

Project Deliverable G: Prototype II and Customer Feedback

GNG 1103 - Engineering Design

Section #A03

Team #12

Feryza Damji, Katrine Labonte, Anissa Millington, Zhehao Xu, Dawood Kalair

November 11, 2021

Faculty of Engineering

University of Ottawa

Introduction

Prototyping is the most crucial aspect of the design process. It allows for trial and error, while providing ample customer and supervisor feedback. This deliverable will provide a deeper understanding of our second prototype. It will discuss its creation based on feedback from the first prototype, as well as the testing and plans for the third prototype.

We created a plan and estimated the costs of our project's components during the final deliverable. As a result, we developed our second prototype.

For the interface, we decided that creating a second prototype would be futile at this point. This is because the next step would be adding functionality to the interface, which is dependent on prototypes 2a and 2b being complete.

Prototypes

Point Use Efficiency:

Define Efficiency

- the ratio of the points spent to the points earned+current balance
- anything over 0.5 is efficient
- anything under 0.5 is not efficient

Reminders

- if under 0.5 then reminder once a week
- if under 0.2 then reminder twice a week

Curation of Deals

- for everyone, Today's Deals (app), Deals Page (website) would be curated based on customer stats and past spending
- deals for bank items would be based on age

Functional Algorithm

Step 1: Start

Step 2: Declare variables PS, PE, CB, R, E, NE //(PS=points spent, PE=points earned, CB=current balance, R=ratio, E=efficient, NE=non-efficient)

Step 3: Initialize (ask for values) variables PS, PE, CB

Step 4: Read variables PS, PE, CB, E, NE

Step 5: Calculate R

$$R = (PS) / (PE + CB)$$

Step 6: If $0.2 < R < 0.5$

R=NE

Display shows 1 reminder and deals

Else

If $R \leq 0.2$

Display shows 2 reminders and deals

Step 7: If $R \geq 0.5$

$R = E$

Display shows deals

Step 8: Stop

Explanation of Point use Efficiency

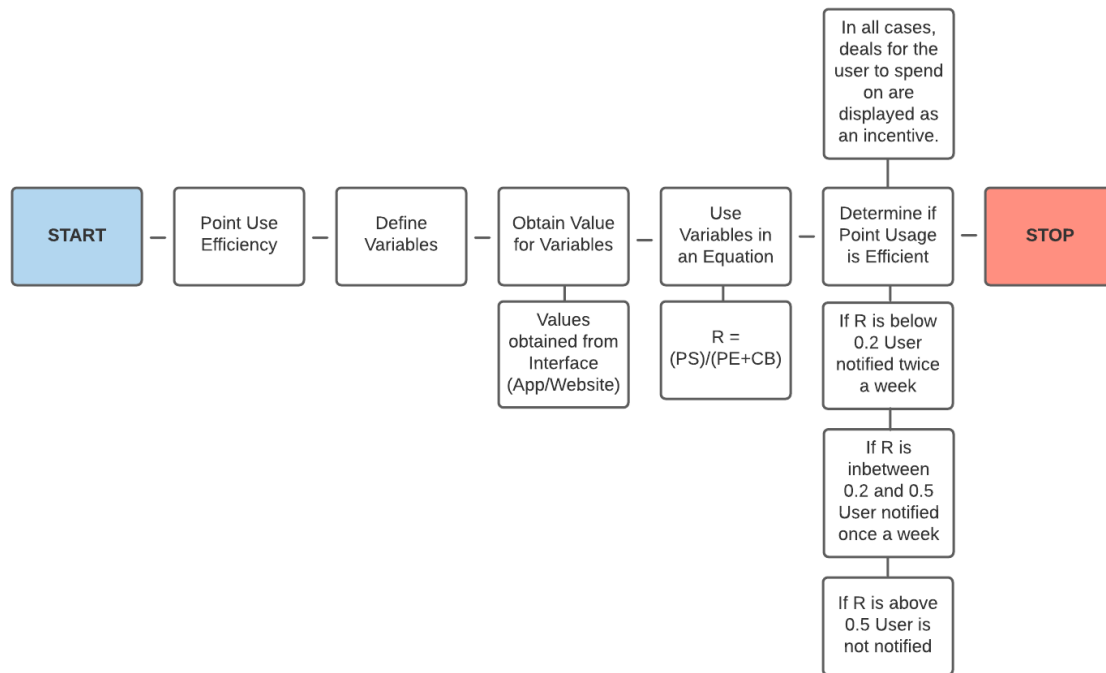
To begin, we've set a defined criteria for what the program will deem as an efficient or inefficient use of loyalty points. We've approached this with methodology involving calculated ratios. The process is initiated with a simple declaration of variables that will be used. These variables include:

Variables	Explanation
PS	The amount of points spent by the customer previously.
PE	The amount of points earned by the customer.
CB	The customer's current balance.
R	The calculated value (ratio) for efficiency.
E	Display indicating efficient usage.
NE	Display indicating inefficient usage.

The next step conducted after the declaration of variables was to receive the values for each variable. The values will be obtained from the user interface (website/app). After the values have been read by the program, it will then process these obtained values into an equation that was formulated for the calculation of our ratio "R". The formula being used is: **$R = (PS)/(PE+CB)$**

After the value for R has been calculated, we've set a range that will help the program understand what values for the ratio is considered to be efficient or inefficient. If the ratio is below or equal to 0.2, the point spending is inefficient, and so the program will notify the user through the app that your usage is inefficient as well as ongoing deals the user can spend on. The frequency for the reminder will be twice a week. If the ratio is between 0.2 and 0.5, spending is inefficient, however the user will be notified only once a week as well as be given a list of deals they can spend on. Finally, if the ratio is calculated to be greater than 0.5, the point usage is efficient and only a list of deals for spending will be displayed. This entire process will be integrated using a coding language of some sort and will be put into the application/website that is responsible for including the interfaces.

Flowchart



Point Democratization:

Define Point Democratization:

The goal that this subsystem of the prototype hopes to achieve is the conversion of points based on predetermined factors between different companies involved in the Zafin program. The points will be converted into a standard point currency that will be used on the user interface this group will engineer.

Code:

CODE

DefaultClass File:

```

package defaultPackage;

import java.util.ArrayList;
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.HBox;
import javafx.scene.text.Font;
import javafx.stage.Stage;
import javafx.scene.control.ComboBox;

```

```

@SuppressWarnings("restriction")
public class classDefault extends Application {
    //global variables
    TextField txtVendorPts;
    Label lblResult;
    BorderPane root;
    GridPane inputArea;
    ComboBox<String> options;
    ArrayList<Vendor> vendorDataBase;
@SuppressWarnings("unchecked")
@Override
public void start(Stage myStage) throws Exception {
    // variables for start method
    final int GAP = 15;
    final int LARGE_FONT = 25;

    // Setting up BorderPane
    root = new BorderPane();

    // setting up gridpane inputArea where all labels, input buttons and textFields
    // will be located
    inputArea = new GridPane();
    inputArea.setHgap(GAP);
    inputArea.setVgap(GAP);
    inputArea.setPadding(new Insets(GAP, GAP, GAP, GAP));
    root.setTop(inputArea);

    // setting up convert Label
    Label lblConvert = new Label("Convert");
    lblConvert.setFont(Font.font(LARGE_FONT));
    inputArea.add(lblConvert, 0, 0, 9, 1);

    // setting up "to" Label
    Label lblTo = new Label("to");
    lblTo.setFont(Font.font(LARGE_FONT));

    // setting up vendorPts TextField
    txtVendorPts = new TextField();

    //setting up dropdown menu
    @SuppressWarnings("rawtypes")
    final ComboBox comboBox = new ComboBox();
    comboBox.getItems().add("Sephora");//will implement this to be linked to vendorDataBase

    // setting up vendorPts HBox
    HBox hbxVendorPts = new HBox(GAP, txtVendorPts, comboBox);
    inputArea.add(hbxVendorPts, 0, 1, 9, 1);

    // setting up vendorPts Label
    lblResult = new Label();

```

```

//setting up dropdown menu
@SuppressWarnings("rawtypes")
final ComboBox comboBox2 = new ComboBox();
comboBox2.getItems().add("Walmart");//will implement this to be linked to vendorDataBase

// setting up vendor2Pts HBox
HBox hbxVendor2Pts = new HBox(GAP, lblTo, comboBox2);
inputArea.add(hbxVendor2Pts, 0, 2, 9, 1);

//setting up result prompt label
Label lblPrompt=new Label("This is equivalent to: ");

//setting up result HBox
HBox hbxResult = new HBox(GAP, lblPrompt, lblResult);
inputArea.add(hbxResult, 0, 3, 9, 1);

// setting up convert Button
Button btnConvert = new Button("Convert");
inputArea.add(btnConvert, 0, 4);
btnConvert.setOnAction(event -> convert());

// setting up screen display
Scene scene = new Scene(root);
myStage.setMinWidth(1000);// set screen width
myStage.setHeight(700);// set screen height
myStage.setTitle("Conversion");
myStage.setScene(scene);
myStage.show();
}

private void convert() {
    //note: will implement user entering vendor data which will be stored into database
    Vendor vendor1 = new Vendor("Sephora", 100); // making vendor
    Vendor vendor2 = new Vendor("Walmart", 10000); // making vendor
    vendorDataBase = new ArrayList<Vendor> ();//will be implemented in dropdowns
    vendorDataBase.add(vendor1);
    vendorDataBase.add(vendor2);

    double vendor1pts = Double.parseDouble(txtVendorPts.getText());
    double result= calculateConversion(vendor1pts, vendor1, vendor2);
    updateLabel(result, vendor2);

}

private void updateLabel(double result, Vendor vendor2) {
    lblResult.setText(result + " " + vendor2.getVendorName()+ " points")
    ;
}

private double calculateConversion(double ptsToConvert, Vendor vendor1, Vendor vendor2) {
    double convertTo = vendor1.getConversionTo();
    double convertFrom = vendor2.getConversionFrom();
    double result = ptsToConvert * convertTo *convertFrom;
    return result;
}

```

```
    }  
  
}
```

Vendor File

```
package defaultPackage;  
  
public class Vendor {  
    String name;  
    int userNumber;  
    public Vendor(String vendorName, int totalUserNumber) {  
        name=vendorName;  
        userNumber=totalUserNumber;  
    }  
  
    public int getUserTotal() {  
        return userNumber;}  
  
    public String getVendorName() {  
        return name;}  
  
    public int getLevel() {  
        int level=0;  
        if(getUserTotal()<=100) {  
            level=1;  
        } else if(100<getUserTotal() && getUserTotal()<=1000) {  
            level=2;  
        } else if(1000<getUserTotal()) {  
            level=3;  
        }  
        return level;  
    }  
  
    public double getConversionTo(){  
        double conversion=0;  
        if(getLevel()==1){  
            conversion=0.1;  
        } else if(getLevel()==2) {  
            conversion=0.5;  
        } else if(getLevel()==3) {  
            conversion=2;  
        }  
        return conversion;  
    }  
  
    public double getConversionFrom(){  
        double conversion=0;  
        if(getLevel()==1){  
            conversion=10;  
        } else if(getLevel()==2) {  
            conversion=2;  
        } else if(getLevel()==3) {
```

```
        conversion=0.5;
    }
    return conversion;
}
```

Conversion

Convert

to

This is equivalent to:

Conversion

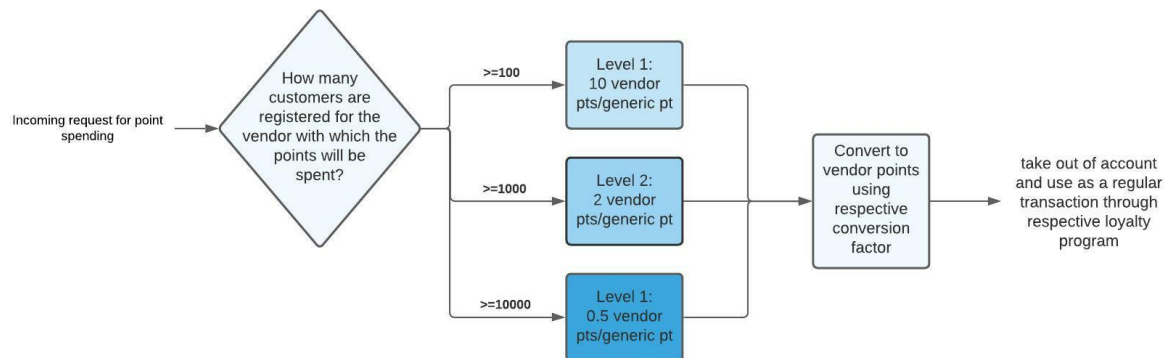
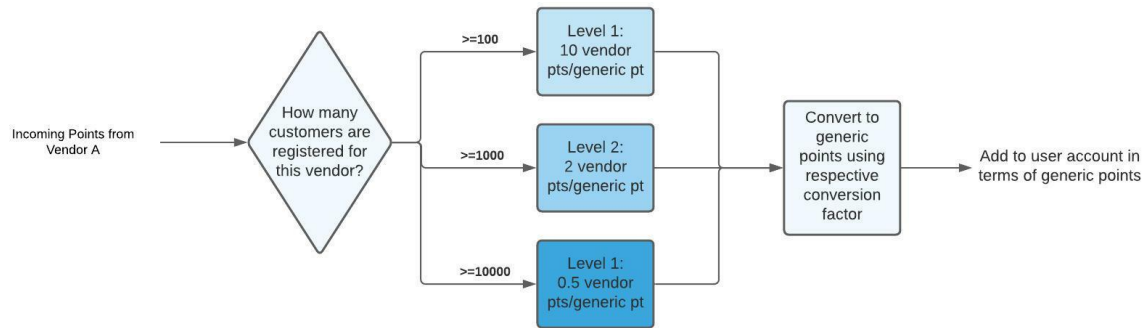
Convert

1000

to

This is equivalent to: 50.0 Walmart points

Flowchart:



Explanation

The process begins with obtaining a value for points that's coming from a vendor into the engineered program. Once the value has been received, the evaluation of each predetermined factor required is accessed. The first question that is asked to the program is the number of customers that are registered for the respective vendor. The answer to the question results in one of the three following cases that determine the value:

- ❖ Level 1: 10 Vendor Points/Generic Points || If the number of customers is greater than or equal to 100.
- ❖ Level 2: 2 Vendor Points/Generic Points || If the number of customers is greater than or equal to 1000.
- ❖ Level 3: 0.5 Vendor Points/Generic Points || If the number of customers is greater than or equal to 10000.

Based on these three classifications, the generic points are calculated using the vendor points and finally added into the user's account.

The process continues by now accounting for when the user decides to spend points in contrast to the previous case where the user was obtaining them. This process starts with an incoming request for point spending. Once the value has been obtained, a question is asked to the program that entails the following: “How many customers are registered for the vendor with which the points will be spent?” Once again, the answer of this question will help with the classification type for how the points will be converted. The three classifications that determine the value are:

- ❖ Level 1: 10 Vendor Points/Generic Points || If the number of customers is greater than or equal to 100.
- ❖ Level 2: 2 Vendor Points/Generic Points || If the number of customers is greater than or equal to 1000.
- ❖ Level 3: 0.5 Vendor Points/Generic Points || If the number of customers is greater than or equal to 10000.

Based on these three classifications, the generic points are calculated using the vendor points and finally removed from the user’s account.

Prototype Testing Analysis

Point Use Efficiency

To test the point use efficiency coding plan, the previously displayed algorithm was presented to each individual and was briefly explained so that the critics had some context as to what our prototype entailed. The goal for the testing was to understand whether or not the results of our prototype manufacturing so far was detailed yet simple to fathom. We wanted to know whether or not our plan for this subsystem of our prototype was relevant and useful for the final presentation of our engineered product. We also aimed to obtain any possible feedback to better our prototype before we present it to the clients in order to narrow down any points for potential flaws. This would help our clients pick between specific points of our prototype, allowing for a thoroughly efficient, feasible and quality final design. A risk that will be accounted for is a lack of simplicity in terms of design. This will be dealt with by making sure the interface is well made and fluid in terms of user understanding. Due to the measures that will be taken, the risk of this problem impacting our prototype will be diminished.

Point Democratization

To test the point democratization aspect of our prototype, a multi step process was conducted in order to reach an accumulation of desired results as far as feedback goes. To begin with, the code written for the conversion of points is tested for functionality, once that has been done it will be presented to a number of critics. The goal for the testing is to make sure that it is user friendly and that the functionality of the program works fluidly and efficiently in time for the final presentation. A risk accompanied with this section of the prototype is the possibility of

dysfunctionality of the program. This will be accounted for by doing numerous tests before any sort of presentation of the prototype. Each and every case for different outputs will be gone through numerous times as well as checked by each group member on their respective hardware devices.

Summary of Prototype 2 Testing

Prototype ID	Test Objective (Why?)	Description of Prototype used and of Basic Test Method (What)	Test duration and planned start date (When)
2a	Verifying feasibility	Point Use Efficiency Algorithm Code Logic Medium fidelity, test logic feasibility to ensure writing such a code is possible and feasible	2-3 days November 8
2b	Verifying feasibility	Point Democratization Algorithm Code Medium high fidelity, test with entering false data to see if code is accurately displaying information	2-3 days November 7

User Feedback

We gathered feedback from a computer programmer, a technology center manager at Bell, and a technology and IT manager at Bell. After creating prototype II with the coding for point democratization and point use efficiency to implement into the app for the user interface and the website for the bank interface. The point use efficiency is an algorithm but will be implemented into the javascript to make it functional. We asked if the code could be functional and if it could be composed into a script. We received feedback that the algorithm is functional and can be written in a script with minor adjustments. Further analysis of the code for point democratization is still being reviewed by all three members that were asked to give feedback, the results will be available for the next prototype. However, the implemented functions so far output accurate results when data is manually fed into the algorithm, so the feasibility of the code has been checked and approved. For the next prototype we will work on implementing the algorithm into a script to make it functional. We were also able to improve prototype I by using

the feedback we received and implemented it into our prototypes. Which will be seen in the finalized prototype design.

Detailed Design

After tests and feedback was received from uninvolved individuals, the resultant data recordings were listed as follows.

- **Subsystem 1(1): For the end-user Interface (application):**
 - Have a successful login page
 - A home page
 - A search engine at the top right corner
 - A help icon
 - A chat tab for customer to get information
 - the stores that the customer has selected as their favourites
 - the point balance of the customer
 - the deals available for the customer that day
 - the amount of points they have and need to complete a milestone
 - an account/setting icon
 - a dropdown menu in the left hand corner:
 - ❖ a conversion calculator
 - ❖ a contact us link
 - ❖ adding a new loyalty program
 - ❖ linking accounts and referrals
 - ❖ a points tracker
 - ❖ suggestions for effective spending
 - ❖ a section for past transactions

- **Subsystem 1(2): For the Bank Data Analysis (website):**
 - Section for transacting points
 - A secured place to store transaction records
 - A place to set values for points
 - A place to add new loyalty programs or alter existing programs
 - Demographics show customers' information
 - A statistics tab that shows statistics per vendor and per customer
 - The vendors' statistics includes:
 - ❖ Demographics
 - ❖ Percentage of returning customers
 - ❖ A points overview
 - The customers statics includes:
 - ❖ Top vendors
 - ❖ The top way that points were received and redeemed

- ❖ Amount of points collected and redeemed.

- **Subsystem 1(3): For the Vendor Interface (website):**

- Have early access to upcoming deals involving their store(s)
- Have access to their stores' sales statistics that involve points
- Stores' sales statistics include:
 - ❖ Percentage of returning points shoppers
 - ❖ The amount of points collected through sales
 - ❖ The amount of points redeemed at their store
 - ❖ The number of users that are affiliated with the points program

- **Subsystem 2: Point Democratization**

- Assign respective values to each organization's loyalty points
- Convert all organization's points to a neutral point system
- Ability to assign point value based on predetermined factors
 - ❖ Organization size
 - ❖ Customer demographic
 - ❖ Bank demographic
- Convert points from one loyalty program to another
- Convert points into a value for real life currency respective to each country

- **Subsystem 3: Point Use Efficiency**

Definition:

- the ratio of the points spent to the points earned+current balance
- anything over 0.5 is efficient
- anything under 0.5 is not efficient

To promote point use efficiency:

- reminder :
 - if under 0.5 then reminder once a week
 - if under 0.2 then reminder twice a week
- Provide user with efficient spending advice
- Display a variety of recommended products/services to spend points on
- Ability to analyze customer data and advertise products accordingly
- Invent incentives to avoid point hoarding
 - ❖ Point expiry dates
 - ❖ Promotional events
 - ❖ Discounts
 - ❖ Persuasion through product/service advertising
- Deals offering
- Offer rewards as listed in the previously illustration of point use efficiency

- Point promotions (seasonal x2 earning events)
- Product/service advertising relative to current customer point balance

Target Specifications

Zafin is a financial technology enterprise with a global presence. Their primary objective is to supply banks with software that enhances customer connections and boosts income. Zafin has requested our help in developing a new platform with the primary purpose of democratizing loyalty rewards points. The design criteria needed to satisfy the client's needs and the clients' priorities from analyzing the target specifications. The design criteria and target specifications that we established will allow us to design and ameliorate our prototypes. We have incorporated each design criteria into our prototypes to ensure that our prototypes meet the client's aspirations and what they emphasized in our meetings.

Bill of Materials

Table 1: Bill of Materials

Item Number	Item Name and Description	Quantity	Unit Price	Amount
1	Powerapps:	1	\$0	\$0
2	Javafx:	1	\$0	\$0
3	Figma:	1	\$0	\$0
4	Moqups:	1	\$0	\$0
5	PC/Mac:	1 (per person)	\$0 (previously owned)	\$0
6	Excel:	1	\$0	\$0
7	Mouse	1 (per person)	\$0 (previously owned)	\$0
8	Wix:	1	\$0	\$0

9	Canva:	1	\$0	\$0
10	Lucid:	1	\$0	\$0

The BOM showcases all of the potentially usable apparatus our group plans to utilize to bring our project plans into reality (may be modified in the future). This list includes many software and hardware that will be of significant use in the engineering process of our prototypes. The resources and their uses include:

Powerapps is a microsoft software that allows you to use a large variety of business apps to construct an application tending to your needs. This can be used to add functionality to our prototypes as well as design user interfaces with the respective features our subsystems entail.

Javafx is a developer platform that can be utilized in order to create a software or application. This is done through the means of coding. Javafx can be used to code and add functionality to our features that adhere to our project plans. This application will be relied on in the creation of the second prototype.

Figma is a website which allows for many prototype configurations through the means of simple graphic design. This will be useful for the second and third prototype construction process due to the aesthetic and functional aid it can provide us with. Figma can be used to create an outline or a goal and potentially even a final product if it adheres to our wants and needs.

Moqups is a tool used to aid in collaborative efforts. This can be used to make diagrams or visuals crucial in helping the engineering process of our prototypes. Moqups may be used in the first prototype process due to the visual aid aspect that it can provide us with. This will allow the team members to have an easier to follow linear path in creating a finished product.

PC/Mac is a piece of hardware necessary for everything relating to this project. This is a requirement for each group member.

Excel is a microsoft software that allows you to create spreadsheets for storing data or formulating values. This can be used by the group to create displays for each data type whilst working on the subsystems adhering to banking and vendor interfaces. Data displays are crucial and a part of our design as of currently.

Mouse is a piece of hardware which is not necessary, however it may greatly aid in the creation of a proper prototype due to the advantages it may provide. This hardware is simply for the quality of life aspect for the group members to explore if they wish.

Wix is a tool used to create websites. It helps the group to create the prototypes of websites for the banks to use internally. It has several functions which are useful to create an ideal website.

Canva is a graphic design platform, used to create social media graphics, presentations, posters, documents and other visual content. It helps the group to create the prototype's interface and presentations.

Lucid is an online software that allows the user to create descriptive and detailed flow charts displaying a numerous variety of different information. Lucid will be used to create one of the flowcharts that will display the efficiency aspect of the program and user spending for the submission of the first prototype.

The links to each of the listed materials in the BOM include:

- ❖ Powerapps
 - <https://powerapps.microsoft.com/en-us> (Free Through University)
- ❖ Javafx
 - <https://openjfx.io> (Free of Charge)
- ❖ Figma
 - <https://www.figma.com> (Free of Charge)
- ❖ Moqups
 - <https://moqups.com> (Free of Charge)
- ❖ PC/Mac
 - (Previously Owned Product)
- ❖ Excel
 - <https://www.microsoft.com/en-us/microsoft-365/excel> (Free Through University)
- ❖ Mouse
 - (Previously Owned Product)
- ❖ Wix
 - <https://www.wix.com> (Free of Charge)
- ❖ Canva
 - <https://www.canva.com> (Free of Charge)
- ❖ Lucid
 - <https://lucidchart.com> (Free Signup)

Project Plan

Tasks Breakdown

Table 2: Tasks Breakdown

Prototype	Tasks	Time Estimate	Start Date	Target Date	Dependencies	Owner
Prototype I	Divide the concepts equally	1 hour		Oct 24	N/A	All
	Learn more about Powerapps and Javafx	2 days		Oct 27	N/A	All

	Integrate ideas for subsystem 1: User Interface	1 week		Nov 3	Learn more about Powerapps and Javafx	Anissa Feryza
	Integrate ideas for subsystem 2: Point Democratization	1 week		Nov 3	Learn more about Powerapps and Javafx	Kat
	Integrate ideas for subsystem 3: Point Use Efficiency	1 week		Nov 3	Learn more about Powerapps and Javafx	Paul Dawood
	Customer Feedback and questions for Prototype I	3 hours		Nov 5	Integrate ideas for subsystem 1, 2, and 3 : User Interfaces	All
Prototype II	Use Feedback to update Prototype I	2-3 hours	Nov 3	Nov 5	Customer Feedback and questions for Prototype I	All
	Adding Functionality to Prototype I	5 days	Nov 5	Nov 10	Use Feedback to update Prototype I	All*
	Update detailed design and BOM	5 minutes	Nov 11	Nov 11	Customer Feedback and questions for Prototype I	Dawood Paul
	Prototype II Finalization	1-2 hours	Nov 10	Nov 11	Adding Functionality to Prototype I	All*
	Gather Feedback and Comments on Prototype II	3 hours	Nov 11	Nov 15	Prototype II Finalization	All
Prototype III	Use Feedback to update Prototype II	2-3 hours	Nov 15	Nov 20	Gather Feedback and Comments on Prototype II	All*
	Refining the functionality for Prototype II	4 days	Nov 20	Nov 24	Use Feedback to update Prototype II	All*

	Getting users feedback on Prototype III	2 days	Nov 24	Nov 26	Refining the functionality for Prototype II	All
	Prototype III Finalization	1-2 hours	Nov 26	Nov 26	Getting users feedback on Prototype III	All*
	Improve the aesthetic of the design	1 day	Nov 27	Nov 28	Customer Feedback and questions for Prototype I	All*
	Integration of Prototypes into Presentation	1 hour	Nov 28	Nov 28	Prototype III Finalization	All*
	Presentation/Visual Aid Design/Aesthetic	2 hours	Nov 28	Nov 28	N/A	Kat Anissa Feryza
	Design Day Presentation Rehearsals	1 day	Nov 29	Nov 30	Presentation/Visual Aid Design/Aesthetic Integration of Prototypes into Presentation	All
Design Day - December 2nd						

Prototype II (Medium to High Fidelity test)

Prototype 2 will focus on adding functionality to our previously engineered design in prototype 1. With the respective changes and growth made in our project so far, it's now time to make our prototype bloom. Using the feedback we will add functions to each predetermined feature. We keep the three interfaces in the first prototype since it's functional enough. And we give conceptual algorithms to the point efficiency, to make it more specific and functional, and the testing analysis of it is done. The uncertainties and the risk of the algorithm are tested as well. We also test the code in the point democratization part to see if it works. Based on the changes we've made, there will be an update to the detailed design plan and drawing relative to the recently obtained feedback. After all of the functionality has been completed, the prototype will be finalized and prepared for another evaluation from our clients.

Prototype III (High fidelity fully functioning prototype)

For prototype 3, using the newly obtained feedback, we will analyze the comments we have received and yet again add or modify some features in our program in respect to our clients' wants and needs. A series of updates will be performed on our prototype to make sure it is fit to be our final product. After the modifications, a process of refinement will be conducted in order to ensure that the functionality previously integrated in prototype 2's construction is working and is easy to use for the entirety of our target audience. Next, the refined prototype will be used to interview and obtain feedback from sources other than our clients. This feedback will then serve as another source of ideas for further refinement and finalization. Once the product's functionality and criteria has been met, the aesthetic aspect of the prototype will be addressed and modified for viewing pleasure and to act as a sort of hook for the target audiences. All of the prototypes' features are compiled into this final product, so to present this as our grand final outcome, this prototype will be accompanied by a presentation slide show allowing us to explain our product in a fluid and proper fashion. Once that has been done and rehearsed the final prototype is ready for presentation at design day.

Conclusion

A prototype is a model used to test the functionality of a design before moving forward. It is a critical component that will ensure our client's success at the end of the project. The quality of the specified requirements provided to clients is improved through experimentation. Our client will be able to see and engage with a working model of the project through the prototype, providing comments, requesting project adjustments, and changing model specifications. It has been instrumental in removing flaws from the design. Finally, the prototype assisted us in determining early on what the client requires and what we need to change in our design in order to increase efficiency.