

Dapper

Mohamed Ryad Ammar, 260928403, ECSE 458N1-001
Mohamed Amine Benakmoum, 260928587, ECSE 458N1-001
Zakaria Bouhemhem, 260930807, ECSE 458N1-001

DP 02

Prof. Majid Babaei

McGill University

TABLE OF CONTENTS

Abstract	3
Introduction	4
Background	5
Requirements	6
Design	7
Business Process Model and Notation (BPMN)	7
Unified Modeling Language (UML)	7
User Experience	9
Main Page:	9
Sign In Page:	10
Select Wear Style Page:	12
User Profile Page:	13
About Page:	14
Billing and shipping information pages:	15
Database	16
NoSQL Database for Dapper	16
Relational Database for Dapper	16
Exploring three NoSQL databases	17
Language	19
Back-end Framework	19
Front-end and CSS Framework	20
Tools	20
Continuous integration and Continuous development	20
Architecture	21
Plan for Next Semester	22
Impact on Society and the Environment	24
Individual Contribution	26
Collaboration:	26
Difficulties and Plan:	26
Conclusion	27
References	28

Abstract

Dapper is a subscription-based application that delivers personalized outfits based on personal preference. The outfits proposed would be tailored by fashion stylists to the user's individual fashion preferences, current season, preferred colors, size, and subscription level. The outfits will be shipped to the user's residence every month and are picked up by the delivery man the next month. The service provides users with the opportunity to experiment with various outfits and offers them the option to keep any items they like by purchasing them.

The project team's goal is not only to develop a fully functional web application but also to gain essential startup skills and marketing expertise. These skills will serve as valuable tools for team members in their future careers, regardless of their specific industry.

Creating a domain model and BPMN diagram was a crucial step for the project team. It allowed them to understand the system's requirements and visualize the process flow of the application, which led to more efficient and effective development. Through extensive research on the fashion industry and market analysis, the team identified potential opportunities and challenges, providing valuable insights into the industry's trends and demands. In addition to research and modeling, the project team has utilized other methods to achieve their milestones, such as prototyping, designing and selecting the most suitable technologies for implementing the application at a later stage.

The expected outcome of the project is not only to establish a successful startup company but also to provide customers with personalized outfits that make them feel confident and fashionable. By combining fashion and technology, Dapper has the potential to change the way people shop for clothes and contribute to a more sustainable and inclusive fashion industry.

Introduction

Fashion has always been an integral part of human life. It has evolved over the years, but the conventional methods of shopping for clothes have remained largely unchanged. Traditional shopping involves physically visiting stores, spending hours browsing through clothes, and often settling for options that don't meet our needs. This approach is not only time-consuming but also unsustainable and can lead to environmental concerns.

Dapper aims to revolutionize the way people shop for clothes by providing a personalized and sustainable experience. Through its user-friendly web application, customers can easily get dressed through a curated selection of outfits tailored to their preferences and sizes. This eliminates the need for physically visiting stores, saving customers time and reducing their carbon footprint.

The fashion industry's sustainability practices have also been a topic of concern in recent years. The industry is responsible for a significant amount of environmental pollution, and fast fashion has contributed to the generation of textile waste. Dapper aims to address these concerns by promoting sustainable fashion. The platform features eco-friendly and sustainable clothing options, encouraging customers to make environmentally conscious choices while shopping.

Dapper's potential impact is not limited to sustainable fashion. The platform also aims to provide a hassle-free shopping experience to customers. With its personalized approach, customers can feel confident in their purchases and experiment with new styles without the pressure of overspending. This can lead to increased customer satisfaction and loyalty, which is crucial for any business's success.

In addition to providing a unique shopping experience, Dapper also serves as a learning opportunity for the project team. Through this project, the team aims to gain essential startup skills and marketing expertise that will be valuable in their future careers. The platform's success will depend on the team's ability to develop and implement effective marketing strategies, highlighting the importance of gaining such skills.

In conclusion, Dapper is a subscription-based fashion platform that aims to revolutionize the way people shop for clothes. By promoting sustainable and personalized fashion, Dapper has the potential to contribute to a more inclusive and environmentally conscious fashion industry. Additionally, the platform's hassle-free shopping experience and the project team's learning opportunities make Dapper an exciting venture with significant potential for success.

Background

The fashion industry is a dynamic and multifaceted field that encompasses a broad range of activities, including design, manufacturing, marketing, and retailing. To comprehend the opportunities and challenges of the industry, extensive research was conducted on its trends and demands. The current market landscape was analyzed, potential competitors were identified, and the potential for sustainable and personalized fashion was evaluated.

To develop innovative solutions, the principles of design thinking, user-centered design, and agile development methodologies were studied. Design thinking emphasizes empathy, ideation, and iteration to solve problems. User-centered design prioritizes designing products and services that meet the needs and preferences of users, while agile development methodologies prioritize flexibility, collaboration, and rapid iteration to deliver products and services.

Furthermore, the concepts of business modeling and market analysis were explored to identify potential opportunities and challenges. Business modeling involves creating a blueprint for how the business will operate, including revenue streams, cost structure, and value proposition. Market analysis involves evaluating the target market's size, demographics, behavior, and preferences to understand the market's potential demand for the product or service.

In addition to the concepts discussed above, the team also delved into the emerging trends in the fashion industry. The rise of e-commerce and social media has transformed the way consumers shop for fashion, and has created new opportunities for brands to connect with their audience. In particular, social media platforms such as Instagram and TikTok have become powerful tools for fashion marketing and have created a new breed of fashion influencers who wield significant influence over their followers.

Another important trend in the fashion industry is sustainability. As consumers become increasingly aware of the environmental impact of the fashion industry, there has been a growing demand for sustainable and eco-friendly products. This has led to the emergence of new sustainable fashion brands, as well as established brands shifting their focus towards sustainable practices.

The team also studied the role of data analytics in the fashion industry. In recent years, the use of data analytics has become increasingly important for fashion brands to understand their customers, predict trends, and optimize their operations. This has led to the emergence of new technologies such as artificial intelligence and machine learning, which are being used to analyze large volumes of data and make informed business decisions.

Another area of focus was the importance of user experience in the fashion industry. With the rise of e-commerce, the online shopping experience has become a critical factor in driving sales and building brand loyalty. This has led to the emergence of new technologies such as virtual try-on and augmented reality, which allow customers to visualize how products will look on them before making a purchase.

Finally, the team also studied the impact of globalization on the fashion industry. With the rise of fast fashion and outsourcing, the production and supply chains of many fashion brands have

become global in nature. This has created new opportunities for brands to access new markets and reduce costs, but has also led to ethical and sustainability concerns.

Overall, the team's background research aimed to provide a comprehensive understanding of the fashion industry and its evolving landscape. This knowledge has been instrumental in guiding the development of the Dapper platform and ensuring its success in the market.

Requirements

The Dapper platform aims to address the growing demand for personalized and sustainable fashion in the market. The current fashion industry is dominated by fast fashion brands that produce clothing in mass quantities, resulting in a negative impact on the environment and exploitation of labor in developing countries. Furthermore, the one-size-fits-all approach of these brands fails to cater to the diverse preferences and body types of customers, leading to a lack of individuality in fashion.

The Dapper platform seeks to solve these problems by providing a personalized fashion experience that takes into account the unique preferences and body types of each customer. The platform will offer a range of sustainable fashion options from different brands, allowing customers to choose products that align with their values and ethics. The platform will also leverage technology such as artificial intelligence and data analytics to provide personalized recommendations and a seamless shopping experience.

To achieve this, the following requirements and constraints were identified:

1. The platform must be user-friendly and easy to navigate, with a clear and intuitive user interface.
2. The platform must incorporate features such as virtual try-on and augmented reality to enhance the shopping experience and provide customers with a realistic view of the product.
3. The platform must have a robust data analytics system that can analyze customer data to provide personalized recommendations and insights.
4. The platform must integrate with existing systems such as payment gateways and inventory management systems to ensure a seamless shopping experience for customers and brands.
5. The platform must prioritize sustainability by offering products that are ethically sourced, manufactured, and packaged, and by minimizing waste and carbon footprint.
6. The platform must comply with relevant laws and regulations such as data privacy laws, consumer protection laws, and environmental regulations.
7. The platform must be developed within a limited budget and time frame, with a focus on delivering a minimum viable product (MVP) that can be iterated upon in future versions.
8. The platform must have a responsive customer support system that can address customer queries and complaints in a timely and effective manner.
9. The platform must offer different customization options for customers, such as size, color, and style, to cater to their diverse preferences.
10. The platform must provide a platform for small and independent fashion brands to showcase their products and reach a wider audience.

Overall, the Dapper platform seeks to provide a solution to the problems of the current fashion industry by offering a sustainable and personalized fashion experience. The identified requirements and constraints will guide the development of the platform and ensure that it meets the needs of customers and brands while delivering a positive impact on the environment and society.

Design

Business Process Model and Notation (BPMN)

A Business Process Model and Notation (BPMN) diagram is a visual representation of a business process, which can help in planning, designing, and implementing the project effectively. In the case of Dapper, a BPMN diagram can be useful for several reasons. Firstly, it can provide a clear understanding of the various steps involved in the process, such as user registration, outfit selection, shipping, and billing. This can help in identifying any potential bottlenecks and areas for improvement. Secondly, a BPMN diagram can help in defining roles and responsibilities for each step in the process. This can ensure that everyone involved in the project has a clear understanding of their tasks and can work together efficiently. Finally, a BPMN diagram can serve as a communication tool between team members, stakeholders, and other involved parties. Overall, a BPMN diagram can be a valuable starting point for the Dapper project, helping to ensure that the process is well-defined, efficient, and effectively communicated to all stakeholders.

Due to the size of our BPMN diagram, we are unable to include it in the report, and therefore, we have provided a [link](#) to it for reference.

Unified Modeling Language (UML)

Unified Modeling Language (UML) is a graphical language used to visualize, specify, and document the software system's design. In the case of Dapper, a UML diagram can play a crucial role in representing the software architecture and system design. Firstly, it can help in identifying the various components of the system and their interactions, such as the user interface, database, and backend systems. This can ensure that the system is well-organized and each component is well-integrated into the overall system. Secondly, a UML diagram can help in representing the software system's behavior, such as the sequence of user interactions and the system's response. This can help in identifying any potential issues with the system's functionality and ensure that it meets the user's requirements. Finally, a UML diagram can serve as a communication tool between developers, designers, and other stakeholders involved in the project. It can help in conveying complex information in a clear and concise manner, making it an effective tool for collaboration and decision-making. Overall, a UML diagram can be crucial for the success of the Dapper project, ensuring that the software system is well-designed, functional, and meets the user's needs.

Please see the UML diagram on the next page.

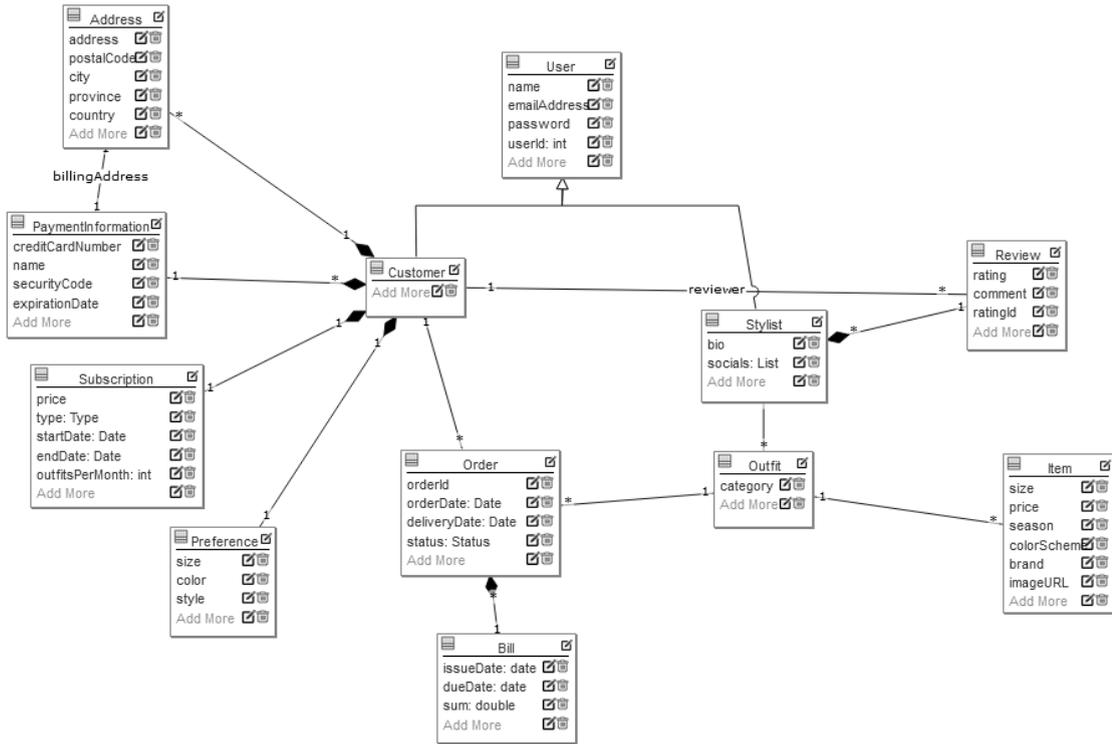


Figure 1. Visualizing the architecture of Dapper with UML diagram

User Experience

Main Page:

The main page of the Dapper application serves as the user's entry point to the application. The page has a clean and modern design that reflects the brand's style and values. The page provides the user with a clear and concise overview of the application's main features and benefits. The page allows the user to choose between signing up or signing in, making it easy to get started with the application. The user can also navigate to the "Contact Us" page to get in touch with the Dapper team, learn more about how the application works through the "How it works" page, and discover more about the brand through the "Who we are" page. Additionally, the page allows the user to choose a plan that best suits their needs, making it easy for them to subscribe to the service. Overall, the main page of the Dapper application is designed to provide the user with a seamless and user-friendly experience that encourages them to engage with the brand and explore the application's features.

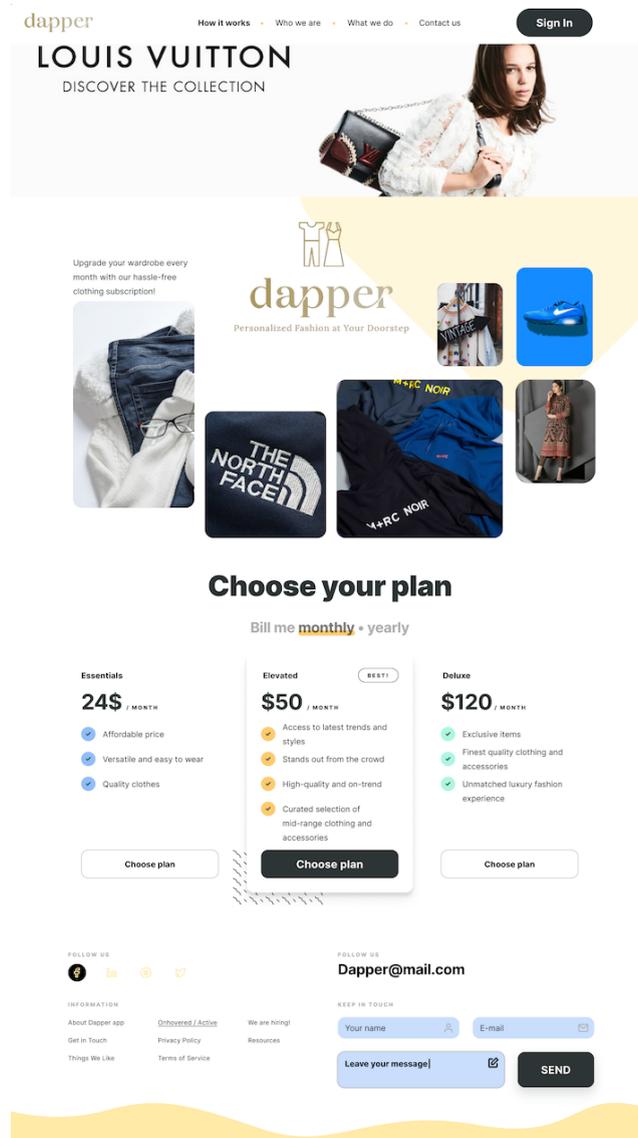


Figure 2. Main Page

Sign In Page:

The sign-in page of the Dapper application is designed to be simple and intuitive, making it easy for users to log in to their accounts. The page allows users to enter their login credentials such as their email and password, or sign in using their Google, Apple or Facebook accounts. By providing multiple sign-in options, the page ensures that users can easily and quickly access their accounts using their preferred method. Additionally, the sign-in page provides users with links to the mobile app version of the application on both Android and iOS platforms, making it easy for users to access the application from their preferred devices. The links to the mobile app versions are prominently displayed on the page, making it easy for users to navigate to the mobile app store and download the application. In addition to providing the option to sign in, the Dapper sign-in page also offers users the ability to register for a new account if they do not have one yet. The registration process is simple and straightforward, and users are guided through the process step-by-step. Additionally, the sign-in page provides a "forgot password" option that allows users to reset their password in case they have forgotten it. This feature adds an extra layer of security to the sign-in process and ensures that users can easily regain access to their accounts in the event that they forget their password. Overall, the Dapper sign-in page provides a range of features and options that make it easy and convenient for users to sign in to their accounts, register for a new account, and recover their password if necessary.

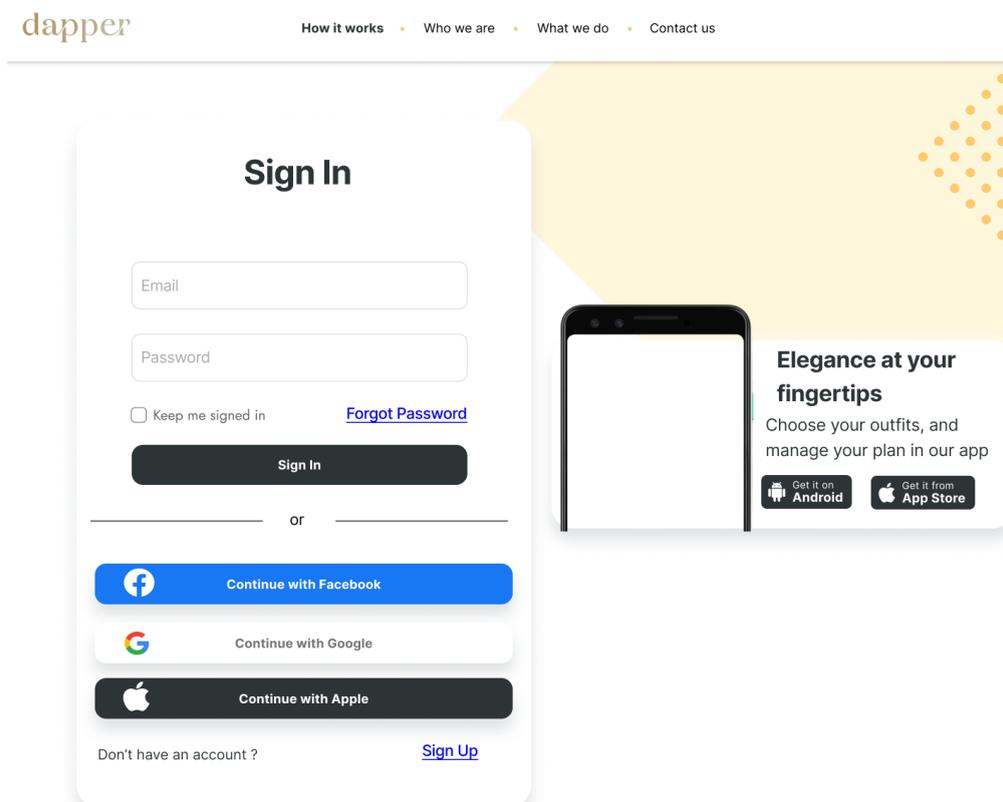


Figure 3. Sign In Page Create Account Page:

The sign-up page of the Dapper application is designed to be straightforward and easy to use. The page allows users to create a new account by providing their first and last name, email address, and a password of their choice. The page also provides users with links to the mobile app version of the application on both Android and iOS platforms, making it easy for users to access the application from their preferred devices. Overall, the sign-up page of the Dapper application is designed to provide a simple and streamlined sign-up process for users, while also encouraging them to download the mobile app version of the application for a more comprehensive experience.

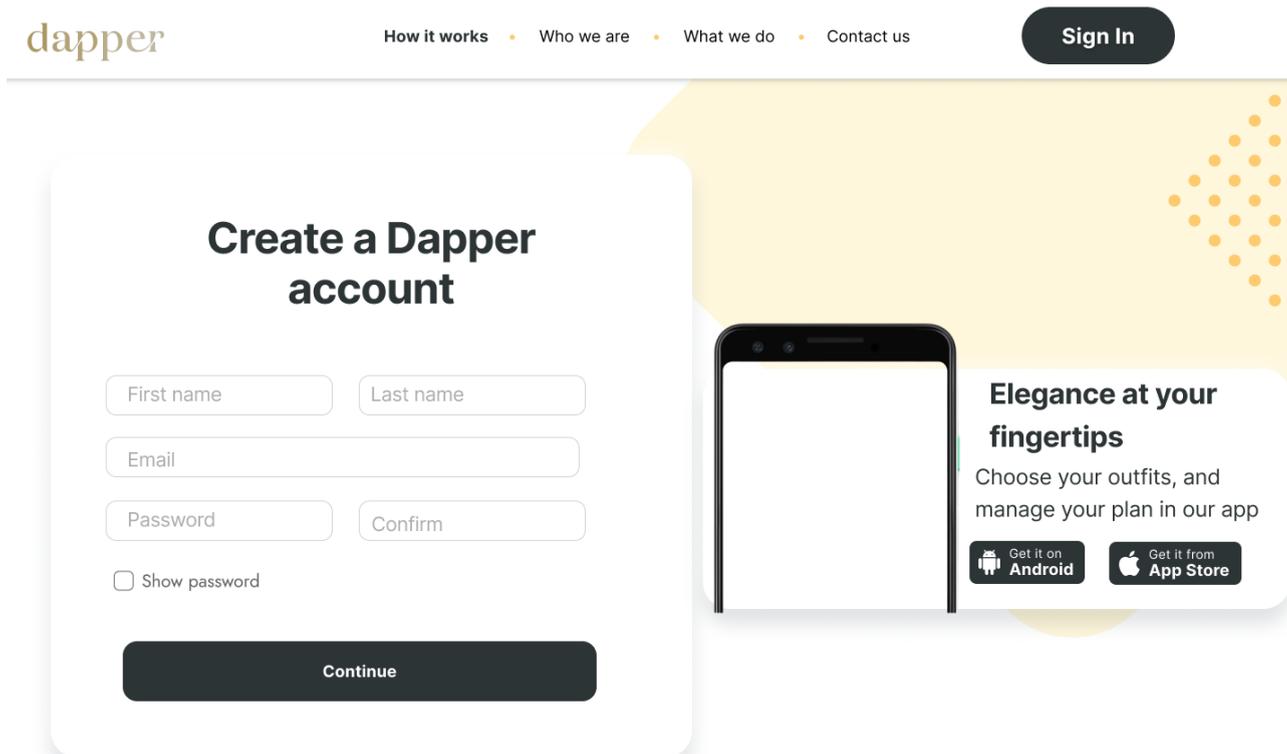


Figure 4. Create Account Page

Select Wear Style Page:

The select wear style page of the Dapper application is designed to provide users with a range of fashion styles to choose from, allowing them to find the perfect look that suits their individual preferences. The page displays a variety of men's and women's wear styles, including popular options such as streetwear, casual wear, business attire, and more. Each style is presented with high-quality images that showcase the key features and elements of the outfit, giving users a clear idea of what they can expect. Users can browse through the available styles and select the one they are interested in trying on. The page is designed to be intuitive and easy to navigate, with clear and prominent buttons that allow users to view additional details about each style and select their preferred option. Overall, the select wear style page of the Dapper application is designed to provide users with a range of stylish and fashionable options to choose from, while also making it easy and convenient for them to select their preferred style and begin the outfit selection process.

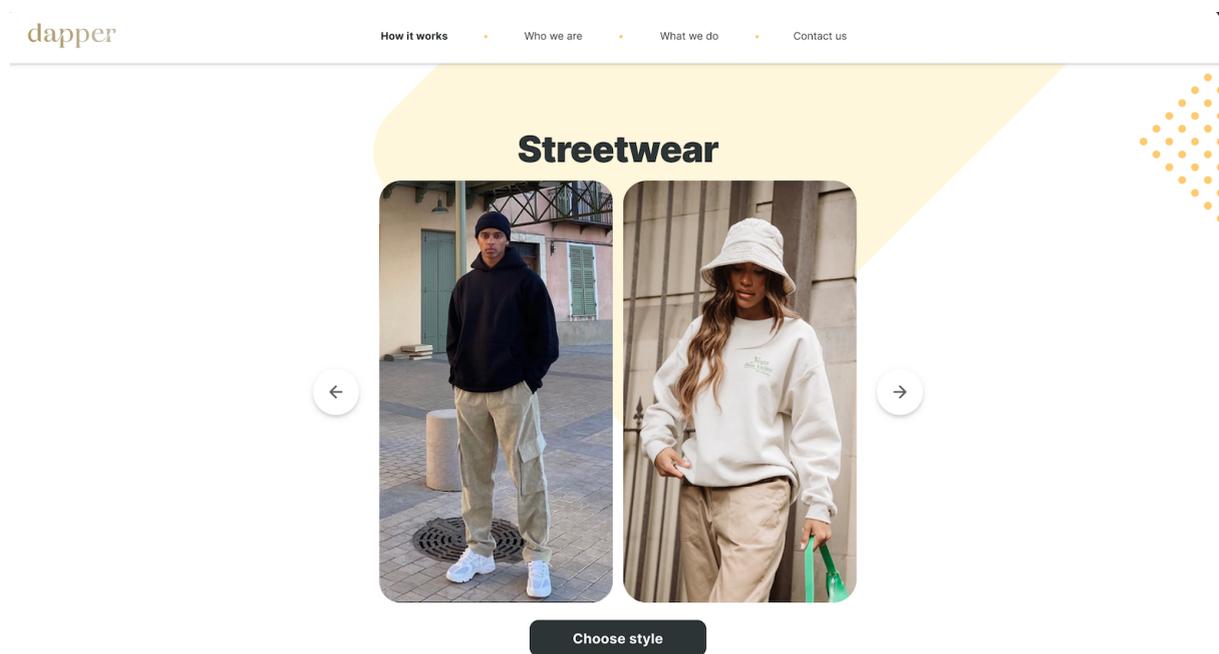


Figure 5. Select wear style Page

User Profile Page:

The user profile page of the Dapper application is designed to provide users with a personalized and customized experience, allowing them to view and manage their style preferences. Once the user has selected their preferred wear style, the user profile page displays a range of options that can be customized to suit the user's individual preferences. This includes a palette of color preferences, allowing users to select the colors they prefer for their outfits. In addition, the page allows users to select their gender and size of the clothes, ensuring that the outfits selected are tailored to the user's individual needs and preferences. Overall, the user profile page of the Dapper application is designed to provide users with a simple and user-friendly interface for managing their style preferences, ensuring that they receive outfits that match their individual preferences and needs.

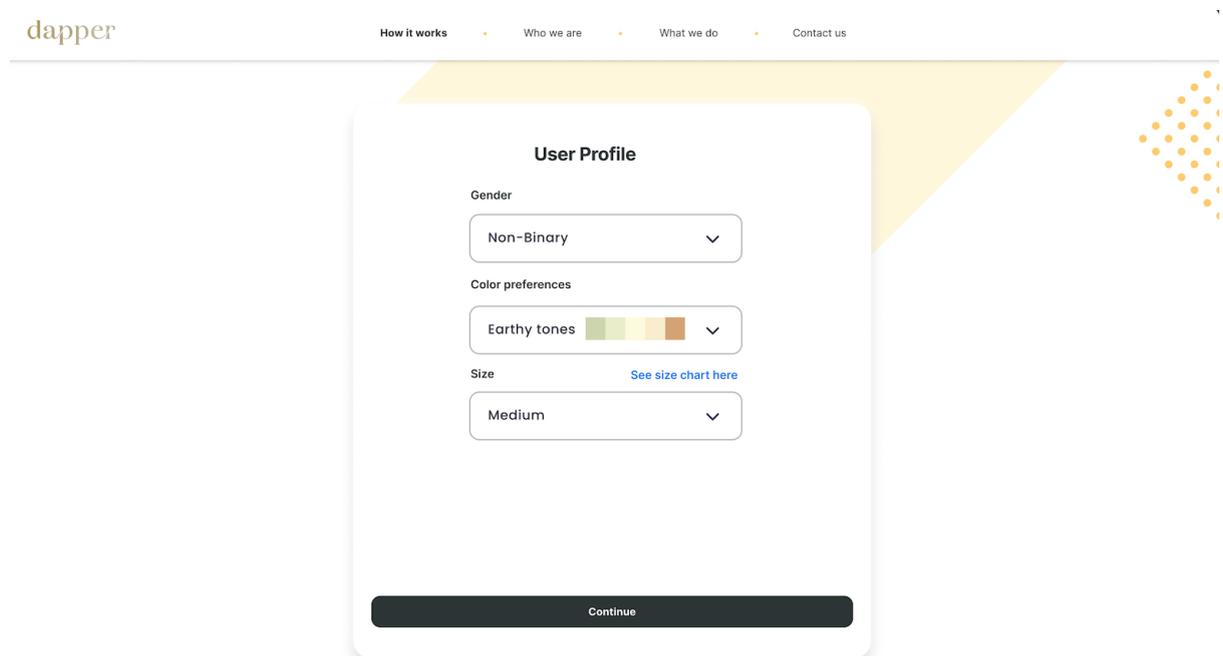


Figure 6. User profile Page

About Page:

The About page of the Dapper application is designed to provide users with an insight into the people behind the app. The page displays the images of the developers and founders of the app, along with background text for each of them. This allows users to learn more about the team that created the app and their motivations for doing so. The developers and founders are presented in a professional manner, with their images accompanied by a brief summary of their experience and qualifications. The text is designed to provide users with an understanding of the expertise that the team brings to the development of the app. The About page also provides users with an overview of the app and its features, highlighting the unique selling points of the app and its potential benefits to users. This information is presented in a clear and concise manner, making it easy for users to understand the value that the app provides.

Overall, the About page is designed to provide users with an insight into the team behind the Dapper app and the benefits that the app can provide. It is an important element of the app, providing users with confidence in the team's expertise and motivations, and helping to build trust in the app's brand.



Figure 7. About Page

Billing and shipping information pages:

The Shipping and Billing pages are two important elements of the Dapper application, providing users with a seamless and secure checkout process. The Shipping page is designed to allow users to enter their shipping information, including their full name, shipping address, and phone number. Users can also provide delivery instructions, such as specific delivery times or instructions for leaving the package in a secure location. This page is crucial to ensure that the user receives their monthly outfit delivery without any issues. The Billing page is where users can provide their billing information, including their full name, billing address, and payment details. In addition to standard credit card payment options, users are also able to pay using Google Pay and PayPal. This provides users with a range of secure and convenient payment options, ensuring that they are able to complete their purchase in a way that suits them. Both pages are designed with a clean and user-friendly interface, making it easy for users to enter their information quickly and efficiently. The security of user data is a top priority, and the pages are designed with encryption and other security measures to ensure that user information remains secure at all times.

Overall, the Shipping and Billing pages are essential components of the Dapper application, providing users with a streamlined and secure checkout process, and allowing them to receive their personalized outfits with ease.

The figure displays two screenshots of the Dapper application's checkout process. The left screenshot shows the Shipping Information page, which includes fields for First name *, Last name *, Address line 1 *, Address line 2, City *, Postal code *, and Phone *. There is also a Delivery instructions dropdown menu with the option "Leave on front porch" and a Continue button. The right screenshot shows the Billing Information page, which includes a "Pay with" section with options for Google Pay, PayPal, and "See More Methods". Below this is a "Pay with Credit Card" section with a checkbox for "Use shipping address". The Billing Information section includes fields for First name *, Last name *, Address line 1 *, Address line 2, Country *, City *, State/Province *, and Postal code *. There is also a Credit Card Information section with fields for Credit Card Number, Expiry Date (MM/YY), and CVV. Both pages have a Continue button at the bottom.

Figure 8. Billing and Shipping Information Pages

Database

The choice of a suitable database for Dapper is critical for the project's success, as it can impact the application's scalability, performance, and flexibility. In this section, we will compare the pros and cons of using a NoSQL database versus a relational database for Dapper.

NoSQL Database for Dapper

NoSQL databases are designed to handle large volumes of unstructured or semi-structured data efficiently. They are also highly scalable and can handle high-velocity data updates with ease. In the case of Dapper, a NoSQL database like MongoDB could be a better choice if the application needs to handle large amounts of unstructured data, such as user-generated content like photos and comments. The following are the pros and cons of using a NoSQL database for Dapper:

Pros

- **Scalability:** NoSQL databases are highly scalable and can handle large volumes of data efficiently. This is essential for an application like Dapper that may need to store and retrieve large amounts of data.
- **Flexibility:** NoSQL databases offer greater flexibility when it comes to changing the data model and schema. This can be useful for applications that are still evolving or undergoing frequent changes, making it easier to add new features or modify existing ones.
- **High Performance:** NoSQL databases can handle high-velocity data updates with ease, making them ideal for applications that require high read/write performance.
- **Cost-effective:** NoSQL databases are often less expensive to operate than relational databases, as they require less hardware and maintenance.

Cons

- **Data Consistency:** NoSQL databases are designed for eventual consistency, which means that data may not be consistent across all nodes immediately. This can be a problem for applications that require immediate consistency, such as financial transactions.
- **Limited Query Capabilities:** NoSQL databases offer limited query capabilities compared to relational databases, which can make it difficult to perform complex queries.

Relational Database for Dapper

Relational databases like MySQL or PostgreSQL are well-suited for handling structured data and provide reliable and efficient performance for large datasets. In the case of Dapper, a relational database could be a good fit if the application needs to store and retrieve structured data related to users, outfits, and transactions. The following are the pros and cons of using a relational database for Dapper:

Pros

- **Data Consistency:** Relational databases offer strong data consistency, making them ideal for applications that require immediate consistency, such as financial transactions. [4]
- **Advanced Query Capabilities:** Relational databases offer advanced query capabilities, making it easier to perform complex queries and retrieve specific data.
- **Mature Ecosystem:** Relational databases like MySQL and PostgreSQL have a mature ecosystem and a wide range of tools and resources available, making it easier to manage and maintain the database over time. [2]

Cons

- **Limited Scalability:** Relational databases may struggle to handle large volumes of data efficiently, making them less suitable for applications that require high scalability. [3]
- **Rigid Schema:** Relational databases have a rigid schema, which can make it challenging to add new features or modify existing ones. [3]

After careful consideration of the pros and cons of both NoSQL and relational databases for Dapper, the team has decided to opt for a NoSQL database like MongoDB. The team believes that the flexibility and scalability of a NoSQL database will be crucial in accommodating the evolving needs of the application as it grows. Additionally, the ability to handle large amounts of unstructured data and provide high read/write performance is an essential factor in ensuring the smooth functioning of the application. Finally, the cost-effectiveness of NoSQL databases makes it an attractive option for a project like Dapper that is in its early stages and has a limited budget. [2, 4, 5]

Exploring three NoSQL databases

Before making a decision on which database to use for the Dapper project, we compared three popular NoSQL databases - MongoDB, Cassandra, and Couchbase - based on several key criteria:

1. **Data Model:** Refers to the way data is structured in the database.
2. **Scalability:** Refers to how well the database can handle increasing amounts of data and traffic. Automatic scalability means that the database can scale up or down as needed without manual intervention, while limited scalability requires manual adjustments.
3. **Data Size:** Refers to the amount of data that can be stored in the database.
4. **Querying:** Refers to the ability to search and retrieve data from the database.
5. **Write Speed:** Refers to how fast the database can write new data.
6. **Read Speed:** Refers to how fast the database can retrieve data.
7. **Consistency:** Refers to how consistent the data is across multiple nodes in a distributed database. Eventual consistency means that the data will eventually become consistent, while tunable consistency allows for adjusting the level of consistency. [4]
8. **ACID:** Refers to the set of properties that guarantee reliability and consistency in database transactions. Partial ACID means that the database provides some but not all of the ACID properties, while optional ACID means that ACID compliance is optional.

9. **Community:** Refers to the size and activity level of the user community around the database. A larger and more active community can provide better support and resources.
10. **Cost:** Refers to the price of the database. All three databases listed are available for free, but some enterprise-level features may come with a cost. [2, 3]

Table 1: Comparison of NoSQL Databases for the Dapper Project

Criteria	MongoDB	Cassandra	Redis
Data Model	Document	Column	Key-value
Scalability	Automatic	Automatic	Limited
Data Size	Large	Very Large	Small to Medium
Querying	Rich	Limited	Limited
Write Speed	Fast	Very Fast	Very Fast
Read Speed	Fast	Fast	Fast
Consistency	Eventual	Tunable	Eventual
ACID	Partial	Partial	Optional
Community	Large	Moderate	Large
Cost	Free	Free	Free

Based on the comparison table (See Table 1), we can see that MongoDB would be a good fit for the Dapper project for several reasons: [8]

1. **Data Model:** MongoDB's document-based model aligns well with Dapper's need to store personalized outfit information for each user.
2. **Scalability:** MongoDB's automatic scalability means that as the Dapper user base grows, the database can easily scale up or down as needed.
3. **Data Size:** Dapper's need to store large amounts of outfit and user data makes MongoDB's ability to handle large data sets a good fit.
4. **Querying:** MongoDB's rich querying capabilities will enable Dapper to quickly search and retrieve outfit data for each user based on their preferences.
5. **Write Speed:** MongoDB's fast write speed is a good fit for Dapper's need to store and update outfit data for each user in real time.
6. **Read Speed:** MongoDB's fast read speed will enable Dapper to quickly retrieve outfit data for each user when needed.
7. **Consistency:** MongoDB's eventual consistency aligns well with Dapper's need for consistent outfit data over time.
8. **ACID:** Although MongoDB only provides partial ACID compliance, it provides enough reliability and consistency for Dapper's needs. [4]

9. **Community:** MongoDB has a large and active user community, which provides a wealth of resources and support.
10. **Cost:** MongoDB is available for free, which aligns well with Dapper's need to keep costs low.

Overall, based on the comparison table and Dapper's specific needs, MongoDB would be a good choice for the project.

In MongoDB, the data is stored in collections, which are similar to tables in a relational database. Each document within a collection represents a record, and each document can have its own unique structure. While this flexibility allows for easier data storage and modification, it also means that careful consideration should be given to designing the structure of collections and documents to ensure that they align with the application's needs and enable efficient querying and indexing. As of now, we have not designed the database structure for the Dapper project. However, we plan on doing so in the next iteration of the project. We understand the importance of database design, and we will ensure that the structure is well thought out and aligned with the project's needs before moving forward with implementation. [6, 7]

Language

We chose Python as the language for implementing the Dapper project for several reasons. First, Python is a popular language for web development and has a large and active community of developers, which means there are many resources available for learning and problem-solving. Second, Python is known for its ease of use and readability, which can lead to more efficient development and easier maintenance of the codebase. Third, Python has a vast array of useful libraries and frameworks for web development, such as Flask and Django, which can save development time and simplify the implementation of certain features. In addition to the above reasons, another factor that influenced our decision to choose Python for the Dapper project is that the development team is already familiar with the language. Having prior experience with Python means that the team can hit the ground running and be more productive from the outset of the project. This familiarity with the language can also help to reduce the learning curve and potential errors during development, which can ultimately lead to a more efficient and successful project outcome.

Back-end Framework

We ultimately decided to use Django as the web framework for the Dapper project for several reasons. First, Django is a highly popular and widely-used web framework for Python, with a large community of developers and a vast library of resources available for learning and problem-solving. Second, Django provides many built-in features and functionalities, such as user authentication, database administration, and routing, that can save development time and simplify the implementation of complex features. Finally, Django has a strong emphasis on security and includes many built-in protections against common web vulnerabilities, which can help ensure the application is secure from potential attacks.

Front-end and CSS Framework

After evaluating several options, our development team has decided to use React as the frontend framework for the Dapper project, along with Materialize as the CSS framework. We chose React because it offers a robust set of tools and libraries that can help us to build complex user interfaces quickly and efficiently, while also providing excellent performance and scalability. Additionally, many members of our team have prior experience working with React, which means we can get up and running quickly without significant ramp-up time. We also selected Materialize because it offers a comprehensive set of pre-designed UI components that can help us to streamline the design and styling of our web application. Overall, we believe that this combination of React and Materialize will allow us to build a beautiful, responsive, and highly functional user interface that meets the needs of our users.

Tools

These are a few of the essential tools that the team will be using to implement the application:

1. **MongoDB Compass:** a GUI tool for managing MongoDB databases, which can help developers to visualize the data and manage the schema.
2. **Postman:** a popular API testing tool that can help developers to test and debug their RESTful APIs.
3. **Git:** a popular version control system that can help developers to collaborate and manage code changes effectively.
4. **Visual Studio Code:** a popular integrated development environment (IDE) for Python and JavaScript development, which provides powerful editing features and extensions for improved productivity.
5. **Material-UI:** a popular React UI library that provides a comprehensive set of pre-designed UI components based on Google's Material Design system.
6. **Docker:** a containerization platform that allows developers to package, distribute, and run applications in isolated environments, ensuring consistency across different environments and simplifying deployment workflows.

Continuous integration and Continuous development

Here is an overview of the CI/CD process that will be used during the development of the project:

1. **Development:** Developers write and test their code locally using their preferred development environment, such as PyCharm or Visual Studio Code.
2. **Version Control:** Developers commit their code changes to GitHub, which serves as the central repository for version control. This allows the team to collaborate and keep track of changes made to the codebase.
3. **Continuous Integration:** Jenkins automatically builds and tests the code every time changes are pushed to the GitHub repository. If there are any failures, developers are notified immediately.
4. **Containerization:** Once the code passes the tests, Docker is used to containerize the application and create a portable package that can be deployed to different environments.
5. **Continuous Deployment:** Jenkins deploys the containerized application to Heroku, a cloud platform that allows for easy deployment and scaling of applications.

6. **Automated Testing:** Automated tests are run on the deployed application to ensure that it is functioning as expected.

By following this CI/CD process, the development team can ensure that code changes are thoroughly tested and validated before being deployed to production. This helps to reduce the risk of errors and downtime, and allows for faster iteration and deployment of new features.

Architecture

The Model-View-Controller (MVC) architecture is a popular choice for web development, and it was the architecture that we chose for Dapper. We selected this architecture because it provides a clear separation of concerns between the three main components: the Model, which represents the data and business logic; the View, which presents the data to the user; and the Controller, which handles user input and manages the flow of data between the Model and View (See Figure 9). This separation of concerns makes it easier to develop and maintain code, as changes to one component are less likely to affect the others. Additionally, MVC architecture is highly modular, allowing us to reuse components across the application and scale the application more easily.

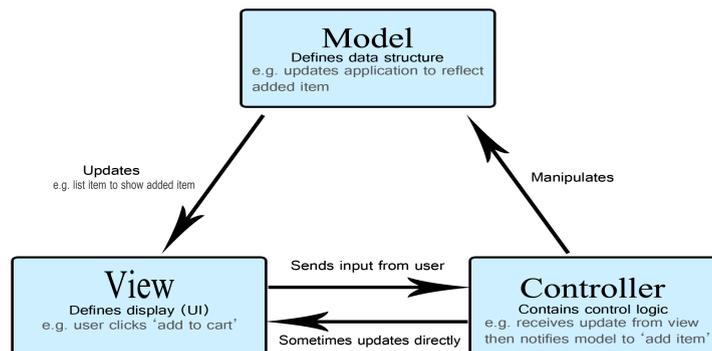


Figure 9: Model-View-Controller (MVC) Architecture [1]

Plan for Next Semester

During the next semester, we plan to focus on the following key activities to successfully complete the Dapper application project:

- **Creation of a Web Application:** We will begin by building a web application that meets the requirements we identified in the previous semester. This will involve designing the database architecture, developing the backend and frontend components, integrating the required frameworks, and ensuring that the platform is user-friendly, easy to navigate, and incorporates features such as virtual try-on, augmented reality, and customization options. We will also ensure that the platform integrates with existing systems such as payment gateways and inventory management systems to ensure a seamless shopping experience for customers and brands.
- **Preparing a Presentation:** We will prepare a comprehensive presentation of our product that will be presented in a way that meets the expectations of judges and potential investors. The presentation will highlight the features of the Dapper application, including its user-friendliness, sustainability, and personalized recommendations. We will also showcase the benefits of collaborating with stylists who will curate outfits based on the customer's preferences, providing a unique and personalized shopping experience.
- **Collaboration with Stylists:** We will reach out to stylists to discuss potential collaborations and partnerships. This will involve identifying stylists whose style and approach align with our sustainability goals and inviting them to curate outfits on our platform. We will also discuss how we can work together to improve the sustainability of the fashion industry and promote ethical fashion practices.
- **Logistics Management:** We will consider all logistics involved after the app delivery to ensure that the platform runs smoothly and efficiently. This will involve monitoring and managing inventory, shipping and delivery, and customer support. We will also work on developing a sustainable and environmentally friendly packaging system to minimize waste and reduce our carbon footprint.

By focusing on these activities, we aim to create a successful and sustainable platform that meets the needs of customers, stylists, and the environment. We believe that by implementing these strategies, we can develop a competitive edge and attract potential investors to support our vision of ethical and sustainable fashion

To make sure that the final product can be tested and that the product can be thoroughly tested for usability, functionality, and performance, and can meet the desired quality standards, the team will stick to several design decisions:

- **Modular Design:** The application's design should be modular, meaning it should be broken down into smaller components that can be tested independently. This will make it easier to identify and isolate defects, allowing for quicker and more efficient testing.

- **User-Centered Design:** The design should be based on a user-centered approach, with an emphasis on user experience and user feedback. This will ensure that the platform is user-friendly and meets the needs and preferences of its intended audience.
- **Prototyping:** The team should create prototypes of the application early on to test the usability and functionality of different features. This will enable the team to identify and fix issues early in the development process, saving time and resources in the long run.
- **Testing Frameworks:** The team should choose appropriate testing frameworks and tools that can help automate testing and ensure that the application meets the desired quality standards. This will reduce manual testing efforts and increase the accuracy and reliability of testing.
- **Performance Optimization:** The design should incorporate performance optimization techniques to ensure that the application is scalable and performs well under different user loads. This will prevent any potential bottlenecks and ensure that the platform can handle a large number of users without compromising its performance.

Impact on Society and the Environment

The Dapper application has the potential to both positively and negatively impact both society and the environment. On the positive side, the application could encourage people to experiment with their fashion choices, leading to increased self-expression and confidence. Additionally, the service provides a convenient way for users to update their wardrobe without contributing to fast fashion's detrimental impact on the environment. By having a stylist curate outfits based on personal preferences, users can avoid the waste and environmental damage associated with shopping for clothing they may never wear.

However, the service's manufacturing and distribution processes could have negative environmental impacts. The production of clothing requires a significant amount of energy, water, and other resources. Additionally, the transportation of the clothing to the user's residence every month could contribute to carbon emissions and other pollutants. The disposal/recycling of the clothing could also pose challenges, as many textiles end up in landfills and take years to decompose.

In terms of safety and risk, there are potential concerns related to user privacy and security. As a subscription-based service that requires personal information, there is a risk of data breaches or misuse of user data. Additionally, there could be physical safety concerns related to the clothing, such as poor quality or sizing issues.

The societal benefits of the Dapper application could include increased accessibility to fashion expertise and personalized styling, potentially benefiting those who may not have access to these services otherwise. The economic benefits could include supporting small businesses and stylists who work with the company.

Overall, it is essential for the Dapper team to consider the full lifecycle of the product and strive to minimize negative environmental impacts while maximizing societal benefits. Careful consideration of the sourcing of materials, manufacturing processes, transportation, and disposal/recycling will be critical to ensuring the application's environmental sustainability. Additionally, implementing robust privacy and security measures will be essential to building trust with users and avoiding potential risks.

To address the potential negative impacts on the environment, the Dapper team could explore using sustainable and eco-friendly materials for the clothing. They could also consider implementing a recycling program for the clothing, encouraging users to send back unwanted items to be repurposed or recycled.

In terms of transportation, the team could explore using electric or hybrid vehicles for delivery, or even implementing a system for users to drop off the clothing at designated locations to reduce carbon emissions associated with transportation.

The team could also prioritize working with stylists and suppliers who prioritize ethical and fair labor practices, promoting social responsibility and ethical business practices.

Finally, the Dapper team could explore ways to make the service more accessible to individuals of different income levels, potentially partnering with community organizations or offering sliding-scale subscription pricing.

By prioritizing sustainability, social responsibility, and accessibility, the Dapper team could make a positive impact on both the environment and society while providing a valuable service to users.

Report on teamwork

Individual Contribution

Zakaria Bouhemhem: Was responsible for conducting research on the fashion industry and market analysis. They also contributed to creating the BPMN and UML diagrams and were involved in researching technologies for the web application. Their meticulous attention to detail and strong analytical skills were evident in their contributions, and they consistently met deadlines.

Amine Benakmoum: Was responsible for creating the User Experience (UX) on Figma and contributed to the research on competitors and similar apps. They also contributed to creating the BPMN and UML diagrams. Their design expertise and keen attention to detail were apparent in their work, and they consistently delivered on time.

Ryad Ammar: Was responsible for researching technologies for the web application, including backend and frontend frameworks, databases, and other technologies. They demonstrated strong technical skills and attention to detail, and were able to propose several innovative solutions to address the project's requirements.

Collaboration:

The team was able to collaborate effectively and communicate well throughout the project. Each member was able to contribute their expertise and ideas to the project, and there was a good balance of tasks and responsibilities among the team members. The team members were respectful and responsive to each other's feedback and suggestions.

Difficulties and Plan:

One difficulty that the team encountered was coordinating schedules due to individual commitments and workload from other courses. To address this, the team plans to set up a regular meeting schedule that works for everyone and prioritize attendance. The team also plans to establish clear communication channels and protocols to ensure that everyone is up-to-date on project progress and has a clear understanding of their roles and responsibilities.

Another difficulty that the team encountered was the initial phase of the project, where team members had varying levels of familiarity with the project requirements and goals. To address this, the team plans to establish a system for sharing and consolidating information on a regular basis. This could include bi-weekly progress reports and team updates, as well as clear documentation of project milestones and deadlines.

Additionally, the team plans to establish a system for tracking and assigning tasks to ensure that everyone is contributing equally and that deadlines are met. This could include using project management tools such as Gantt chart, where team members can assign tasks, track progress, and communicate with each other.

Overall, the team is committed to working collaboratively and supporting each other to ensure the success of the project

Conclusion

In this semester, our team made significant progress towards developing Dapper, a subscription-based application that delivers personalized outfits based on personal preference. We were able to establish a solid foundation for the project by creating a Business Process Model and Notation (BPMN), Unified Modeling Language (UML), User Experience (UX) on Figma, and conducting extensive research on the fashion industry, market analysis, and competitor analysis. We also researched technologies for the web application, including backend, frontend, databases, and frameworks, which helped us to make informed decisions regarding the project's development.

Moving forward, our team plans to implement the app and meet the project requirements, which include delivering personalized outfits based on personal preferences, current season, preferred colors, size, and subscription level. We also plan to contact our first collaborators, which will include fashion stylists and clothing manufacturers, to provide high-quality outfits to our users.

Through this project, we have gained valuable skills in startup development, project management, market research, and teamwork. We learned the importance of effective communication, regular meetings, and establishing clear goals and responsibilities. We also learned the importance of conducting thorough research and analysis to make informed decisions and adapt to changing market trends.

Overall, we are excited to continue working on Dapper and believe that our application has the potential to revolutionize the fashion industry by providing personalized outfits and contributing to a more sustainable and inclusive fashion industry.

References

1. Mozilla Developer Network. (n.d.). Model-View-Controller (MVC). Retrieved April 11, 2023, from <https://developer.mozilla.org/en-US/docs/Glossary/MVC>
2. N. J. Mitra and T. A. Deshpande, "Comparative Analysis of NoSQL and SQL Databases: A Survey," International Journal of Computer Applications, vol. 85, no. 3, pp. 1-7, Feb. 2014. Available: <https://www.ijcaonline.org/archives/volume85/number3/14868-2009> [Accessed: Apr. 01, 2023].
3. S. S. Srinivas and S. S. Iyengar, "A Comparative Study on SQL vs. NoSQL Databases," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 6, no. 2, pp. 501-508, Feb. 2016. Available: <https://www.semanticscholar.org> [Accessed: Apr. 01, 2023].
4. M. A. Abdur Rahman, M. A. Ahad, and M. A. Azim, "A Comparative Study of NoSQL and SQL Databases," 2021 International Conference on Electronics, Information, and Communication (ICEIC), 2021, pp. 1-5, doi: 10.1109/ICEIC52636.2021.9355017.
5. D. Wu and D. Feng, "Comparative Study of NoSQL and SQL Databases," 2022 International Conference on Computer Science and Software Engineering (CSSE2022), 2022, pp. 36-40, doi: 10.1145/3600043.3600047.
6. MongoDB official website: [Online]. Available: <https://www.mongodb.com/>. [Accessed: Apr. 06, 2023].
7. MongoDB documentation: [Online]. Available: <https://docs.mongodb.com/>. [Accessed: Apr. 06, 2023].
8. N. Janetakis, "Why MongoDB Is the Best NoSQL Database for Your Next Project," Nick Janetakis Blog, Sep. 20, 2021. [Online]. Available: <https://nickjanetakis.com/blog/why-mongodb-is-the-best-nosql-database-for-your-next-project>. [Accessed: Apr. 01, 2023].