lobbies will require constant power for the display to work. Without power, the system will be turned off. Users will also require access to a computer or a smartphone in order to access the web application in order to make a request for DPL services. Customer packages must also not exceed the size of the parcel lockers, or else DPL is unable to provide its services.

A.5. Engineering Standards

DPL shall comply with various well-established industry engineering standards relating to the design of user interfaces. Therefore, the user interface of DPL must be built and tested by the following Engineering Standards and Safety Standards.

IEEE 1621 - 2004	IEEE Standard for User Interface Elements in Power Control of Electronic Devices Employed in Office/Consumer Environments [1]
IEEE 1012 - 2012	IEEE Standard for System and Software Verification and Validation [2]
ISO 9241 - 161 - 2016	Ergonomics of human-system interaction – Part 161: Guidance on visual user-interface elements [3]
ISO/IEC 13251:2004	Collection of graphical symbols for office equipment [4]
IEC TR 61997:2001	Guidelines for the user interface in multimedia equipment for general purpose use [5]

Table 17: Engineering Standards

IEC 61508 – 2 : 2010	Requirements for electrical/ electronic/ programmable electronic safety-related systems [6]
IEEE C2 – 2017	2017 National Electrical Safety Code(R) (NESC(R)) [7]
EU RoHS 2	Directive 2011/65/EU on the restriction of the use of certain hazardous substances in electrical and electronic equipment [8]

Table 18: Safety Standards

A.6. Analytical Usability Testing

This section will outline the analytical usability testing taken by the designers, using the heuristics evaluation. This evaluation is a report used to identify problems in the user interface design and is in the form of a checklist. The designers complete it independently at least twice. The heuristics are the following:

- General

    - Instructions on the use of the system are either visible or easily retrievable
    - Any text or pop-up messages are reduced and held to a minimum so as not to decrease the relative exposure of important information.
    - Errors messages are displayed in plain language, indicate the problem and suggest a solution
    - Pop-up warnings are displayed in plain language, demonstrate the problem and propose a solution.
    - there are no spelling mistakes in the field labels and web page headings
- DPL station (through Touch Screen)
    - The passcode and barcode scan inputs are authenticated within a reasonable time frame, under ~5 seconds.
    - The UI/GUI does not contain any engineering specific technical terms, instead only makes use of conventional vocab.
    - UI elements must not be overlapping in Touch Display

- Web-Application
    - “Delivery request” and “home” page routes are available at all pages.
    - Hyperlinks are visible and distinguished from other text.
    - elements are not breaking or distorted on different browsers and screen resolutions
    - there are no broken links on the web page
    - Confirmation on successful action on each step completion

A.7. Empirical Usability Testing

Empirical testing will be incorporated as part of our development process in order to receive feedback from potential users. This will only occur at the prototype stage before finalization, as we believe it is better to provide users with a complete sample of our product. In the earlier stages of our design, key components will be in development, thus conducting a user test at these stages will be less effective as their feedback is only limited to how the system works at that time.

To perform the usability testing in this stage, a selection of students and faculty members from the faculty of applied sciences who can evaluate our product in a more technical way will be provided with our product to conduct testing. This testing will also be conducted with non-technical members from other faculties. This will ensure that we are provided with constructive feedback from both technical and non-technical users. During the testing phase, we will provide documentation that outlines the intended usage, safety, and troubleshooting information of the product, and encourage users to interact with all major components.

Additionally, the team members of Parcel Solutions will be on site to ensure the testing phase is conducted smoothly. An overview of the system will also be available in the documentation to help participants understand how different features interact with one another.

During this period of Empirical testing, comments, feedback, suggestions, complaints and bugs/errors are to be recorded by the user. We will maintain contact with all the users and provide support as necessary. At the end of the Empirical testing period, we will ask the participants to complete a survey to help us further understand their overall experience with DPL.

All the collected comments, feedback, and suggestions will be reviewed and we will update our design accordingly to produce the final product. Finally, we will run one last iteration of unit and integration testing before releasing the final version of our product.

A.8. Conclusion

User Interface Design is a crucial aspect of any product. It enables users to learn and use the product without having to worry about any hurdles along the way. During the implementation phase it's very easy for developers to overlook the fact that many users may not have the same technical skills as others, so it's important to design an interface that caters to such people. With this goal in mind, along with Don Norman's text, we have identified key qualities and components necessary for an intuitive and powerful interface. Parcel Solutions Inc plans to demo DPL to fellow students and online shoppers at Simon Fraser University for feedback on ease-of-use since these people are our intended audience. For the proof-of-concept design, we aim on providing a minimalistic design, while showing the basic functionalities of all the locks and their ability to work concurrently. For the prototype version, Parcel Solutions' goal is to have an intuitive and easy-to-use UI while meeting all the requirements of the prototype stage.

Reference

[1] “1621-2004 - IEEE Standard for User Interface Elements in Power Control of Electronic Devices Employed in Office/Consumer Environments,” IEEE. [Online]. Available: https://standards.ieee.org/standard/1621-2004.html. [Accessed: 9-July-2020]

[2] “1012-2012 - IEEE Standard for System and Software Verification and Validation,” IEEE. [Online]. Available: https://standards.ieee.org/standard/1012-2012.html. [Accessed: 9-July-2020]

[3] “ISO 9241-161:2016,” ISO, 17-Feb-2016. [Online]. Available: https://www.iso.org/standard/60476.html. [Accessed: 9-July-2020]

[4] “ISO/IEC 13251:2004”, ISO, 2004 [Online]. Available: https://www.iso.org/standard/21486.html. [Accessed: 9-July-2020]

[5] "IEC TR 61997:2001 | IEC Webstore", Webstore.iec.ch, 2018. [Online]. Available: https://webstore.iec.ch/publication/6269. [Accessed: 9-July-2020]

[6] “IEC,” IEC 61508-2:2010 | IEC Webstore | functional safety, smart city. [Online]. Available: https://webstore.iec.ch/publication/5516. [Accessed: 9-July-2020]

[7] “C2-2017 - 2017 National Electrical Safety Code(R) (NESC(R)),” IEEE. [Online]. Available: https://standards.ieee.org/standard/C2-2017.html. [Accessed: 9-July-2020]

[8] "RoHS Compliance Guide: RoHS 10 Restricted Substances", Rohsguide.com, 2018. [Online]. Available: http://www.rohsguide.com/rohs-substances.htm. [Accessed: 9-July-2020]

Appendix B: Supporting Test Plan

In order to enhance the accuracy of the system outcome and meet the functional standards, the testing outlines will be divided into three sections:

- Hardware testing
- Software testing
- Safety design testing

All three test plans will be emphasized on the design specification in order to perform our primary goals and requirements of DPL.

B.1. Hardware Testing

The hardware testing of DPL will consist of three components: the camera detection module, the lock module and then feedback module.

B.1.1. Camera Detection Module

The testing plan of camera detection module include camera and Raspberry Pi:

Test Case	Verification	Result*	Comments
Accuracy of the sensing range and width of camera	The distance of detection of QR code should be consistent for every test trial for camera	Pass □ Fail □	
False detection - place the QR code away from	The camera should receive an error when	Pass □ Fail □	

the camera	the QR code is out of range		
Reading speed of the camera - Place the QR code in reading range and time how long it takes	The camera should takes less than 1 second to read if the QR code is in the reading range	Pass □ Fail □	
Max reading angle - Place the QR code in different angle and orientation	The QR code should be read and identified for different angle and orientation	Pass □ Fail □	

Table 19: Test Plan for Camera detection module

* Pass = functionality meets standards

Fail = functionality does not meet standards. Further description will be provided in "Comments" column

B.1.2. Lock Module

Test Case	Verification	Result*	Comments
Transmitting speed of relay - Press the lock or unlock buttons on touch display and time how long it takes	The relay should takes less than 1 second to lock or unlock after receive the command from users	Pass □ Fail □	
Accuracy of relay - Press the unlock button on touch display and check whether the corresponding locker is opened	The corresponding locker should be opened after receive the command from users	Pass □ Fail □	
Transmitting speed of slave controller - Open or close the locker and time how long the door status appears on the display	The door status should be appeared on display less than 1 second after open or close the locker	Pass □ Fail □	
Accuracy of slave controller - Open or	The correct door status should be presented on	Pass □ Fail □	

close the locker and check the door status on the display	the display after open or close the locker		

Table 20: The testing plan of lock module containing signal relay, slave controller and Raspberry Pi

* Pass = functionality meets standards

Fail = functionality does not meet standards. Further description will be provided in "Comments" column

B.1.3. Feedback Module

The testing plan of feedback module contains touch display and Raspberry Pi:

Test Case	Verification	Result*	Comments
Touch Display format display	The touch display should display the appropriate format to let the users able to read	Pass □ Fail □	
Accuracy of touch display at different power levels	The touch display should work properly and accurately at different power levels	Pass □ Fail □	
Accuracy of the buttons on display - Press the buttons	Buttons on touch display are able to execute their corresponding functions	Pass □ Fail □	

Table 21: The testing plan of feedback module containing touch display and Raspberry Pi

* Pass = functionality meets standards

Fail = functionality does not meet standards. Further description will be provided in "Comments" column

B.2. Software Testing

The software testing of DPL will consist of three sections: the DPL station, the database server and the web application.

B.2.1. DPL Station System

Test Case	Verification	Result*	Comments
Must correctly authenticate users based on passcode input	The processed input should be cross-referenced with inputs in the local database.	Pass □ Fail □	Pass/Fail upon correct authorization output to different users.
Must correctly authenticate users based on QR input	Upon successful reading of QR, the demodulated reading must be cross referenced with codes in the local db.	Pass □ Fail □	Pass/Fail upon correct authorization output to different users.
Must correctly identify a delivery based on the tracking number barcode	The camera should takes less than 1 second to read if the QR code is in the reading range	Pass □ Fail □	Pass/Fail upon correctly detecting or discarding provided tracking number based on stored data.
Must make sure that a QR/Passcode is expired after the package is picked up.	The authentication token is not accepted once it has been used.	Pass □ Fail □	The used authentication tokens will not be used in future.

Table 22: The testing plan for the presentation layer of DPL station.

* Pass = functionality meets standards

Fail = functionality does not meet standards. Further description will be provided in “Comments” column

B.2.2. Database Server

Test Case	Verification	Result*	Comments
DBMS will not store any input not satisfying the constraints.	DBMS should raise an exception in the case where an input is not satisfying constraints.	Pass □ Fail □	The constraints are the following: Type, Null, Unique, Length, Primary and Foreign Key constraints.

Protection against SQL-Injection	DBMS and ORM procedures must detect malicious inputs such as codes or comments in the input and parse them before storing.	Pass □ Fail □	
Multi-threaded regression Testing for concurrency and data integrity	Upon performing regression testing the finalized state of database must be as expected.	Pass □ Fail □	Performing regression testing using external tools.

Table 23: The testing plan Database server.

* Pass = functionality meets standards

Fail = functionality does not meet standards. Further description will be provided in "Comments" column

B.2.3 Web Application

Test Case	Verification	Result*	Comments
Functionality: - Links - Forms - Cookies - HTML & CSS	Test Forms are working correctly. All links are working correctly and no broken links. Testing cookies (sessions) are deleted either when cache is cleared or when they reach their expiry. Delete cookies (session), login credentials are asked for when you visit the site. Ensure that search engines can crawl our site easily.	Pass □ Fail □	
Interface: - Application - Web Server - Database	Test requests are sent correctly to the DB and output client side is displayed correctly.	Pass □ Fail □	

Server	Web server is handling all application requests without service denial. Queries sent to database give expected results		
Performance: - Stress testing - Load testing - Crash testing	Ensure that the site works under all loads. Ensure the website is able to recover from a crash.	Pass □ Fail □	
Security	Verify unauthorized access to secure pages should not be permitted. Sessions are automatically killed after prolonged user inactivity	Pass □ Fail □	Restricted files, information, URLs should not be accessed without appropriate access.

Table 24: The testing plan web-application.

* Pass = functionality meets standards
Fail = functionality does not meet standards. Further description will be provided in "Comments" column

B.3 Safety Design testing

Safety design testing is the most important test to prevent the hazard of using DPL improperly. As such, performing a comprehensive safety test is key for ensuring users' safety. The following test plan is as follows:

Test Case	Verification	Result*	Comments
Waterproofing - Perform water injection test to DPL	DPL should resist the penetration of water and none of the component is soaked	Pass □ Fail □	
Wiring enclosure - All the wires place at a enclosed connection	All wires should not be exposed to the users.	Pass □ Fail □	

points			
Overheating	The components of DPL should not be overheated and should operated properly	Pass □ Fail □	Operate DPL in an outdoor area for at least 8 hours
Wire protection - Use a non-contact voltage detector to measure all the voltage of wires	The maximum value from the detector should not be greater than 9V. All the wires should have plastic cover to prevent electrical shocks	Pass □ Fail □	

Table 25: The testing plan for safety design.

* Pass = functionality meets standards

Fail = functionality does not meet standards. Further description will be provided in “Comments” column