|  |
| --- |
| Student Name: Obada Alhalaybeh  Student ID: 23110107  Section #: 3 |



Prototyping – RQF 2022

Assignment 1

Spring 2023-2024

Contents

[**1-** **Different Forms of prototyping** 2](#_Toc136776873)

[1.1 Discuss the different forms of prototyping, their purpose, their advantages and disadvantages, their testing outcomes. 2](#_Toc136776874)

[1.2 Review the Standard Prototyping tools by providing a brief explanation about each tool and how they can be used in identifying and testing user requirements effectively. 2](#_Toc136776875)

[**2-** **Review different end user categorizations, classifications and behavior modelling techniques.** 3](#_Toc136776876)

[**3-** **Examine available prototyping methodologies to test with a specific end user from the user population.** 3](#_Toc136776877)

[**4-** **Prototyping Plan** 3](#_Toc136776878)

[4.1 Generic Description of Your Product Idea That Is Being Developed. 3](#_Toc136776879)

[4.2 Identify your product end users by choosing the suitable end user categorizations, classifications (Segmentation, persona canvas, The value proposition canvas, customer journey map….) and behavior modelling techniques (BJ Fogg’s Behavior model), **choose the suitable ones not necessarily all of them.** 3](#_Toc136776880)

[4.3 Determine the suitable methodology (Rapid throwaway prototype Evolutionary prototype Incremental prototype Extreme prototype), **choose the suitable one.** 3](#_Toc136776881)

[4.4 Outline the right tools to prototype your idea (example, JustInMind, Figma, Mockups…) 4](#_Toc136776882)

[4.5 Plan your prototype evaluation methods (Testing methods: Usability Testing, Focus Groups, beta Testing, A/B Testing, Surveys), **choose the suitable ones not necessarily all of them.** 4](#_Toc136776883)

[**5-** **evaluate the impact of common prototyping methodology within the software development lifecycle.** 4](#_Toc136776884)

[**6-** **Low-fidelity Wireframe** 4](#_Toc136776885)

[6.1 Wireframes screenshots before applying feedback. 4](#_Toc136776886)

[6.2 Iteration 1 4](#_Toc136776887)

[6.2.1 Feedback – outline the end users’ feedback on the first iteration of your wireframe. 4](#_Toc136776888)

[6.3 iteration 2 5](#_Toc136776889)

[6.3.1 Feedback – outline the end users’ feedback on the second iteration of your wireframe. 5](#_Toc136776890)

[6.4 Screenshots after applying feedback (of the updated wireframes only) 5](#_Toc136776891)

[**7-** **Mid-fidelity Mock-up** 5](#_Toc136776892)

[7.1 Mockups screenshots before applying feedback. 5](#_Toc136776893)

[7.2 Iteration 1 5](#_Toc136776894)

[7.2.1 Feedback - outline the end users’ feedback on the first iteration of your Mockup. 5](#_Toc136776895)

[7.3 iteration 2 5](#_Toc136776896)

[7.3.1 Feedback- - outline the end users’ feedback on the second iteration of your Mockup. 5](#_Toc136776897)

[7.4 Screenshots after applying feedback (of the updated mockups only). 5](#_Toc136776898)

[**8-** **High-fidelity Prototype** 6](#_Toc136776899)

[8.1 Outline the added features to build your high-fidelity prototype. (What are the interactions, validations added to your prototype to make it similar to the final product) 6](#_Toc136776900)

[8.2 Review end-user feedback from multiple iterations of your prototype and justify the updates to the final prototype based on end user feedback and testing. 6](#_Toc136776901)

[**9-** **Critically evaluate the prototype against the original plan and how user feedback and testing was implemented.** 6](#_Toc136776902)

[**References** 6](#_Toc136776903)

# **Different Forms of prototyping**

## 1.1 Discuss the different forms of prototyping, their purpose, their advantages and disadvantages, their testing outcomes.

Prototypes can take four primary forms in software development, and their functions vary based on the stage and objective of the product design process.

**Low-fidelity:**

Low-fidelity prototypes are quick and easy visualizations of the product that typically take the shape of wireframes, and sketches, they seek to test fundamental user processes and functionality as well as investigate early concepts , Low-fidelity prototypes are perfect for early feedback because of their speed and cost-effectiveness, on the other hand the lack of details (both visual and elements) and realism dose restrict and limit the usefulness when testing the design aesthetics and interactions. In conclusion, for the low-fidelity prototypes the best use of it is to gather initial information.

**High-fidelity:**

Very realistic representations of the final product, high-fidelity prototypes include real UI elements, interactions, and content that are nearly final. These are useful as prototypes to validate the design with stakeholders as well as for conducting usability testing in-depth, although they provide a very realistic user- experience, the time and effort spent on building and maintaining can result in less and slower iterations. In conclusion, the high-fidelity prototypes can be close up to 90% of the original product as it may only miss one or two functionalities.

**Feasibility prototypes:**

Feasibility prototypes are primarily developed to test the technical possibility of a proposed feature or project, these prototypes are written out of fast implementing code, in order to check if a functionality is doable or not, like an algorithm or backend process. The benefit of feasibility prototypes being that they can help mitigate technical risk prior to large-scale development taking place. But these solutions don't offer User Interfaces in most cases, are unusable when it comes to Usability testing because they are not made for the end user but for the technical means. In conclusion this type of prototype is not aimed at the user feedback, as it focus on if the idea is achievable or not (with least amount of code written for example to be fast).

**Live-data prototypes:**

You can evaluate a design's functionality in real-world scenarios by mixing it with actual data and traffic using live-data prototypes, these prototypes are helpful for verifying features or workflows under practical conditions and for evaluating the feasibility of the product concept, accurate performance data and user behavior insights are among its advantages. However, they are technically challenging to make and are not suitable as final goods, They serve more as experimental tools to gather data and feedback in real time before they are fully developed. In conclusion this type is just to see if the idea will work with the live data from real word (for example some beta testers).

## 1.2 Review the Standard Prototyping tools by providing a brief explanation about each tool and how they can be used in identifying and testing user requirements effectively.

There are several standard tools available in the industry that support effective prototyping, each offering unique features to help designers and developers identify and test user requirements.

**Figma**

One of the most popular programs for designing collaborative interfaces is Figma. Within a web-based platform, it facilitates interactive prototyping, wireframing, and UI design. It's great for team projects and gathering early input on workflows and visual hierarchy because of its real-time collaboration features.

**InVision & Balsamiq**

InVision focuses on transforming static designs into interactive prototypes. This is able to collect feedback and versioning, which is a perfect solution for presentations to stakeholders or iterative testing of medium-high fidelity prototypes. Balsamiq, however, specializes in low-fidelity wireframing. With easy drag & drop interaction, designers can quickly apply interface point of views, where it comes especially handy during brainstorming or early stage of a concept validation.

**JustInMind**

JustInMind is a complete tool that facilitates high-fidelity and low-fidelity prototyping, it enables the development of advanced interactions, responsive layouts, and form simulations, when extensive simulations are required to test user flows and design logic, this tool is especially helpful.

**Axure RP**

Axure RP is unique in that it allows complicated logic and conditional interaction, allowing designers to prototype interfaces with a lot of logic and dynamic content, it is most effective when used to test more complicated user needs and to confirm how the system responds to user activities.

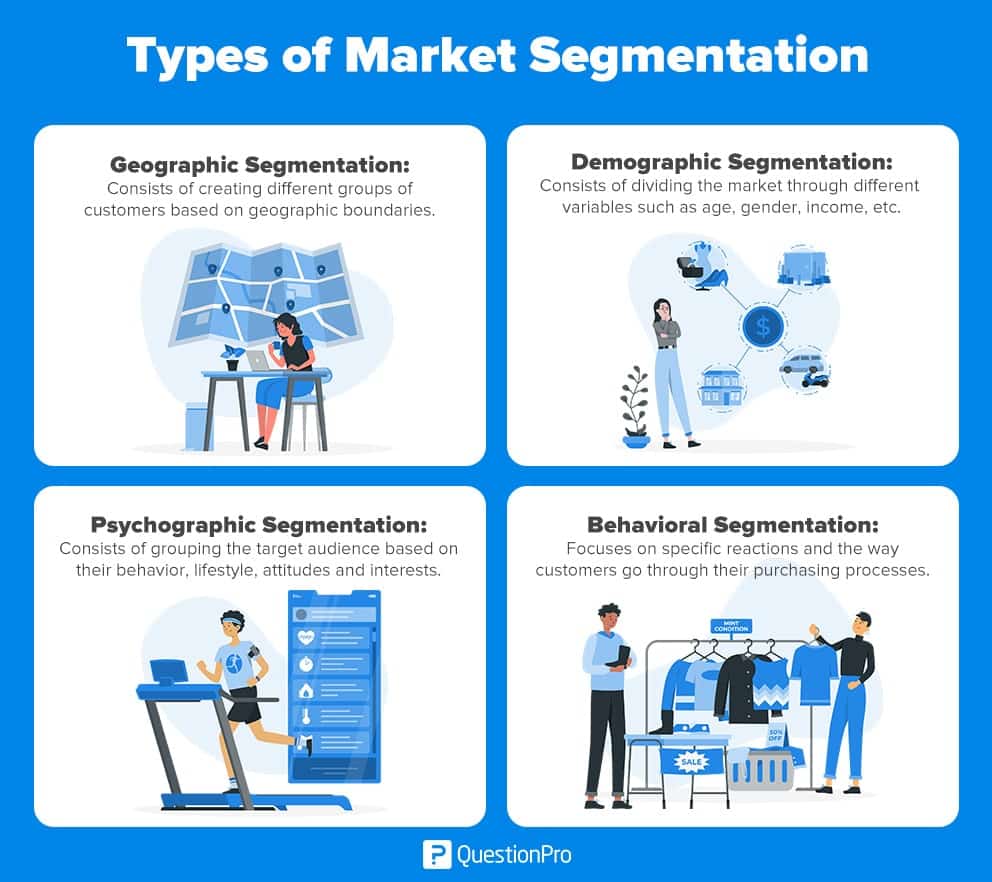
**To sum up, the goals of the project, the level of fidelity needed, and the stage of development all influence which of these tools is best for the prototyping process.**

# **Review different end user categorizations, classifications and behavior modelling techniques.**

Since a product's usability and efficacy for the final user determine its success, end-user analysis is a crucial step in the prototyping process. The individuals or the groups that uses the finish product are known as the end users, as a customer could buy the product but he will offer it to be used by the people who is serving so they are called the end users. And in prototyping the end user is very important role since it give feedback to the designers to improve the app.

We need some tactics to be able to serve the end users with better designs like:

* **End user segmentation:**  
  it is the process of dividing users into distinct groups that groups share similar characteristics or needs, this segmentation make it easier for the designer to create more personalized solutions for each group and this will help cover more variety of people since you can group a wide range of the people with it. We different forms of segmentation those segmentations are not the same for each company so you need to be carful what you use from them, here are some of them: Demographic Segmentation, Geographic Segmentation, Lifestyle Segmentation, Behavioral Segmentation, Business Customer Segmentation

  
(1)

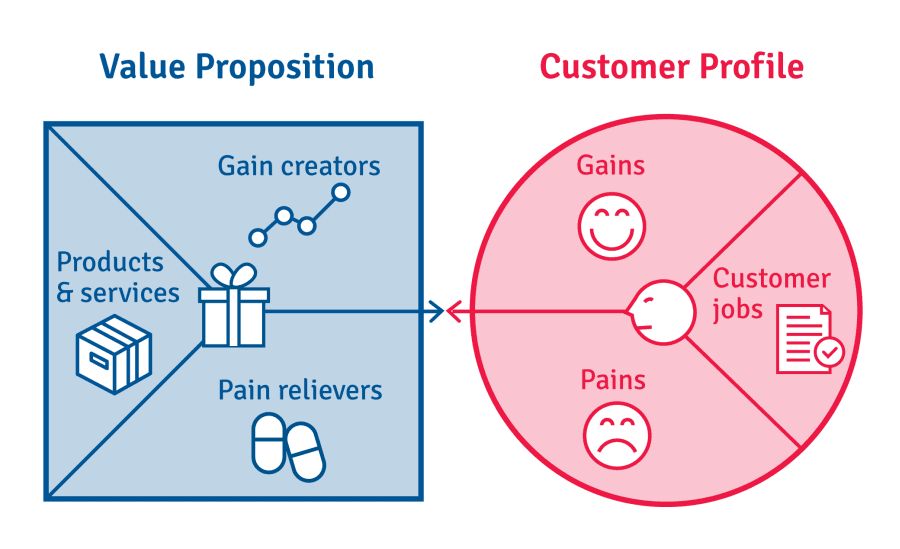
* **Persona canvas:**

Designers use the persona canvas to detail the segmentations. A persona is a fictitious character which describes a set of end-users. There is also very specific information about each persona, such as name, job title, personal and professional goals, fears, pain points, and dreams. The persona canvas enables design teams to visualize their end users, their behavior, and how they will likely interact with the product. It also promotes team empathy and improved decision-making based on actual user needs rather than conjecture.



(2)

* value proposition canvas:

Another powerful tool to match the design of the product to user expectations, is the value proposition canvas. It comprises two parts: the customer profile (covering customer jobs, pains and gains) and the value map (products/services, pain relievers and gain creators). Customer jobs are what the user is trying to achieve, pains are obstacles and frustrations that hinders achieving their goals, gains are those benefits they anticipate.” The value map explains how your product alleviates pains and allows the customer to experience their gains. This mapping ensures that the product is set up to solve a real customer problem and delivers the maximum value for the customer.

(3)

Now for behavior modeling techniques:

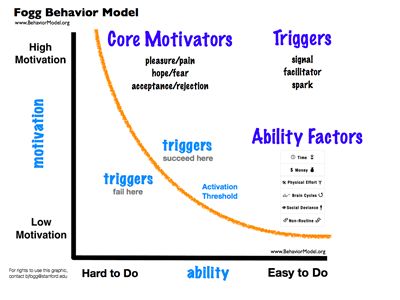
Behavior modelling techniques are methodologies that allow understanding and predicting of user's behavior while interacting with products or services. These techniques are about track and analyze users’ actions and decision-making to enable designers to deliver a more effective and user-friendly experience.

One common model is BJ Fogg’s Behavior Model, which explains that for a behavior to occur, Motivation, Ability, and Triggers must converge at the same moment.

* **Motivation** is the internal force that drives a user to act. It is subject to physical forces (for the sake of pleasure or to avoid pain), emotional states (hope or fear) and social factors (membership in a group).
* **Ability** explains the degree of ease or difficulty the user faces to perform the behavior, factors affecting ability include the time required, effort needed, cost, compatibility with routine, social norms, and mental load.
* **Triggers** can be external prompts or cues that start the behavior. These can be:
  + Sparks: increase motivation,
  + Facilitators: make the behavior easier,
  + Signals: nudge users to do something.

The desired behavior won't happen if any of these components are absent. By guaranteeing that there are suitable triggers to direct user behavior, that the necessary actions are simple to complete, and that motivation is high, this model assists designers in optimizing user interfaces.

Segmenting users according to their behavior and forecasting future actions like conversion, churn, or reactivation are further aspects of behavioral modeling. This enhances user engagement and retention by enabling tailored experiences and focused marketing tactics.



(4)

# **Examine available prototyping methodologies to test with a specific end user from the user population.**

Methodologies for prototyping offer organized ways to include end users in the creation of new products. The project's nature, the requirements' clarity, and the target user population's characteristics all influence the methodology choice. With the help of these approaches, teams can get insightful input, improve designs, and make sure the finished product successfully satisfies user needs.

* Rapid Throwaway Prototyping.

The goal of rapid throwaway prototyping is to quickly build simple, basic prototypes in order to test concepts and get early feedback. Once their function is fulfilled, these prototypes are usually thrown away. This approach works well for projects where end users' expectations need to be clarified quickly, and requirements are vague or changing quickly. By addressing problems early in the development lifecycle, it enables teams to avoid expensive rework, and its worthy to note that the strength of this type is the interfaces that is created since the user can see the system and that gives him a good grasp on the functionality of the system and how it works.

* **Evolutionary Prototyping**

the main part of the evolutionary prototype that it is a functional software not a simulation, it requires an incremental modifications at every feedback, this prototyping method help in saving time since every modification will be built on the previous one , also it thrive in situations where the requirements are not clear or when the functionality needs to be at constant checking state, the journey here starts with product that has only the clear system requirements then new features can be added along the way when they are understood and clear. So in a simple way on the Evolutionary Prototyping the system will be under constant modification.

* **Incremental Prototyping**

The concept of this type is to split the software into smaller fragments and work on them all at the same time this will help reduce the time needed to meet the deadlines, but it has a risk, this risk is represented as the difference between each prototype, the solution is to have rules and guidelines in advance so everyone know what to expect.

* **Extreme Prototyping**

This prototyping method is used in web development and it consists of 3 phases, first phase start with creating a static web and it is mainly from HTML, the second phase the pages are made to be functional, on the last phase the services are implemented.

# **Prototyping Plan**

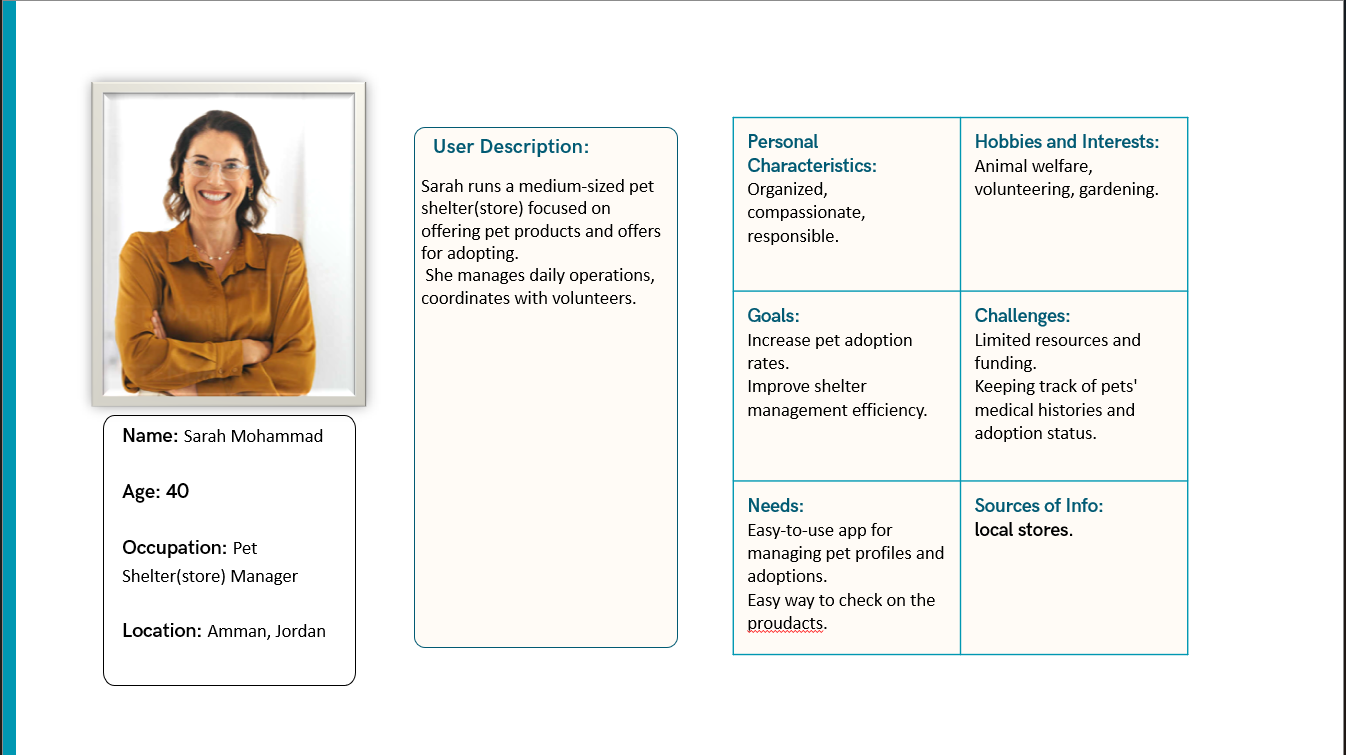
## Generic Description of Your Product Idea That Is Being Developed.

I am making an application simulating a pet shelter where there will be pets to adopt like dogs and cats. Also, there will be a part for buying various pet products and accessories. The aim of this application is to encourage the users to adopt pets and this could be achieved through good design and user friendly interface.

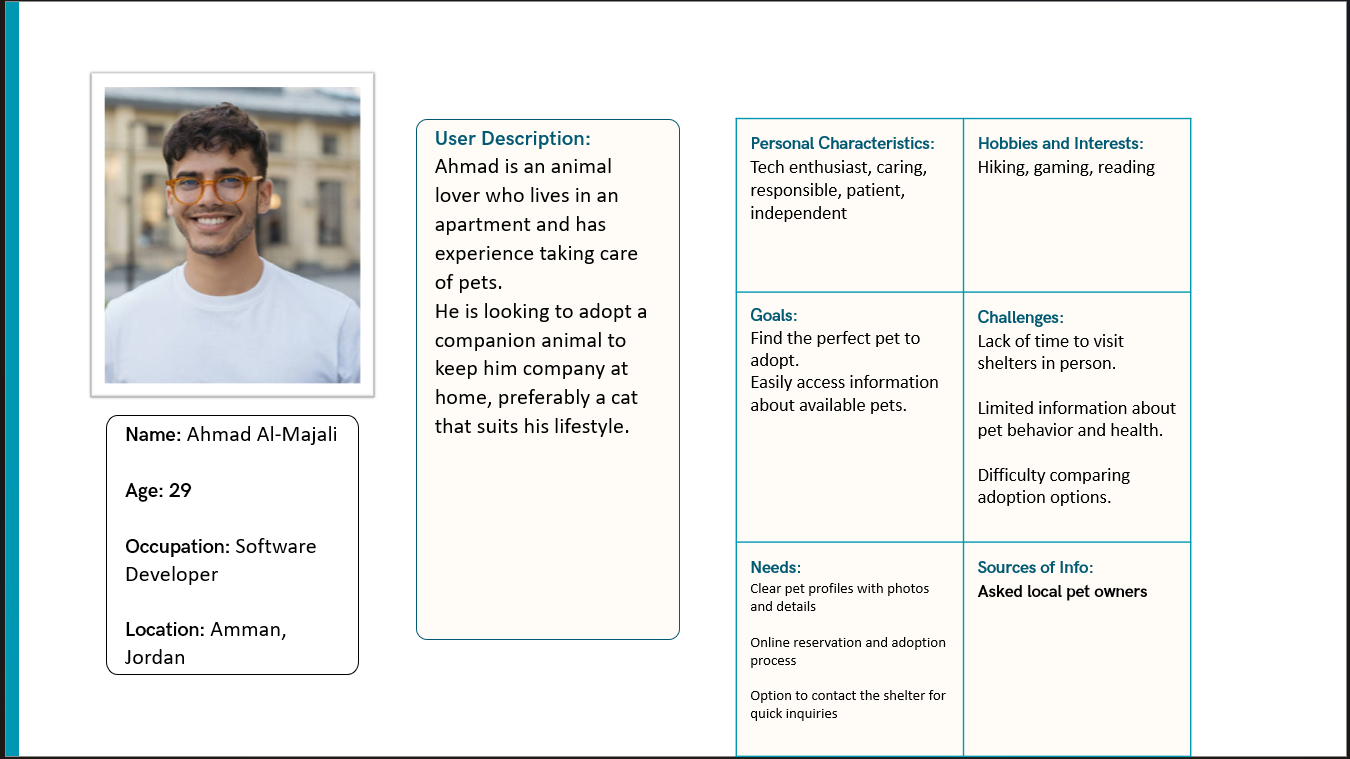
## Identify your product end users by choosing the suitable end user categorizations, classifications (Segmentation, persona canvas, The value proposition canvas, customer journey map….) and behavior modelling techniques (BJ Fogg’s Behavior model), **choose the suitable ones not necessarily all of them. Minimum three (Persona, segmentation, value proposition)**

**Persona canvas**

**Pet Shelter(store) Manager**



**Pet Adopter**



**New Pet Adopter**

A person with a picture of her

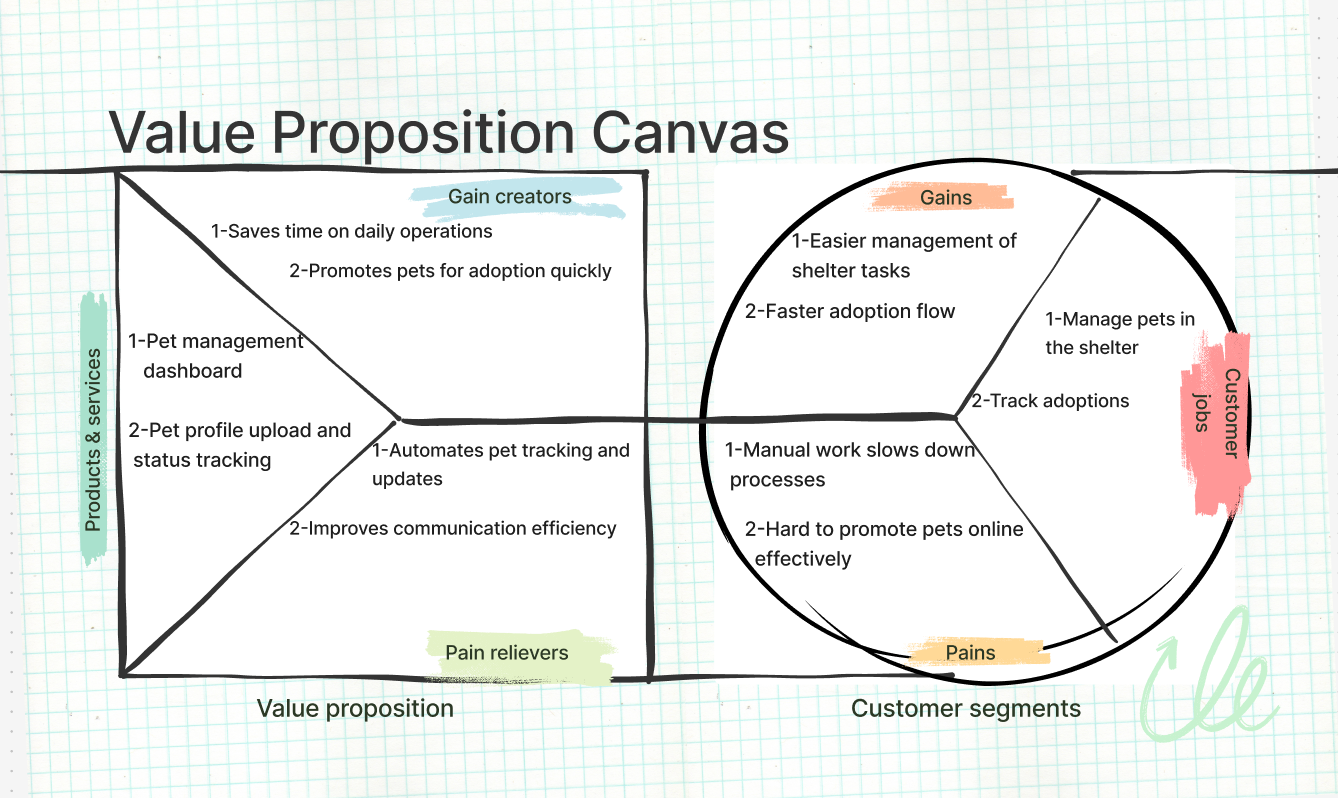
AI-generated content may be incorrect.

**Segmentation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Segment name | User type | Demographic traits | Behavioral traits | Lifestyle traits |
| Owners | Pet shelter(store) owners | M/F, age 35-50, store owner, bachelor’s degree | Manage pet shelter(store) operations, uploads pet profiles, tracks adoptions | Structured, community focused |
| **Experienced Adopters** | Pet Adopter | M/F, age 25-35, professional | Has adopted before, seeks specific pets | Busy schedules, value companionship and activity |
| Newbie adopters | New Pet Adopter | M/F, age 18-25,  Student | Needs guidance, emotionally driven | Learning-focused, emotionally open, seeking emotional support from pets |

**Value Proposition and value map**

**Pet Shelter Owner**



The value: A smart system that helps shelter owners manage pets, keep records, saving time and boosting adoptions.

**Experienced Pet Adopter**

A diagram of a diagram

AI-generated content may be incorrect.

The value: A modern adoption app with search filters, pet profiles, and one-click application made for confident adopters who know what they want.

**New Pet Adopter**

A diagram of a diagram

AI-generated content may be incorrect.

The value: platform that guides first-time adopters with learning tools, and pet care resources making the journey stress-free.

**BJ Fogg’s Behavior Model**

First of all the formula is:

Behavior (B) = Motivation(M) X Ability(A) X Trigger(T)

now when we apply it to each segment we have(owner, newbie, experinced)

* Pet Shelter Owner
  + **Motivation:** Wants to improve adoption rates and shelter efficiency
  + **Ability:** capable of managing data (high ability)
  + **Trigger:** Notification to update a pet’s status, add a new pet, or respond to an adopter
* Experienced Pet Adopter
  + **Motivation:** Strong desire to find the right pet quickly
  + **Ability:** Tech-enthusiast, able to use filters, forms, and search tools (high ability)
  + **Trigger:** Alert for a new pet that matches their preferences
* New Pet Adopter
  + **Motivation:**  
    Emotionally driven, eager but nervous about adopting
  + **Ability:**  
    limited knowledge of adoption procedures or pet care, but proficient in technology and willing to learn. Their skills rapidly improve with instruction. (low ability at the start)
  + **Prompt:**  
    Friendly messages like “Need help choosing your first pet?” or “Start your pet care journey here”

## Determine the suitable methodology (Rapid throwaway prototype Evolutionary prototype Incremental prototype Extreme prototype), **choose the suitable one.**

for this project will choose the evolutionary prototype method, since the application I am trying to design has multiple user roles like: shelter(store) owners, experienced adopters, and first-time adopters, it is crucial to gather early feedback and refine the prototype over time, over time this will generate a user-friendly application since the evolution will be based on the users liking or not.

## Outline the right tools to prototype your idea (example, JustInMind, Figma, Mockups…)

After taking a look at all the available tools I have chosen Figma to create all of the design from the wire frame to the final prototype, the reasoning behind me choosing Figma over the others was the features it offers like the real-time collaboration, the ease of the file sharing system since I use both laptop and desktop it help a lot, and we do not forget the free plans that it offers and the free premium for the students.

## Plan your prototype evaluation methods (Testing methods: Usability Testing, Focus Groups, beta Testing, A/B Testing, Surveys), **choose the suitable ones not necessarily all of them. Minimum three**

* **Usability testing**

I will get the feedback through a the moderated testing as I will give the prototype to the shelter(store) owner and collect the data from him as if he could add pet to the adoption lest, and I will give him guidance through out the testing.

For the another side of the app were the users will try and adopt and I will be taking the feedback of it.

After both are done I will go on into the unmoderated testing to see if other users face any major problems in the testing period.

* **Surveys  
  Surveys will be used to gather data from a larger group of users after the prototype reaches a more functional state. Participants will be asked to provide feedback on various aspects of the app, such as usability, ease of navigation, and overall satisfaction. This method allows for the collection of both quantitative and qualitative data, offering valuable insights into user preferences and pain points. The surveys will be distributed via online forms, ensuring a wide reach and enabling the collection of a broad range of feedback.**
* A/B testing

The concept of the A/B is to create 2 variants of the UI elements and compare them to see which one preform better.

This testing method will give the user the choice of finding the best and comfortable design for him for example how the layout for the pet cards will be.

# **evaluate the impact of common prototyping methodology within the software development lifecycle.**

When applied to conventional SDLC models such as Waterfall, Agile, Iterative, and RAD, the four prototyping approaches—rapid throwaway, evolutionary, incremental, and extreme—play distinct roles and have differing degrees of impact. Here are the effects of each prototyping technique on these models.

* Waterfall

Waterfall is a linear and sequential process where one step (requirements, design, development, test, deploy) builds sequentially upon the previous step. It doesn't inherently support iterations or prototyping as a part of its basic design.

Prototyping Impact

Rapid Throwaway Prototyping: Applied in the requirements stage to create vague specs. But it can't influence subsequent stages because Waterfall presumes each one of them will be completed prior to the next step.

Evolutionary Prototyping: Unsuited for Waterfall because the strict Waterfall model will never allow for iterative change after the design step has been finalized.

Incremental Prototyping: Waterfall cannot take advantage of incremental prototyping since it doesn't support development in parallel. But the parts of the system may be developed by prototyping and testing in increments if needed.

Extreme Prototyping: May be applicable during the design stage to construct functional web-based interfaces but there will be difficulty in having them incorporated into the inflexible system of the Waterfall model.

Ideal for: Waterfall works best where specs are to be clear and will remain unchanged throughout development. Prototyping is one ambiguity resolution for ambiguous specs but with limited scope.

* Agile  
    
  Agile emphasizes continuous development, incremental improvement, and user feedback. Agile leaves space for adaptability and constant change and hence is very compatible with prototype approaches.
* Prototyping effect

Rapid Throwaway Prototyping: Suitable for early development to develop basic product iterations in a short time and provide immediate feedback and iterate on features to establish improved user stories.

Evolutionary Prototyping: This is a key part of Agile. It accepts constant improvement in the guise of iterative cycles so the features can evolve with continuous user input. Agile models like Scrum and XP thrive at this practice because it goes hand-in-hand with these approaches' focus on creating evolving functionality.

Incremental Prototyping: Promotes Agile's principle of breaking a product into smaller functional modules. Each module or increment can be developed and tested independently in an effort to reduce the time to market.

Extreme Prototyping: Of great importance for Agile but greatly suited for web development. Allows very rapid front-end interface prototyping and immediate end-user or stakeholder feedback, fitting nicely with Agile's short sprint cycles.

Best used with: Agile methods work best with evolutionary and incremental prototyping since these two techniques also emphasize iterative development and continuous user feedback.

* **Iterative**

The Iterative Model develops software in small iterative cycles or iterations. It adds and refines functionality with each iteration, incorporating user feedback in each cycle. The model is similar to Agile but less concerned with early delivery.

Prototyping impact:

Rapid Throwaway Prototyping: May be used during requirements gathering in order to establish key features and functionality prior to starting the full iteration. Not typically used in subsequent iterations.

Evolutionary Prototyping: Suitable for the Iterative Model. Each cycle makes the product better, with feedback at the conclusion of each cycle. Teams are able to create the product according to continuous user feedback.

Incremental Prototyping: The system is divided into parts or modules, all of which are created simultaneously but are tested individually, allowing for faster feedback and development for specific parts of the system. The Iterative Model truly takes advantage of this, as it can develop both the design and the functionality over successive iterations. Extreme Prototyping: Best suited for Iterative models, particularly in the case of web applications. It can quickly test UI as well as functionality prior to integrating backend systems. Feedback from such prototypes can be utilized to influence later iterations.

Best for: Iterative Model is best suited for incremental prototyping and evolutionary prototyping since it emphasizes iterative delivery of functionalities and ongoing improvement.

* RAD  
    
  RAD is an incremental process model with a strong focus on quick development cycles, typically weeks or months. One of the fundamental principles of RAD is prototyping, which allows developers to produce rapid prototypes and refine them based on user feedback in real time.

Prototyping Impact

Rapid Throwaway Prototyping: Serves a central role for RAD as it enables groups to create prototypes quickly and get user feedback. The prototypes are used to establish the functional and non-functional requirements at the beginning of the project.

Evolutionary Prototyping: This fits very nicely with RAD's need for continuous iteration. The ongoing input from users continually refines the prototype, slowly evolving into the end product.

Incremental Prototyping: RAD breaks down the product into tiny functional modules and each module can be developed, prototyped, and tested separately. This reduces development time by a significant extent and allows for rapid workable version releases.

Extreme Prototyping: Most suitable for RAD, and very suitable for web applications. Facilitates the development of complex UI and integration with the back end and tests the system within real environments.

Most appropriate for: RAD heavily depends on throwaway and evolutionary prototyping because they support rapid development cycles and continuous improvement.

# **Low-fidelity Wireframe**

## 6.1 Wireframes screenshots before applying feedback.

A screenshot of a login form

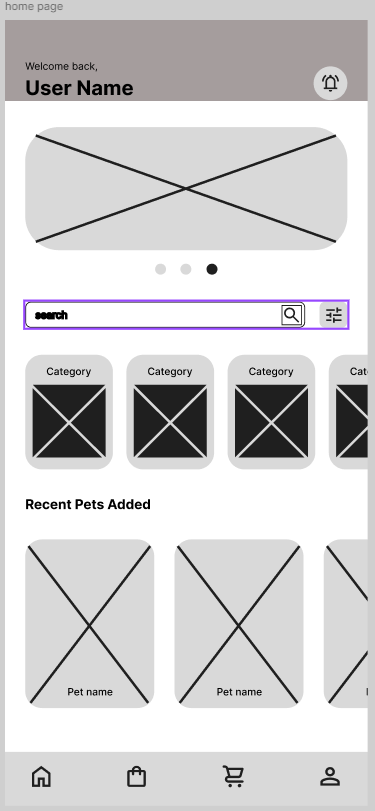
AI-generated content may be incorrect.

A screenshot of a login form

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.



A screenshot of a phone

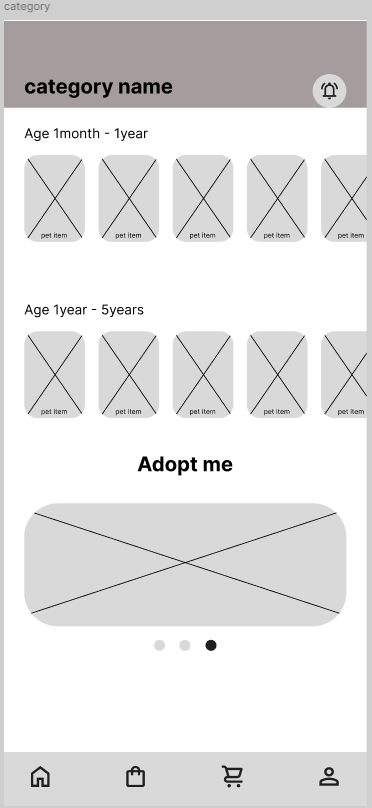
AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a web page

AI-generated content may be incorrect.



A screenshot of a computer screen

AI-generated content may be incorrect.

## 6.2 Iteration 1

### 6.2.1 Feedback – outline the end users’ feedback on the first iteration of your wireframe.

The first feedback I got was for the navigation bar where it did not show where is the user now so I added the effect of hover above and filled icon for the 4 main pages.

And I added a default navigation bar for when the user is not on one of the four pages.

## 6.3 iteration 2

### 6.3.1 Feedback – outline the end users’ feedback on the second iteration of your wireframe.

The feedback on this iteration was aimed at one page the category page it was about the lay out of the screen

Also the feedback gave me the idea of adding cancel button to the profile to revert any unwanted change it also has been changed to per page and item preview too.

## 6.4 Screenshots after applying feedback (of the updated wireframes only)

A screenshot of a login form

AI-generated content may be incorrect.

A screenshot of a login form

AI-generated content may be incorrect.

A screenshot of a cell phone

AI-generated content may be incorrect.

A screenshot of a mobile app

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

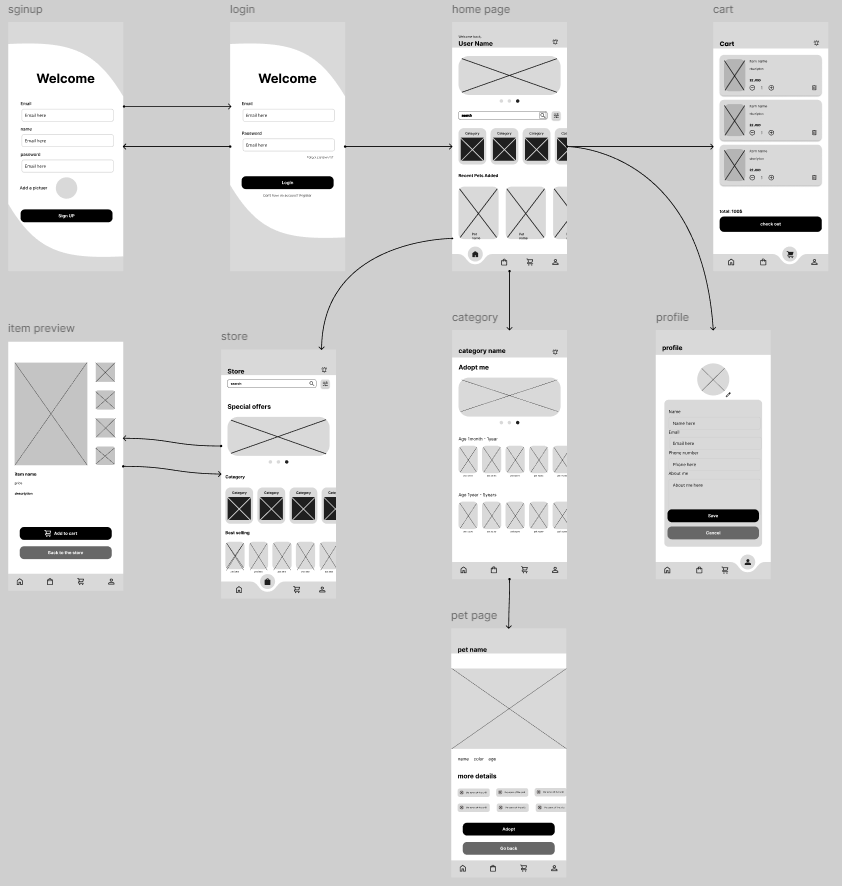
A screenshot of a cell phone

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

**This is an image of the flow**



# **Mid-fidelity Mock-up**

## 7.1 Mockups screenshots before applying feedback.

A screenshot of a login form

AI-generated content may be incorrect.

A screenshot of a login form

AI-generated content may be incorrect.

A screenshot of a cell phone

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a cell phone

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a cat diet

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

## 7.2 Iteration 1

### 7.2.1 Feedback - outline the end users’ feedback on the first iteration of your Mockup.

On the first iteration it was suggested that I add the prices under the items and the pet names too, so it make it much easier and clear for the users to know how much they will spend and what is the name of the pets without going into the pet specific page.

## 7.3 iteration 2

### 7.3.1 Feedback- - outline the end users’ feedback on the second iteration of your Mockup.

On the second iteration I got notes about the cards visibility issues with the back ground so I added shadows for it, and now it is more clear and appealing to the eye and can see the boundaries of the cards.

## 7.4 Screenshots after applying feedback (of the updated mockups only).

A screenshot of a login form

AI-generated content may be incorrect.

A screenshot of a login form

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a cell phone

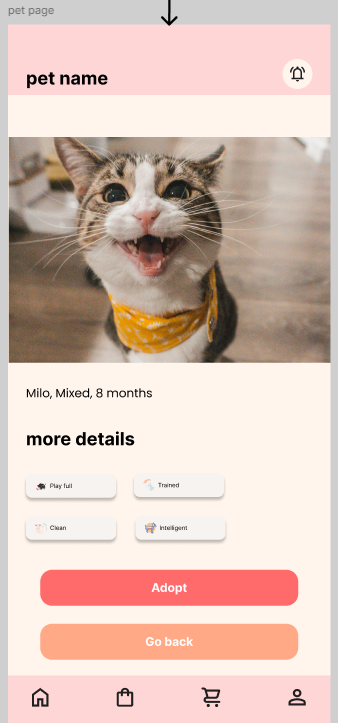
AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a cat food

AI-generated content may be incorrect.



**Overflow**



# **High-fidelity Prototype**

## 8.1 Outline the added features to build your high-fidelity prototype. (What are the interactions, validations added to your prototype to make it similar to the final product)

I started with the validation for the signup and login, I made it that only when all the fields are filled that the user can go on to the next page, for the signup the user needs to provide email, name password and image for the profile, for the login user needed to only provide the email and password to be able to go to the next page, I did something similar to the profile but I mad the condition if the user changed any of the fields he can save as its not required to change everything on the profile.

I also worked on the cart page to create an adder to make it easier to increase the quantity of the item or decrease it, I also added an interaction with the delete icon so it removes the item from the cart.

## 8.2 Review end-user feedback from multiple iterations of your prototype and justify the updates to the final prototype based on end user feedback and testing.

First thing I got from the feedback was about the button looking clickable but not responding since the user did not fill the fields so I added a condition that change the color of them until all fields are filled, I also added an overlay for some buttons to make it feel more real when clicked, for the cart I fixed a problem of the numbers going into 0 and negative that can’t be logically possible on a cart.

Over all after all the testing with the users I got to my final prototype that is shown in the video

# **Critically evaluate the prototype against the original plan and how user feedback and testing was implemented.**

The initial idea was to design an application focused on pet adoption only, but the user feedback I got suggested to me the addition of a side store to the concept, this addition will make it more convenient for the user to adopt and get the essential supplies for the pet, such as food and cleaning supplies, as this change happened as my prototyping method evolves around feedback threw out the designing period as I learn from the user what they want.

**Low fidelity**

I started creating the wireframe to outline the main features of the application such as home page, profile, pet profile, pet page and the navigation bar, the first iteration the feedback was over all about the clarity of the navigation bar since the users struggled to identify which page they are on, so I started gathering feedback with the A/B testing on 2 new styles for the navigation bar, and the one with the most liked view was selected to be present in the application, on the second iteration I fixed a problem with the UX after I got the feedback about how hard it is to cancel and go back and forth between the pages.

**Medium fidelity**

On this part of designing the mockup the focus went to the UI elements like the color picture name and price tags, on the first iteration of the feedback I got notes about the ability to show the price tags of the items in the store and the names of the pets, so I improved the visibility of them as this can be a QoL(quality of life) change so that the user is not forced to go into the item details to see the price or to go into the pet page to see the name, on the second iteration I got the feedback of the cards are blinding in with the background color so I created a shadow for each card so that it show the difference from the background as this count as improving the UX.

**High fidelity**

In this part I moved from simple prototype to a more advanced mechanics that mimic a real program like validation on the login and signup also on the profile page too, and I added a notification that pop up when the adopt button is pressed, some of the feedback I got about the buttons not navigation the user to the next page like the login since it is required to add email and password so I changed the color of the button to a darker color until the fields are filled this improvement helped with the UX and reduced the confusion, I created overlays too to emulate a real app when clicking buttons that need back end job, and I improved the cart to be almost fully functional on a prototype level.

# **References**

Villegas, F. (2021). *Market Segmentation: What it is, Types & Examples | QuestionPro*. [online] QuestionPro. Available at: <https://www.questionpro.com/blog/market-segmentation/>. (1)

Business Design Tools. (2023). *Persona Canvas - Business Design Tools*. [online] Available at: https://businessdesigntools.com/the-tools/persona-canvas/.

‌ Canva. (2021). *Free and customizable user persona templates*. [online] Available at: <https://www.canva.com/templates/s/user-persona/>. (2)

‌ B2B International (2025). *What Is the Value Proposition Canvas?* [online] B2B International. Available at: <https://www.b2binternational.com/research/methods/faq/what-is-the-value-proposition-canvas/>. (3)

‌ Mulray, M. (2015). *Change Behaviour Quotes*. [online] Pinterest. Available at: https://www.pinterest.com/pin/246009198373417915/ [Accessed 29 May 2025]. (4)

GeeksforGeeks (2018). *Software Engineering | Prototyping Model - GeeksforGeeks*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/software-engineering-prototyping-model/>.

‌

‌

‌