Enhancing User Experience with web applications through Emotional Design

Julia Vister

Supervisor:
prof. dr hab. Stanisław Goldstein

Submitted in fulfillment of the requirements for the degree of

Bachelor’s in Computer Science with specialization in English

Faculty of Mathematics and Computer Science

University of Lodz

2024

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1: INTRODUCTION TO EMOTIONAL DESIGN AND USER ENGAGEMENT

The user experience of modern web applications is very important, perhaps even more important than the basic functionality. It is not about getting things done, but about making sure users enjoy doing it. This is where emotional design comes in – making users feel good while using an application. This means creating interfaces that don't just check the boxes for functionality but also make users feel positive. It's about forming a stronger connection between the user and the app, surpassing the purely practical aspects.

Keeping users interested involves more than just making sure that their needs or wants are satisfied through the app. The users must be engaged while satisfying their needs as well. When users are engaged, they tend to try out different features, use the app habitually and stick with it. Emotional design plays a big role in making this happen by creating a meaningful experience for the user. Figuring out how emotions and engagement work together is key to making web applications that both work well and make a lasting impression.

In the context of this thesis, the principles of emotional design and user engagement will be applied to the realm of expense tracking. The expense tracker application, while primarily serving a purpose, offers an opportunity to elevate user experience through emotional design. By understanding users' emotional responses to financial management and tailoring the applications design accordingly, we aim to create an engaging, user-friendly, and ultimately more effective expense tracker.

Budget Buddy is a full stack web application involving a robust technology stack, chosen to meet the project's requirements for efficiency, scalability, and maintainability. The application is implemented in programming languages made for both backend, front-end and database development.

CHAPTER 2: TECHNICAL OVERVIEW AND IMPLEMENTATION DETAILS

i. Programming environment

The application was written using Visual Studio Code (abbreviated to VS code). It's a free code editor that supports standard development operations like debugging, task running and version control. VS code uses Electron, which is an open-source software framework and is also the main GUI framework in VS code. Electron makes it possible to combine web technologies with the same flexibility and efficiency as native apps. The quick code-build-debug cycle of this tool enables the developer to focus only on the code. It also has multiple other services that make development comfortable, such as copilot and IntelliSense – a code completion feature. It also has customization and built-in support for various extensions and tools. VS code is also equipped with built-in support for Node.js that facilitates the development of a project using JavaScript. Another great reason for VS code is that it has tooling for the exact web technologies needed in this application, like HTML and CSS. (Visual Studio Code, 2024)

ii. Programming languages

1. Python

The Budget Buddy application is divided into frontend and backend, where the backend was written in Python. Python was used to ensure efficient data processing logic and seamless communication with both the frontend and database. Python was chosen as the backbone of the application because it has a clean and easily readable syntax. This contributes to the development of both workable and understandable code – mainly because it mimics natural language. The simplicity of Python accelerates the development process and may reduce the likelihood of errors.

Python is a great open-source language made to be used in a range of applications. The reason being it is a general-purpose language - this of course includes web development as well. Python features an extensive ecosystem made up of a multitude of modules, libraries, and frameworks (one of which is Django, used in this project). These include data manipulation, interactions with databases and creation of Application Programming Interfaces, making it an ideal choice for rapid development of an application. (Coursera Staff, 2023).

These are also the reasons why Python is one of the most favored and used programming languages in the world. According to a study conducted by Stack Overflow for their annual developer survey in 2023, Python ranks as the third most used programming language. To note is that Python is often a language of choice for people who are learning to code or those who are not professional developers. This highlights the arguments about it being simple and easy, making it an attractive choice for everyone and anyone. (Stack Overflow, 2023)

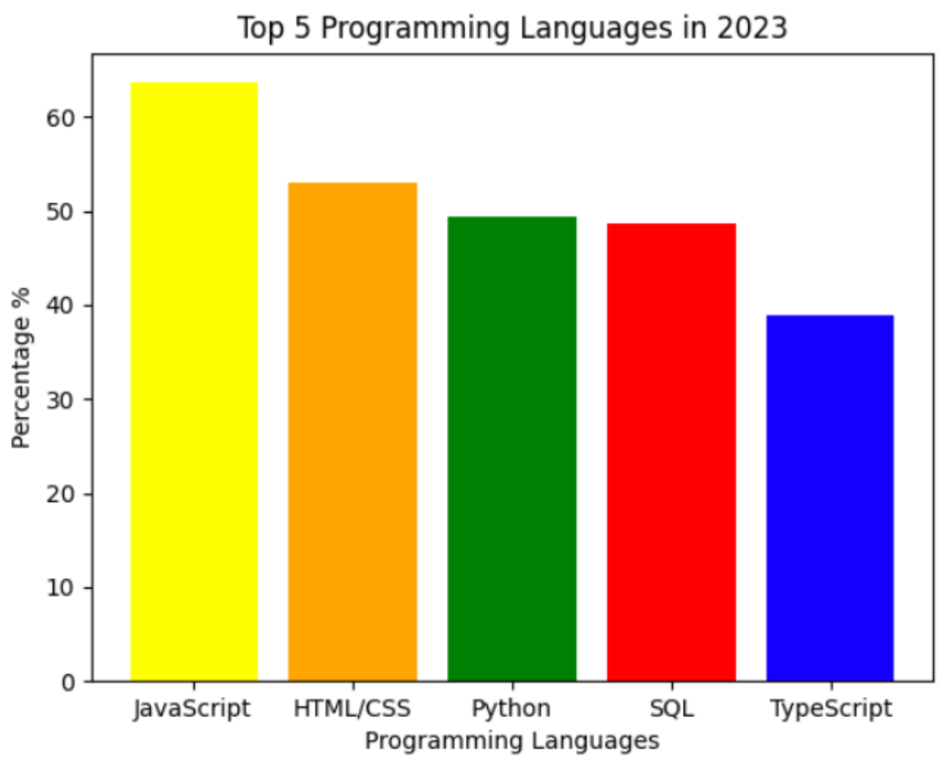


Figure 1: Graph showing the top 5 programming languages. Data taken from the 2023 Stack Overflow developer survey.

2. *HTML & CSS*

The frontend of the application is created using a combination of HTML, CSS, and JavaScript. Such a combination makes it possible to create a responsive, accessible, visually appealing, and interactive user interface. The most important part about the frontend is that it is a bridge between the design and the logic of the application. Therefore, making the website easy to understand, visually appealing and intuitive to use has a great impact on the adoption of the application. The images, body, buttons, navigation, and interaction are one of the things users see upon entering the website - this is all important to give it a "personality".

HTML, short for HyperText Markup Language, is a markup language for creating webpages. With this, we can create and structure sections, paragraphs, links, and other major parts of a webpage using HTML elements. These elements are the building blocks of any webpage, they include but are not limited to tags and attributes. You can think of the tags as categories for data, and with it the possibility to define and describe its purpose - what page elements are and where they should go. In short, they define a layout and a hierarchy which provides a logical structure to the user interface.

HTML serves a variety of purposes, though there are three distinct applications of this technology. Firstly, web development – a way of displaying web page elements within browsers, such as links, media, and text. The second one is internet navigation as HTML is heavily used for embedding hyperlinks and making it easy to navigate and include links in related pages and websites. Lastly there is web documentation, where HTML allows for the organization and formatting of content. Though, to be accurate HTML is not viewed officially as a programming language rather that it is a web standard maintained by World Wide Web Consortium. (S., 2023) For this thesis, I will therefore write about HTML being a “language” – with quotations mark, for simplicity’s sake.

HTML is the second most popular “language” used by both new and old developers, because it is supported by almost any computer, since it is viewable in a web browser. See figure 1. (Stack Overflow, 2023) Yet, it does not come without some disadvantages such as it’s mostly used for creating static websites – yet any dynamism may be programmed using JavaScript. Another disadvantage is regarding browser compatibility, where some browsers might adopt features more slowly than others, thus changing the way different users experience it. (S., 2023)

Whereas HTML is a descriptive “language” of the contents and structure of a document, the idea behind Cascading Style Sheets (CSS for short), is that it describes how the elements are presented. CSS is employed to style the HTML elements to ensure the application gets a consistent and visually pleasing design. CSS allows for a definition of color schemes, fonts and its text sizes, margins, paddings and how the various buttons, other clickables and tables look, to create a cohesive user interface. It uses formatting rules to style the HTML elements while also defining how an element is displayed on different media types (mobile, tablet or laptop), therefore linking HTML and CSS is vital for an engaging website. CSS is especially important for any application’s responsiveness, mainly adapting to various screen sizes. A webpage. Designed according to responsive design principles – guidelines for designers and developers considering web fluidity - ensures a seamless and optimal experience for both desktop and mobile, but not only for those. Some of those principles include appropriate typography, not hiding content and enhancing visual hierarchy. (Designveloper, 2023)

CSS’s consistency makes it possible for changes to one rule to be automatically applied universally to all pages. Another positive side of using CSS is that it allows for a separation between data organization and presentation. A single and separate style file may provide a clean and uniform look and feel to a website. This results in fewer lines of code which leads to a faster loading website. However, this style sheet can also be a bit confusing, especially for beginners. (A., 2024)

3. JavaScript

JavaScript is used in the project to make websites interactive. It's an easy programming language, that is also the most used language in 2023. See figure 1 for percentage of use. (Stack Overflow, 2023). There are many reasons why JavaScript is so popular. It can be used both for frontend and backend when creating a web application, compatible on all devices, an interpreter is bundled with every web browser, and has a wide range of frameworks and libraries free to use. (Simplilearn, 2023)

JavaScript is first and foremost used to add dynamics to web pages and applications. It's meant to add interactive elements to engage users. It complements both CSS and HTML. JavaScript is used to handle events (such as mouse clicks), which manages the users' interactions. Submissions, forms, and other actions the user performs on the application trigger JavaScript functions to do specific tasks.

AJAX, which stands for Asynchronous JavaScript and XML, is used for synchronous data retrieval and enabling dynamic updates without reloading the entire page – updating only certain parts. It uses a combination of JavaScript, HTML DOM, and a browser built-in XMLHttpsRequest object or Fetch API. In simpler terms, requesting data from web server and displaying it. (W3Schools, 2024)

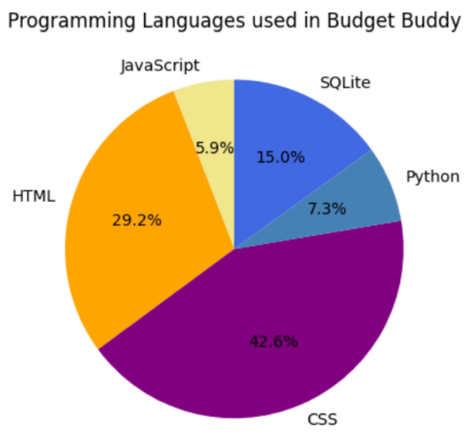


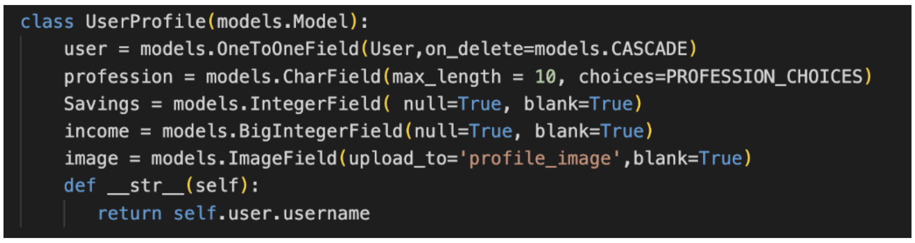
Figure 2: Pie chart showing percentage of programming languages used in the project. Data taken from the project's GitHub repository.

iii. Frameworks

1. Django

Django is used as the only backend - server-side web framework for the expense tracker application. It is high-level, free, and open source. It is meant for building strong and scalable web applications using Python. The principle of Don't Repeat Yourself (DRY) is advocated by Django, especially in component reuse. Django comes with many ready to use features, like database and CRUD operations, short for Create Reade Update Delete. Django follows a Model-View-Template (MVT) architectural pattern. (Buczyński, 2023) Model stands for the data to present taken from the database. View is a request handler that returns content, while template contains the web page layout and its logic. With this architecture, it creates a clean and organized code environment. (W3Schools, 2024)

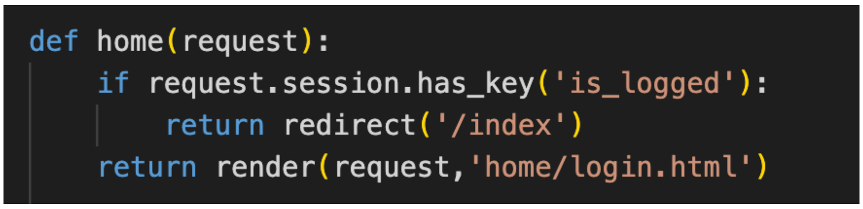
Django models define the web application's data structure and are often defined inside the models.py file. These models are created with Python classes, and are directly linked to a database table, where the fields define its columns. See figure 3 for an example. Django can simplify database operations and interactions using Object-Relational-Mapping (ORM), so that we may use Python code instead of SQL queries. Because of these models, Django also automatically creates a production ready administrative interface. The administrative interface is a webpage which makes it possible for developers to add, delete, or change objects. With this there is no need to create backend interfaces purely for editing and managing content. (Visual Studio Code, 2023)



```python
class UserProfile(models.Model):
    user = models.OneToOneField(User,on_delete=models.CASCADE)
    profession = models.CharField(max_length = 10, choices=PROFESSION_CHOICES)
    Savings = models.IntegerField( null=True, blank=True)
    income = models.BigIntegerField(null=True, blank=True)
    image = models.ImageField(upload_to='profile_image',blank=True)
    def __str__(self):
        return self.user.username
```

Figure 3: An example of a model written in Django. This model describes a user.

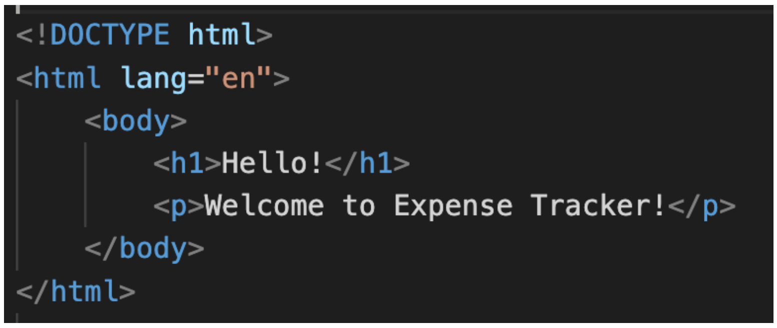
Django views is what handles the logic for processing user requests and returning the appropriate responses. Views are what handle models and templates, processing data and passing it to these templates. A view is essentially a Python function or a method that uses http requests as arguments. See figure 4 for an example. It finds out what data to send to the template from the relevant models and returns the result. It is quite like HTML documents in that sense. (W3School, 2024)



```python
def home(request):
    if request.session.has_key('is_logged'):
        return redirect('/index')
    return render(request,'home/login.html')
```

Figure 4: How a view is written. Screenshot of code from the application.

Templates define the HTML structure and presentation of the application. See figure 5 for an example of what's inside a template. It is not just pure HTML language that is used here as Django uses a template language, called Django tags. This allows dynamic content rendering. These templates can include variables, loops, and conditional statements so that the integration with backend is smooth. These files are always found in a template folder within the application. (W3Schools, 2024)



```html
<!DOCTYPE html>
<html lang="en">
    <body>
        <h1>Hello!</h1>
        <p>Welcome to Expense Tracker!</p>
    </body>
</html>
```

Figure 5: An example of a basic template. It is reminiscent of HTML.

The application uses a variety of Django's in-built libraries which simplify many operations and processes. Django forms simplify handling HTML forms and user input. With it we access a set of classes for creating and validating these forms. The forms classes handle form submission, user input and generate HTML forms.

URL patterns are used to define the mapping between these URLs and their views. The urls.py file contains a list of the URL patterns which simplifies the way to navigate and manage the structure of the application. This list is a mapping between URL patterns and the Python callback functions, also known as Views. According to Django's design, URLs should refrain from using file extensions (such as .php or .asp in other languages and frameworks). The path in the URLs captures values so that when a user requests a page, Django will run through the paths and stop at the one that matches the request.

Django also includes a great user authentication and authorization system. With this library, we get built-in views, forms for registration, login and password reset. This system ensures secure user data and control access. To process requests and responses globally, Django uses middleware components. These enhance customization by adding functionality to the request and response processing pipeline.

Django static files, manages all the static assets in the application in a single location so that it can be easily served in production. These files include CSS, JavaScript and any images used, which are necessary to render the complete web page. Django collects and serves static files during development.

2. Bootstrap

This project also uses Bootstrap for some of the front-end features. Bootstrap is an open-source frontend development framework used to design web applications. It provides a large collection of template designs, along with syntax. With this, developers only need to insert code into a pre-defined grid system. It builds upon HTML, CSS, and JavaScript. In this application bootstrap is especially useful for the data tables, the navigation bar and all icons present.

Bootstrap makes it possible to detect a user screen size and adapt accordingly by including UI components, layout and the JavaScript tools needed in its framework. Bootstrap is easy to use because you can include bootstrap files as CDN links in your projects code. A CDN is a content distribution network and storing the bootstrap files as links to CDN helps improve website performance and rendering. These links help to quickly load Bootstrap CSS, JavaScript, and jQuery's libraries on our projects. (Zola, 2022)

iv. Database

The choice of SQLite as the database backend is very advantageous for this project, especially considering the project's focus on design and UI. Because of Django's Object-Relational-Mapping, it allows the developer to interact with the database using classes and methods from Python. SQLite is small, serverless, full-featured and is easy to use on small and medium-sized projects just like this one. An SQLite database is serverless, and so does not need a separate process. A database is accessed through a library. This results in no need for a separate database server, as the database design and scalability aren't the focus of the project. Thanks to Django's ORM, the database may be changed easily without any changes in the application if such a need arises. SQLite also serves as a single-file storage as well, meaning the database is stored as a single file on the disk – containing its structure and data. It also requires minimal configurations so that developers can start using it without extensive setup. (SQLite, 2023). The simplicity and ease of use makes this database perfect for developers wanting to focus on other aspects without being burdened by a complex setup. The application uses SQLite version 3, known as SQLite3. SQLite is currently the third most used database language. (Stack Overflow, 2023).

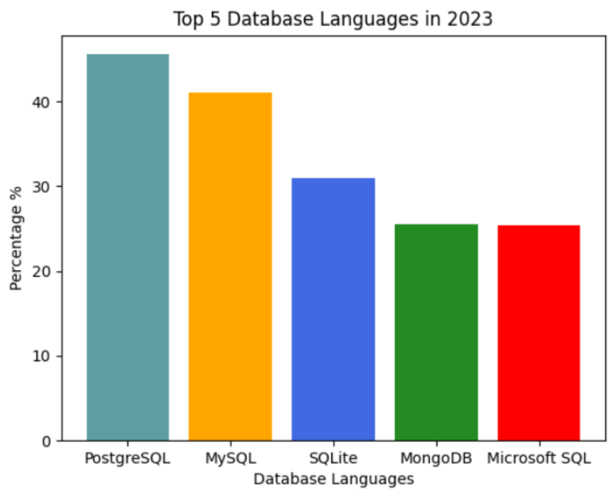


Figure 6: Graph the most popular database languages from the 2023 Stack Overflow developer survey.

v. Libraries and dependencies

The application was created on a MacBook Pro, with the newest macOS system requirements, though this is not a requirement. However, since everything in the application is standard web technologies and uses a cross-platform database there should be no issues. Using other operating systems is fine.

The project uses Node.js, that Visual Studio Code is dependent on. Python version 3.10.7 was used during development, and Python 3.10 or 3.11 should support this application. To run this application, Django needs to be installed as well. For this application, Django 4.2.6 is used, and it should be compatible with all python versions after 3.10. SQLite comes pre-installed with Python.

The application uses Django static files for the CSS and JavaScript. The static path is configured from the root for the project in settings.py. It is labeled as STATIC_URL, and all static files in the project can be found under '/static/' when the server is running. The files are as following:

CSS: static/

JavaScript: static/javascript

All the static files are used in the HTML templates and are referred to with {% static %} template tag.

CHAPTER 3: THEORETICAL FOUNDATIONS

In any web application, user engagement is incredibly important. It is not just about clicking around, but about how satisfied a user feels and how much they will keep on using the app. In this chapter I will explore the theories behind user engagement and emotional design – what keeps people interested and how design evokes emotional responses. This thesis will describe the theoretical framework of emotional design. It will outline the established models and theories which explain how users feel and get involved in digital settings. By examining the principles and components for user engagement and emotional design, we can analyze user behavior and interaction in the context of web-based applications.

i. Unified Theory of Acceptance and Use of Technology

First, let us introduce the main framework designers and developers operate under, the Unified Theory of Acceptance and Use of Technology, shortened UTAUT. It was first introduced by Venkatesh and more in “User acceptance of information technology: Toward a unified view”. (Venkatesh, Morris, Davis, & Davis, "User Acceptance of Information Technology: Toward a Unified View", 2003) It is a detailed framework that combines and expands on various theories to clarify how people adapt and use technology.

Eight previous models of technology are integrated in this UTAUT model.

1. *Theory of Reasoned Action*
2. *Technology Acceptance Model*
3. *Motivational Model*
4. *Theory of Planned Behavior*
5. *Combination Model of models 2 and 4*
6. *Model of Personal Computer utilization*
7. *Diffusion of Innovations theory*
8. *Social Cognitive Theory*

The model has four main determinants of intention and usage. A determinant in this context refers to a variable that influences a user’s decision to use and accept a technology. They are key components in the UTAUT model, and by considering them while developing we can make sure a certain technology is accepted by society. According to figure 6, these determinants are on the left side, while on the right side we have moderators on key relationships. (James & Bewsell, Enablers of Change, 2023)

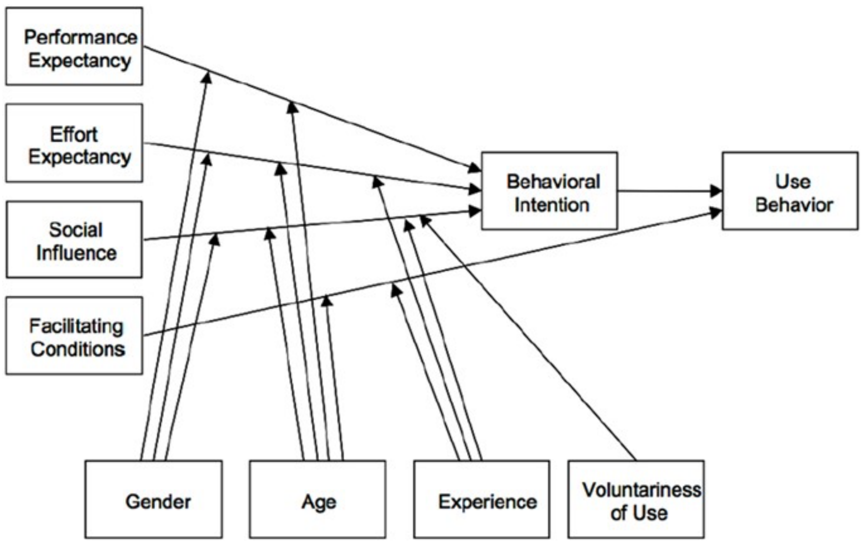


Figure 7: A diagram describing the UTAUT model. (Venkatesh, Morris, Davis, & Davis, User acceptance of information technology: Toward a unified view., 2003)

The UTAUT model provides a framework for understanding the many factors that influence technology adoption and usage. It can offer developers an appreciated insight into how the user experience can be enhanced – how to increase work efficiency and effectiveness of it. The four determinants of the model make it easier to create a conductive environment for developing technologies – and in the case of the expense tracker, hopefully make a positive change to someone's daily habits.

1. *UTAUT Determinants*

The first determinant is *performance expectancy*. It's about how a user believes technology will help them in their task and their productivity. Gender and age are factors that influence this determinant. It's easily applied by introducing technology solutions that directly tackle the challenges a user comes across.

For example, implementing a real-time budget tracking can empower users to efficiently manage their finance, which again can influence their daily habits.

The second one is called *effort expectancy*, which refers to how easy the system is to use. How easy a user finds it depends on age, gender, and experience. Implementing this determinant involves a simple interface or providing user-friendly guides and materials. This can reduce a user's effort to start using the technology. Making sure an application is intuitive and easy to navigate encourages a user to use it confidently. For instance, simplifying the process of inputting financial data in a budget tracker will influence their adoption of the technology, and inspire confidence in a user.

The third determinant is *social influence.* It is about how outside factors affect someone's choice to use the technology – like peer pressure, social norms, or personal relationships. This again can be influenced by age, gender, voluntariness of use and experience. Social influence can be strengthened by building communities or promoting platforms for sharing knowledge. They serve as a space for discussions, sharing their experience and best practices related to the technology. Peer support and positive reinforcement are strategic factors in shaping a person's decisions and can inspire others to adopt a similar practice of using the application.

The final determinant is *facilitating conditions*, which relates to having the correct resources, support and the infrastructure needed to use the technology. This means that having good internet access and technical help available are vital for this determinant. Both age and gender affect this determinant. Enhancing conditions requires ensuring that all needed resources and support are available for users wanting to use the application. It could be as simple as creating a small platform to exchange tips, troubleshoot issues, and provide guidance to each other. (James & Bewsell, Understanding the Unified Theory of Acceptance and Use of Technology, 2023)

ii. Emotional Design Principles

There are many standpoints to understand how technology affects a user's experience on an emotional level. From a psychological view there exists theories like affective computing and emotion psychology that provide insight on how human emotions can be analyzed in a user interface context. On the other hand, design theory provides principles on how arrangement and presentation of design elements evokes certain emotions in users. The Gestalt principle of perception is one of the most important design principles when it comes to improving not only aesthetics of an application, but also its functionality and user-friendliness. Additionally, the HCI – shortened for human-computer interactions, provides frameworks for developing usable products that elicit emotional responses.

Emotional design is a concept on how to develop designs that evokes emotions that results in a positive user experience by focusing on three cognitive levels – visceral, behavioral, and reflective. (Interaction Design Foundation - IxDF., 2016)

"Everything has a personality: everything sends an emotional signal. Even where this was not the intention of the designer, the people who view the website infer personalities and experience emotions"

- Donald A. Norman (Norman, 2024)

As important as a user's need is to a product, a user's response is equally important. Human emotions are complex and so are our thought processes while interacting with something. That is why Donald Norman came up with three cognitive levels to use as basis when designing. The first level is *visceral* which is a user's first impression of the application. If it's a positive one, this will lead to the user continuing to use it. To leave a good impression can be as easy as having few design elements to create an easy interface. The second level is *behavioral* where a user subconsciously appraises the application on how valuable it is to them and how easily they attain their goals. It's important here that a user feels like they are in control and don't have to spend much effort. After getting an impression, the user will consciously evaluate its performance and benefits – this is the *reflective* level. The result of this level will determine whether a user will continue using the application and spread the word about it. (Interaction Design Foundation - IxDF., 2016)

1. Gestalt Principles

Our minds are excellent at filling in blanks and putting together parts into a whole, to see structures and patterns. This is the basis of the gestalt principles of perception and is one of its most important underlying ideas. The principles base themselves on the fact that the human brain always attempts to simplify complex design elements by organizing them into a system that creates a whole. There exist six principles that appease the human brain's way of thinking and elevates any design: similarity, continuation, closure, proximity, figure/ground and lastly, symmetry & order.

Similarity is used to group elements together – as this is human nature to do. In an application all similar elements are visually grouped, without worrying about proximity. There are no specific rules to the grouping process but grouping by color and shape is common. This all influences the way a user perceives an application's structure. *Continuation* is about how the human eye moves and which path our sight follows – we always choose an even one. This is a great element to include when a developer wants a user to go from one item to the next. *Closure* builds on the fact that the human brain fills in blanks, making it a perfect element when you want to show a user there is more to an element – like scrolling or swiping.

The distance between elements is described with *proximity*. In an application, a user groups certain elements or features together when they are closer to each other. Or, by having lots of space you create a separation of the elements. The *figure/ground* principle considers how a user sees an object – if it's in the foreground or the background. A great example of this principle is pop-up windows or whenever a contrast is seen in the application. Lastly, *symmetry and order* are about how our brains view shapes in the simplest way we can. (Chapman, Exploring the Gestalt Principles of Design, 2024)

Another important framework to consider while designing an application is the human-computer interaction field. It includes multiple disciplines such as computer science and cognitive science, making it the forerunner to what we know today as User Experience. When designing with HCI in mind, it's about achieving goals within constraints. Therefore, it's important to understand the product's purpose and navigating its constraints. The main point of designing with HCI is that without a user there is no product – the user is the heart of the project. It's crucial to understand both the user and the technology to achieve a good and impactful design. (Interaction Design Foundation - IxDF., 2016)

CHAPTER 4: APPLYING EMOTIONAL DESIGN PRINCIPLES TO BUDGET BUDDY

i. User Surveys and Testing for BudgetBuddy

User testing is an extremely useful step when developing any web application. By observing how users interact with the application we get an insight into usability, functionality, and overall user experience. This chapter will go into the methodology for testing the expense tracker – what tasks and how it was conducted - and discuss the results and findings of it. The aim of this testing process is to make sure the expense tracker meets both the needs and expectations of users, so that it will get a seamless user experience.

UX usability testing methods are more useful to developers and businesses when they are done throughout the product lifecycle. There are many types of research methods to use for testing, and it can be difficult to find which ones are the best to use. The most used UX testing method today are unmoderated and moderated usability testing. Unmoderated is when participants complete tasks remotely and alone, while moderated is when there is someone guiding and observing the participants. A moderated session allows for understanding user behaviors and easily spot pain points as it encourages a more natural behavior. Therefore, in this case, it is more suitable than an unmoderated one as this one is quite restricted in openness. For example, if a participant is unsure of a task and performs it with uncertainty, the test results will not be in line with the objectives. (PlayBookUX, 2024)

Other than those two, there are methods that are specific to certain areas, like testing websites. These include card sorting, eye tracking and the 5-second test. Card sorting – using either physical or digital cards with layout elements made by testers. This method helps developers uncover if a website's layout matches how users think by organizing cards into groups.

Eye tracking is as the name suggests – it tracks a participant's eye movement. This type of testing is used to determine how participants interact with a web design. Lastly, the 5-second test focuses on users' first impressions and reactions to a website's design. It evaluates if a certain message was appropriately conveyed to a user. (PlayBookUX, 2024).

1. Methodology

All participants in the testing process will remain anonymous – which is a group of ten people. Their age range is 20-25 years, and they all have a high-level experience with technology, including using web applications. Some of the participants are highly familiar with designing web applications, while others don't give it much thought. The testing process will simulate a real-life process of using the expense tracker, both in a quiet environment without many distractions and in a more crowded and busier one.

No tools will be used except the participants' mobile phone or personal computer, and the computer the application is running on, as it is not running on any server. The participants and I will be connected to the same stable internet, so they are able test the application while it's up and running. Therefore, a moderated session will be used. It will be in-person with a moderator present to observe their actions and behavior while also listening to their thought process out loud. The participants will be given a sheet with specific tasks to perform in the expense tracker. Underneath each task there is also some free space for the participant to write notes about their experience performing the task.

2. Testing Process

The participants are given five different tasks to test all the features available in the expense tracker. They are as following:

Task 1: Imagine adding a new expense to your tracker of a recent purchase, including choosing a category, amount, and date. Take notice of the design, describe any aspects of it that you like or find difficult about it and how you feel this impacted your experience.

Task 2: You want to view a summary of your expenses – current week, month, and year. Navigate to this section of the application and describe how you experienced this. Take notice of the design and describe any elements that caught your eye and helped you understand your financial habits – or any that you disliked/found challenging.

Task 3: You realize you made a mistake in adding an expense. Demonstrate how you would correct or delete this expense. How do you feel doing this? Describe any elements that hindered or enhanced your ability to efficiently make changes.

Task 4: Imagine you have a receipt for a recent purchase and want to upload this to the expense tracker. Navigate to the receipt feature while noting the design and share your thoughts on how this experience was. Was it intuitive or difficult to use?

BUDGET BUDDY USER TESTING QUESTIONNAIRE

Task 1: Imagine adding a new expense to your tracker of a recent purchase, including choosing a category, amount, and date. Take notice of the design, describe any aspects of it that you like or find difficult about it and how you feel this impacted your experience.

Your notes:

Graphical glitches on the first page (with two notifications) are distracting.

Wallet name is confusing.It's not obvious it's the right option to add expenses, I'd expect it to be an option to add funds or something like that.

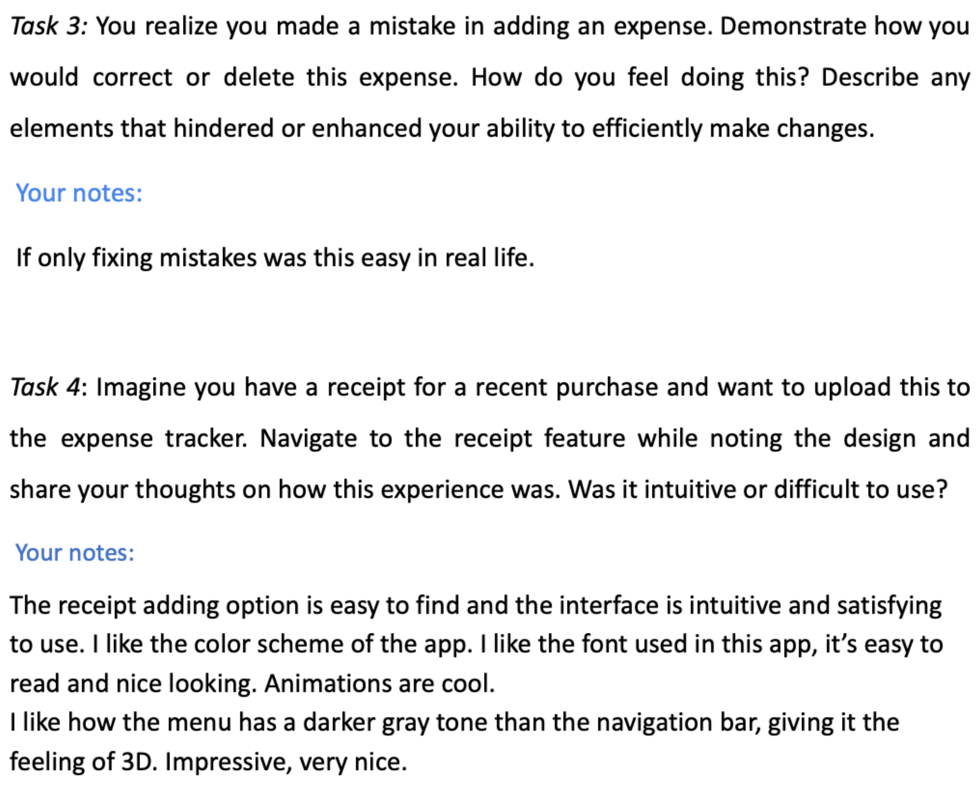
The date field with gray font and current date suggests the app automatically sets the current date which it doesn't.

The menu design looks clean and modern, I like it.

Task 2: You want to view a summary of your expenses – current week, month, and year. Navigate to this section of the application and describe how you experienced this. Take notice of the design and describe any elements that caught your eye and helped you understand your financial habits – or any that you disliked/found challenging.

Your notes:

Section is easy to find. The animations are nice and pleasing to look at. The table with graphs is not aligned with the rest of the interface and it's distracting.



Task 3: You realize you made a mistake in adding an expense. Demonstrate how you would correct or delete this expense. How do you feel doing this? Describe any elements that hindered or enhanced your ability to efficiently make changes.

Your notes:

If only fixing mistakes was this easy in real life.

Task 4: Imagine you have a receipt for a recent purchase and want to upload this to the expense tracker. Navigate to the receipt feature while noting the design and share your thoughts on how this experience was. Was it intuitive or difficult to use?

Your notes:

The receipt adding option is easy to find and the interface is intuitive and satisfying to use. I like the color scheme of the app. I like the font used in this app, it's easy to read and nice looking. Animations are cool.
I like how the menu has a darker gray tone than the navigation bar, giving it the feeling of 3D. Impressive, very nice.

Figure 8: The questionnaire filled out by an anonymous participant during a user testing session.

3. *User feedback*

The feedback of two participants will be used as an example for the user feedback in this thesis. For task one, the participants expressed some confusion over expense entry. One user mentioned confusion about finding where to add expenses, while another found the terminology confusing. There was also some confusion regarding the date field, as the application makes it look like a default date is set. The user testers liked the design aspects, such as the font, simple and organized design of the expense box, and the presence of a back button. One user specifically liked the design of edit and delete buttons on the dashboard page, while another found the navigation design to be neat and modern. Even specifying that they liked the use of two different dark gray colors and white to create some dimensions to the website.

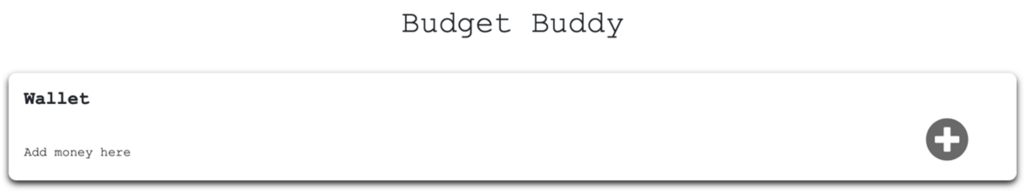
In the second task, both users expressed that they liked the colors used in the summary section, as they found them visually appealing. One user mentioned they liked the graph animation as well. Another user found it difficult to differentiate between categories due to similar colors. Both users found the edit and delete options clear and eye-catching in the third task. In the fourth and final task, they also found the process of uploading receipts clear and intuitive. They really liked the upload icon and the font as well.

Overall, the participants provided valuable insight into areas of confusion and improvement opportunities within the application. But most importantly, they gave positive feedback on design elements such as font choice and organization.

ii. Connecting BudgetBuddy to the design principles

Let's see how Donald Norman's three cognitive levels are incorporated into BudgetBuddy. At the *visceral* level, users form their first impression of BudgetBuddy based on its visual appeal and ease of use. To ensure this impression is positive, BudgetBuddy has a clean and intuitive interface with minimalistic design elements. Having a clean and uncluttered interface makes sure users are not overwhelmed by unnecessary information or design elements. Therefore, it allows users to focus on the task at hand which is managing their expenses. With the minimalistic design BudgetBuddy has – few buttons, neutral colors, and theme consistency, makes it not only easy to use, but also enjoyable to interact with. This aesthetic appeal contributes to a good first impression with the app, something the user testing confirmed it did. See figure 16 for the interface design.

Going further to the *behavioral* level, BudgetBuddy focuses on providing users with a seamless experience. Here users engage with the expense tracker on a practical level. BudgetBuddy takes efficiency in its design and functionality into consideration. Users evaluate the application based on how quickly and easily a task is accomplished – entering expenses, setting a budget, and looking at their money use. From the user testing, a task such as entering an expense may need some improvement, such as rewriting the wallet section. Users got confused with the wording here being "Add money here", thinking it means adding more money to an account or budget, not as in adding an expense or income. Budget Buddy's efficiency will easily increase only by changing this part to "Add expense or income here", which also makes users stay on track on their task.


Budget Buddy

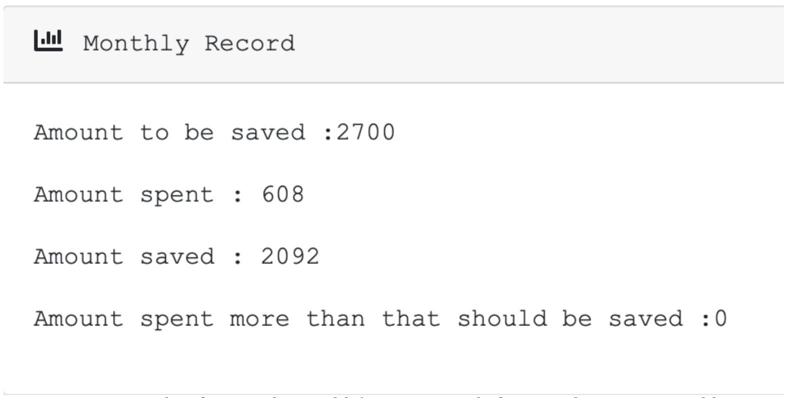
Wallet

Add money here

Figure 9: Screenshot of the Wallet section of BudgetBuddy

As BudgetBuddy offers features such as goal tracking, expense analysis and record keeping, it caters to various financial needs and objectives – making it valuable and relevant to a user's specific circumstances. Not only that, but BudgetBuddy empowers users by giving them a sense of control over their finances so they can make informed financial decisions. This again creates a positive outlook on the application's usefulness and effectiveness.

Lastly, the *reflective* level is where a user consciously checks BudgetBuddy's performance, benefits, and impact on their lives. BudgetBuddy offers users reports, which include weekly, monthly, and yearly records, accompanied by graphical representations and spending reports. These features allow them to reflect on their spending habits, identify improvement areas and track their progress over time – with clarity.



```
Monthly Record

Amount to be saved :2700

Amount spent : 608

Amount saved : 2092

Amount spent more than that should be saved :0
```

Figure 10: Screenshot from BudgetBuddy's Data Records feature, showing a monthly report.

Users reflect on the benefits they get from BudgetBuddy, which in this case are the features and functionalities aimed at improving a user's financial well-being. Features like expense categorization, budget tracking and goal setting. With this, a user evaluates how the features positively impact their lives and meet their expectations, such as saving money or achieving financial milestones. This may lead to users to recommend it to others, and so contributing to its growth and success.

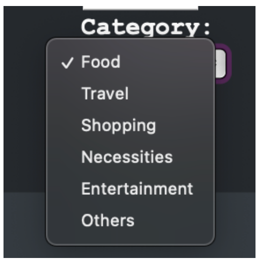
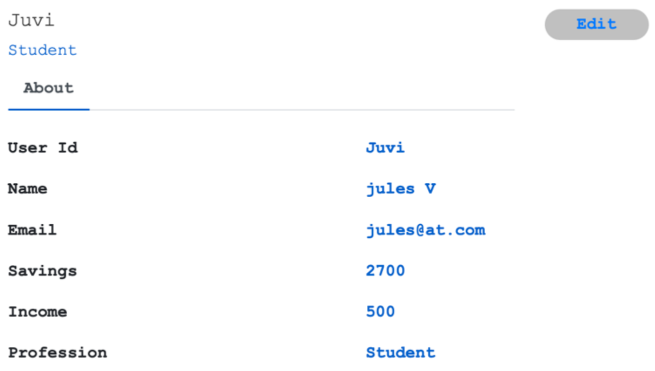


Figure 11: Screenshot from BudgetBuddy Wallet section, where a user can choose an expense category that suits their needs.



Juvi
Student
About
Edit

User Id	Juvi
Name	jules V
Email	jules@at.com
Savings	2700
Income	500
Profession	Student

Figure 12:Screenshot from BudgetBuddy's User profile section, where a user can set a savings goal, their income and profession.

BudgetBuddy also takes the Gestalt Principle of similarity into consideration when it comes to the design and applies them in various ways. Visually *grouping* similar elements together helps a user's perception of BudgetBuddy's structure. For example, expenses categorized under the same budget category are displayed using the same color in the data records.

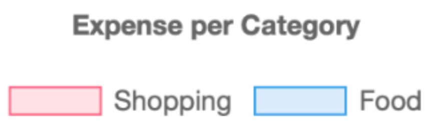
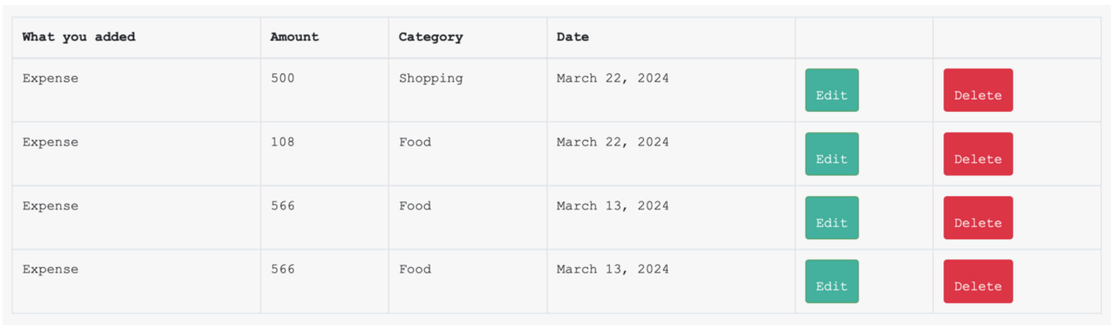


Figure 13: Screenshot of colored expense categories, where expenses are grouped together on the graphs.

Another principle utilized in BudgetBuddy, is *continuation*. This is used to guide a user's navigation through the application or a feature seamlessly. When arranging elements in a logical and fluid manner, users will be naturally led from one section or feature to the next. For example, in the dashboard page of BudgetBuddy, underneath the Wallet section, a user can browse a small table of recently inputted expenses or income. See figure 14. Here a user can easily navigate to related features to a certain expense, like expense adjustment or deletion, while following a smooth path dictated by the interface layout.

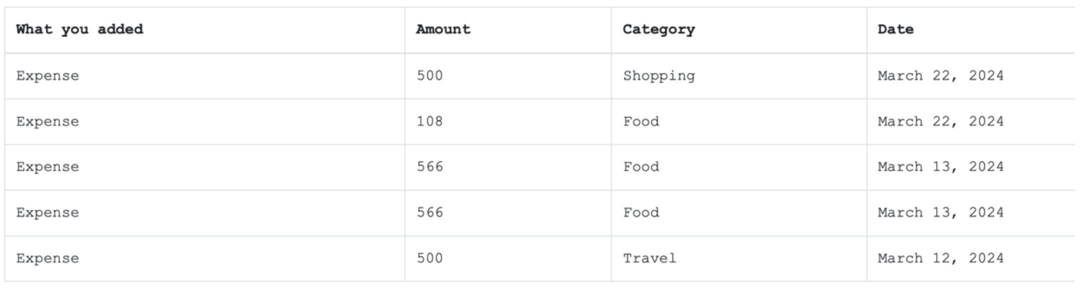


What you added	Amount	Category	Date		
Expense	500	Shopping	March 22, 2024	Edit	Delete
Expense	108	Food	March 22, 2024	Edit	Delete
Expense	566	Food	March 13, 2024	Edit	Delete
Expense	566	Food	March 13, 2024	Edit	Delete

Figure 14: Data table of recently added expenses, from BudgetBuddy's dashboard page.

Not only that, but BudgetBuddy leverages user's tendency to fill in missing information, to indicate more content – the *closure* principle. Let's say a user reaches the end of an expense list, BudgetBuddy hints at the availability of more data with a subtle scrolling indicator. This prompts users to explore further in the application.

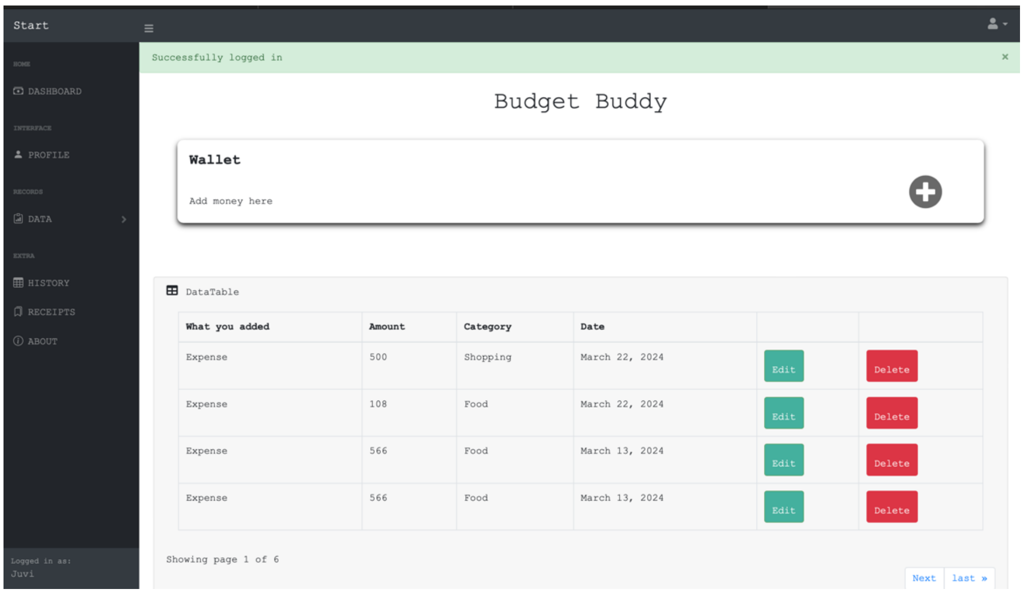
In BudgetBuddy's History section, a user can browse through all their expenses or added incomes entries – no matter how long back in time. A user can even filter their entries by date, which satisfies the *proximity* principle. Elements that are closely related – expense entries within the same timeframe, are displayed near each other, but with enough space to visually separate different sections of the application. This ensures a nice clarity in the application and reduces cognitive load.



What you added	Amount	Category	Date
Expense	500	Shopping	March 22, 2024
Expense	108	Food	March 22, 2024
Expense	566	Food	March 13, 2024
Expense	566	Food	March 13, 2024
Expense	500	Travel	March 12, 2024

Figure 15: Entries within the same timeframe are displayed together in the History section of BudgetBuddy.

The figure/ground principle is employed in BudgetBuddy to highlight important elements and distinguish them from the background. A simple example from the application is the pop-up notifications. For example, a green pop-up for a successful login or a yellow warning for exceeding savings. These are all to ensure a user's attention is drawn to critical updates or reminders.



Start

Successfully logged in ×

HOME
DASHBOARD
INTERFACE
PROFILE
RECORDS
DATA
EXTRA
HISTORY
RECEIPTS
ABOUT

Logged in as:
Juvi

Budget Buddy

Wallet

Add money here

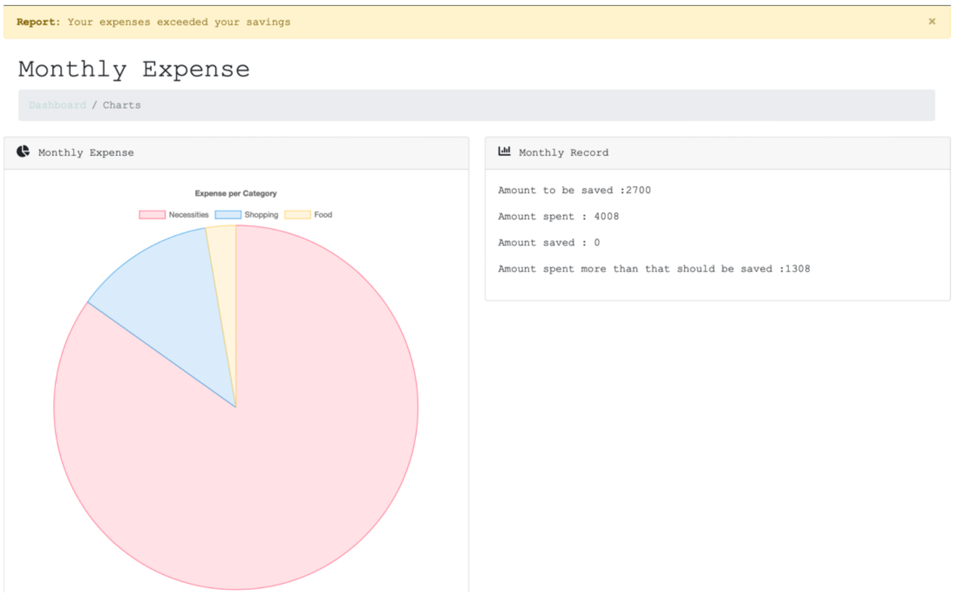
DataTable

What you added	Amount	Category	Date		
Expense	500	Shopping	March 22, 2024	Edit	Delete
Expense	108	Food	March 22, 2024	Edit	Delete
Expense	566	Food	March 13, 2024	Edit	Delete
Expense	566	Food	March 13, 2024	Edit	Delete

Showing page 1 of 6

Next last »

Figure 16: Successful login as a pop-up notification on BudgetBuddy's dashboard page.



Report: Your expenses exceeded your savings ×

Monthly Expense

Dashboard / Charts

Monthly Expense

Monthly Record

Amount to be saved :2700

Amount spent : 4008

Amount saved : 0

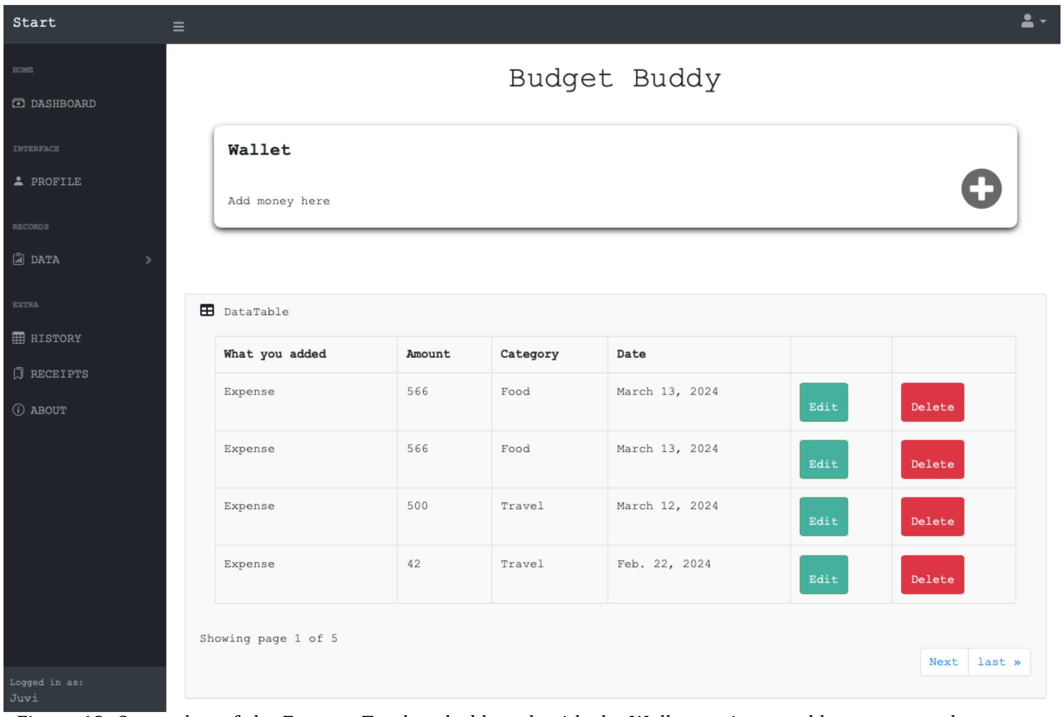
Amount spent more than that should be saved :1308

Figure 17: Expense report as a pop-up notification on BudgetBuddy's monthly expense page.

CHAPTER 5: VISUAL AESTHETICS AND BRANDING

Aesthetic design is an important element in development as it is important for perception of usability. It's not difficult to understand that humans like visually appealing and pleasing design. We can all admit we do in fact judge a book by its cover. This is explained by the phenomenon called "the halo effect", which states that humans look at good-looking people or products and assume it has other positive qualities as well. In the context of the expense tracker, considering how visual aesthetics add to the overall user experience was something that always was in the back of my mind.

There are four main categories when it comes to the aesthetics of any design: Vision, hearing, touch, and taste & smell. The one integrated into the expense tracker is *vision*, as users rely on their sight to engage and appreciate the application interface. There are many elements in this category that help to achieve good aesthetics and enhance user engagement. Some of the elements are colors, patterns, fonts, texture, balance, and shapes. (Nikolov, 2024)



Budget Buddy

Wallet

Add money here

DataTable

What you added	**Amount**	**Category**	**Date**		
Expense	566	Food	March 13, 2024	Edit	Delete
Expense	566	Food	March 13, 2024	Edit	Delete
Expense	500	Travel	March 12, 2024	Edit	Delete
Expense	42	Travel	Feb. 22, 2024	Edit	Delete

Showing page 1 of 5

Next last »

Figure 18: Screenshot of the Expense Tracker dashboard, with the Wallet section to add expenses, and an expense overview.

At first glance, the expense tracker has a rather simple design – there isn't much going on - and with good reason. Keeping a simple design is important both for usability, but also for an aesthetically pleasing design that suits a wide audience. (Nikolov, 2024) Now, let us explore the aesthetic elements a user can find in the expense tracker.

Just from the dashboard screen alone, can we see some elements already in place. The font used here, and in the entire application, is one and the same. This contributes to the overall aesthetics and readability of the interface. Not only that, but typography is a crucial visual element as it influences user perception and guides attention to important information. Bold font is also used in areas to organize content of importance, such as headings, thus making it easier for users to navigate and engage in them. Even though font consistency is a small design decision, it adds to the overall professionalism and unity of the application. (Magipik, 2023)

Next element easily seen is the use of colors, especially contrast colors. As you can see, the application's primary colors are dark gray – side navigation bar, and a white background. Using brighter colors for primary actions and a more muted color for other elements effectively prioritizes information for a user. A white background also provides for a clean and minimalistic setting to show content, while the dark gray navigation bar serves as a visual anchor and gives out a relaxing feeling. This color scheme is also associated with modern aesthetics and a consistent use of the color scheme reinforces the branding as well. Nonetheless, those are not the only colors present here. The use of red and green for the delete and edit button is also a conscious decision. It distinguishes them from the other elements and immediately shows their respective functions. Though the dashboard page is not the only place in the application with more colors. For example, in the yearly expense feature of the application multiple colors are used to showcase a user's amount of money spent in certain categories. This makes it easier to differentiate and enhances readability. (Chapman, Influence with Design: A Guide to Color and Emotions, 2024)

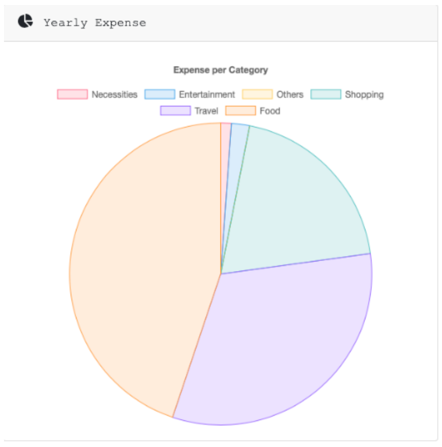


Figure 19: Screenshot of a Yearly Expense graph, taken from the Expense Tracker.

Lastly, as a user navigates through the expense tracker, they will notice the shapes used. The shape element is implemented in this application with the various icons used. Shapes evoke emotional responses and associations. They convey different meanings and in this case the features available for a user. In the navigation bar, each feature has familiar iconography to increase the intuitiveness of the application, while also being aesthetically pleasing. For instance, a person icon commonly represents profile functionality. Icons also contribute to brand recognition, and considering the application is an expense tracker, icons that fit this image are used – like the money icon for the dashboard. (Boutin, 2024). Another thing to take notice of is the use of rounded edges throughout the application. These convey a sense of friendliness in the design, creating a more approachable and inviting appearance.

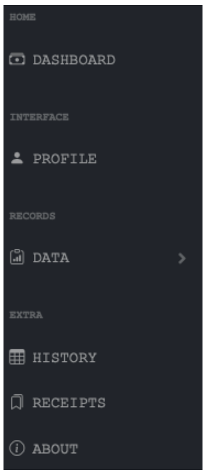


Figure 20: Screenshot from the Expense Tracker of the navigation menu.

CHAPTER 6: SUMMARY AND CONCLUSIONS

The focus of this thesis has been on exploring the connection of user experience principles and the development of an expense tracker web application. The research has explored various aspects of UX design, including emotional design, user engagement, and the practical implementation of these concepts within the expense tracker. Through research and implementation, several design principles have been explored and used in the application to enhance the user experience. Additionally, user testing was conducted to validate the application of these design principles in practice.

During this study, the research findings highlight the importance of emotional design and user engagement in enhancing usability and effectiveness of the expense tracker. User testing further confirms the importance of using design principles, as adding this to the application resulted in a more immersive and satisfying user experience. By including design elements that evoke positive emotions and resonate with users, the application has created a deep connection and increased user satisfaction. The insights gathered from this research have important implications for UX design practices in web application development. Integrating design principles like emotional design underlines the importance of creating an environment that will connect with users on a personal level. By prioritizing user engagement and emotional connection, a developer can create more successful and impactful web applications.

In conclusion, this thesis has shown the value of integrating various design principles into the development of an expense tracker web application. Having done research and including principles involving emotional design and user engagement, the application has been able to create an exciting and user-friendly experience for its audience. Moving forward, it is essential for developers to continue prioritizing UX design principles in their work. When a developer understands the impact design principles have on user satisfaction, it will result in more impactful web applications in the future.

Bibliography

A., J. (2024, January 9). *How to Link CSS to HTML Files in Web Development*. Retrieved from Hostinger Tutorials: https://www.hostinger.com/tutorials/website/how-to-link-a-stylesheet-css-file-to-your-html-file

Boutin, R. (2024, March 24). *The Psychology of Icons: Why They are so Effective*. Retrieved from Circum: https://circumicons.com/blog/the-psychology-of-icons-why-they-are-so-effective

Buczyński, R. (2023, September 21). *MVT Architecture in Django: Introduction and Comparison with MVC*. Retrieved from Medium: https://python.plainenglish.io/mvt-architecture-in-django-introduction-and-comparison-with-mvc-37cd617b542e

Chapman, C. (2024, March 16). *Exploring the Gestalt Principles of Design*. Retrieved from Toptal Designers: https://www.toptal.com/designers/ui/gestalt-principles-of-design

Chapman, C. (2024, March 24). *Influence with Design: A Guide to Color and Emotions*. Retrieved from Toptal: https://www.toptal.com/designers/ux/colors-and-emotions

Coursera Staff. (2023, November 20). *What Is Python Used For? A Beginner's Guide*. Retrieved from Coursera: https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python

Designveloper. (2023, June 28). *8 Responsive Web Design Principles You Need to Know*. Retrieved from Designveloper: https://www.designveloper.com/blog/responsive-web-design-principles/

Interaction Design Foundation - IxDF. (2016, June 3). *What is Emotional Design (ED)?.* Retrieved from Interaction Design Foundation - IxDF.: https://www.interaction-design.org/literature/topics/emotional-design

Interaction Design Foundation - IxDF. (2016, June 6). *What is Human-Computer Interaction (HCI)?* Retrieved from Interaction Design Foundation - IxDF. : https://www.interaction-design.org/literature/topics/human-computer-interaction

James, J., & Bewsell, D. (2023, July 10). *Enablers of Change.* Retrieved from Understanding the Unified Theory of Acceptance and Use of Technology: https://www.enablersofchange.com.au/understanding-the-unified-theory-of-acceptance-and-use-of-technology/

James, J., & Bewsell, D. (2023, July 10). *Understanding the Unified Theory of Acceptance and Use of Technology*. Retrieved from Enablers of Change: https://www.enablersofchange.com.au/understanding-the-unified-theory-of-acceptance-and-use-of-technology/

Magipik. (2023, July 7). *Understanding the impact of typography in design is crucial*. Retrieved from Medium: https://medium.com/@magipik/understanding-the-impact-of-typography-in-design-is-crucial-fd20562b7a1b

Nikolov, A. (2024, March 24). *What is aesthetic design?* Retrieved from Marvel Blog: https://marvelapp.com/blog/design-principle-aesthetics/

Norman, D. A. (2024, March 16). *Donald A. Norman Quotes*. Retrieved from AZ Quotes: https://www.azquotes.com/quote/1185148

PlayBookUX. (2024, April 5). *10 Popular Usability Testing Methods*. Retrieved from playbookux.com: https://www.playbookux.com/10-popular-usability-testing-methods/

S., A. (2023, August 23). *What Is HTML? Hypertext Markup Language Basics Explained*. Retrieved from Hostinger Tutorials: https://www.hostinger.com/tutorials/what-is-html

Simplilearn. (2023, August 7). Retrieved from Simplilearn: https://www.simplilearn.com/applications-of-javascript-article

SQLite. (2023, Oktober 10). *About SQLite*. Retrieved from SQLite: https://www.sqlite.org/about.html

Stack Overflow. (2023, June). *Stack Overflow Developer survey 2023*. Retrieved from Stack Overflow Survey: https://survey.stackoverflow.co/2023/#technology-most-popular-technologies

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). "User Acceptance of Information Technology: Toward a Unified View". *MIS Quarterly*, pp. 425-478.

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*.

Visual Studio Code. (2023, June 19). *Django Tutorial in VS code*. Retrieved from Visual Studio Code: https://code.visualstudio.com/docs/python/tutorial-django

Visual Studio Code. (2024, Januar 2). *Visual Studio Code*. Retrieved from Visual Studio Code: https://code.visualstudio.com/docs/editor/whyvscode

W3School. (2024, February 14). *Django Views*. Retrieved from W3School: https://www.w3schools.com/django/django_views.php

W3Schools. (2024, February 11). *AJAX W3Schools*. Retrieved from W3Schools: https://www.w3schools.com/js/js_ajax_intro.asp

W3Schools. (2024, February 11). *Django Tutorial*. Retrieved from W3schools: https://www.w3schools.com/django/django_intro.php

Zola, A. (2022, August). *Bootstrap*. Retrieved from TechTarget: https://www.techtarget.com/whatis/definition/bootstrap