REFLECT

NEVER A DULL MOMENT WHEN PHYSICAL TURNS VIRTUAL

كلية الجبيل الصناعية
JUBAIL INDUSTRIAL COLLEGE

FINAL REPORT

Supervised by : Dr. Mariam ALQasab

Group name : REFLECT

Group members :

Name
Salma Alatwah
Manar Alali
Raneem Alshehri
Remas Almutairi

TABLE OF CONTENTS

PROJECT DESCRIPTION

PROBLEM DESCRIPTION:

Modern businesses today, have seen a drastic shift with the evolution of technology. Where even the simplest of things are digital right now. And, shopping for clothes is no exception. With over 2.14 billion online shoppers worldwide (Fokina, 2023), it is hard for everyone to be satisfied with their products or have an easy shopping experience. Because the medium in which we view these products is through a screen, it is hard to judge said item and take the risk of buying it. And if we do, the expected product could come with problems, including but not limited to the size guide can be hard to follow (or sizes aren't accurate), the product was not exactly as advertised on the app, or the customer did not like the outfit on them. Noting that 38% of customers return clothing items due to them not fitting right (Statista, 2023). Because of these reasons, return rates are high and customer satisfaction may take a hit. Leading to negative impacts on revenue and the overall standing of the company. Furthermore, the returns might cost businesses 66% of the item's original price (Franklin, 2022). Since, transport, inspection, and labor play major roles in revenue loss. Due to these alarming numbers, both consumer and business may not achieve their expected goals.

SOLUTION:

To combat these problems and to make sure your company earns more, we introduce: 'REFLECT'! REFLECT is software that utilizes elements of 'Augmented Reality' (AR) and projects a real-time simulation of a digital object onto the desired person. Since we're a team of developers passionate about merging the worlds of fashion and technology, REFLECT is our solution for innovative software that could be a key factor in improving the fashion industry. The software works by capturing an image of the consumer and overlaying them with three-dimensional digital clothing that could be examined in real-time. Instead of the item being static like most images in online apps, the item will move in sync with the person. This is usually done via mobile phones, as they are extremely ubiquitous, but they may also work on other devices (McDowell, 2021). This will make for an app that is interactive as well as beneficial. Our company's slogan: "Never a dull moment when physical turns virtual," is the essence that we want to extend to users. Where the normal shopping experience could become something more thrilling by way of AR. REFLECT could solve many underlying issues by showing users how a product would fit and determining if the style is right. Research done by "Shopify" stated that the use of AR can reduce returns by up to 40% and increase conversions by 97% (McDowell, 2021). We guarantee that implementing REFLECT will have a positive effect on sales, and more customers would want to try immersive technology while shopping online. We are excited about providing you with further details about REFLECT as we continue moving forward with the project. And, for any inquiries, please get in touch with any of the contacts provided above.

SOFTWARE AND HARDWARE REQUIREMENTS

- The software and hardware requirements that we need for **REFLECT** software in the table below.

SOFTWARE REQUIREMENTS	HARDWARE REQUIREMENTS
• C++ / Java / JavaScript • MySQL • EasyAR • Adobe Aero • ARKit • ARCore	• GPU (Nvidia GeForce GTX 1660 Super) • CPU with minimum clock speed of 3.5GHz • SSD with at least 512GB Memory with 16GB of RAM • Device - mobile phone preferred

Note: We've provided multiple softwares for AR SDK's, but we'll primarily use Adobe Aero.

BUDGET ESTIMATION

- The estimated budget for REFLECT software .

FOR HARDWARE AND SOFTWARE REQUIREMENTS	$3100
FOR DEVELOPMENT	$9900
FOR TESTING	$7000
TOTAL COST	$20000

SCHEDULE OF THE PROJECT

01
SPECIFICATIONS :
4 WEEKS

02
DESIGN:
8 WEEKS

03
DEVELOPMENT:
24 WEEKS

04
TESTING:
16 WEEKS

IN TOTAL : 52 WEEKS

DEVELOPMENT MODELS

For this software, we have chosen the "Agile Model". This decision was made after considering the nature of the software, the requirements needed, and most importantly the breakdown of the tasks shown on page 22. The agile model will help our team in maintaining a flexible development phase. As our software deals with elements of augmented reality, the chances of mistakes are high. So trial and error when developing is a given. Thus, implementing the agile methodology will work in our favor for REFLECT.

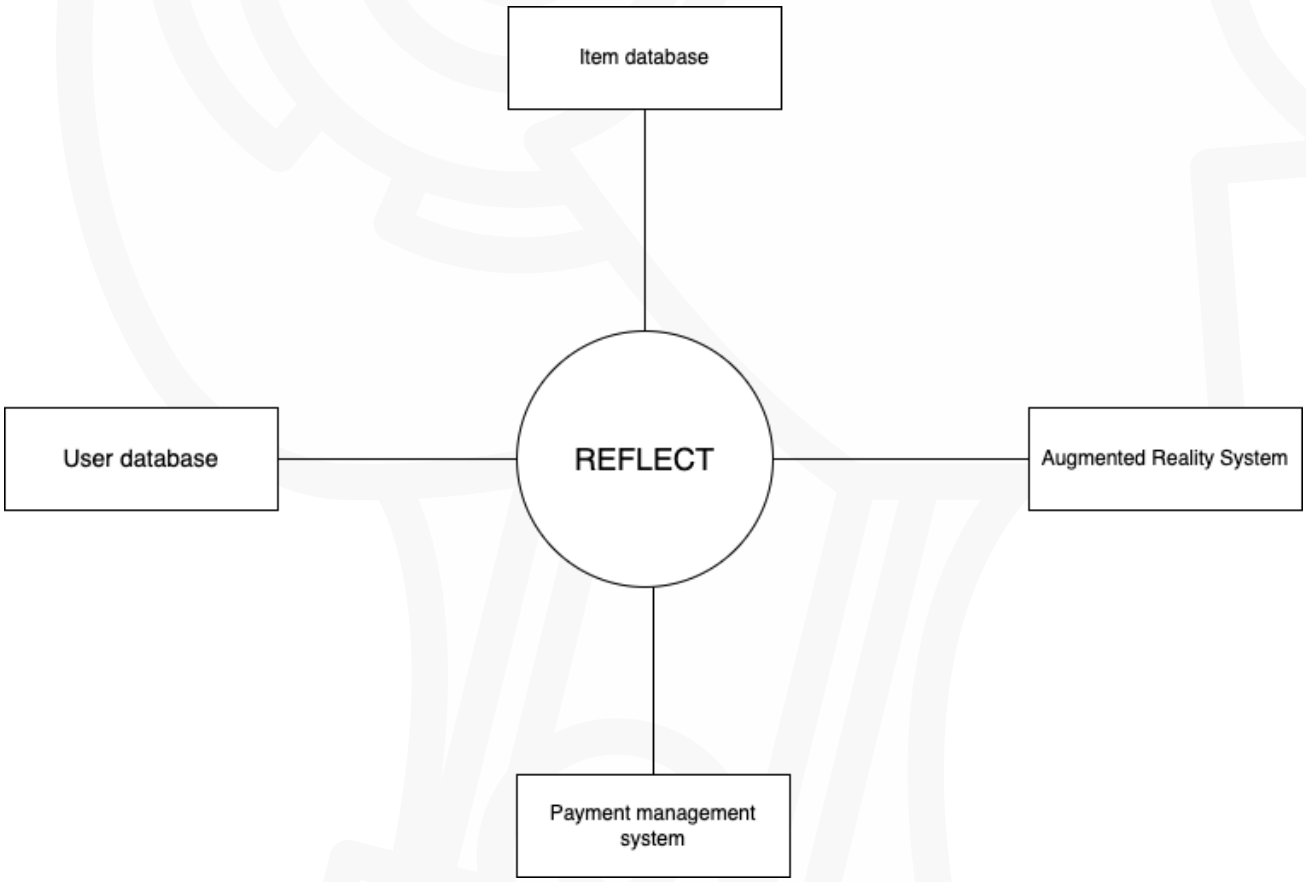
REQUIREMENTS

USER REQUIREMENTS	SYSTEM REQUIREMENTS
1.REFLECT users should have an intuitive and user-friendly interface.	**Functional Requirements:** 1.1 The interface should be clear and simple for the users to interact with the AR system and its features without requiring technical knowledge. 1.2 The system shall offer feedback to users in order to verify that their actions were correctly done or to warn them of any issues that may have occurred. **Non-Functional Requirements:** 1.3 The interface should be compatible with a variety of devices and easily accessible to all users. 1.4 The interface should be responsive with a minimal delay between user inputs and system responses.
2. REFLECT shall have a personalized recommendations system	**Functional Requirements:** 2.1 The system should be able to personalize suggestions based on user interests, purchase history, and browsing activity. 2.2 The system should have a recommendation algorithm. 2.3 The system should be able to provide real-time updates of suggestions depending on user activities and changes. **Non-Functional Requirements:** 2.4 The update response time should not exceed more than 1.5 seconds.
3. REFLECT should provide social sharing option to users	**Functional Requirements:** 3.1 The system should allow users to share their virtual try-on experiences via social media. 3.2 The social sharing feature must be capable of managing an enormous number of users and content items. **Non-Functional Requirements:** 3.3 This functionality should be available at all time (24 hours).

USER REQUIREMENTS	SYSTEM REQUIREMENTS
4. REFLECT users shall visually see an output of themselves overlayed with the selected item.	**Functional Requirements:** 4.1 REFLECT shall take in the user's approximate physical measurements using the AR algorithm implemented in the software. 4.2 The user shall choose a clothing item from the database to be overlayed on them. 4.3 The software should have a clear distinction between the target (user) and any surrounding environment, to make the item only be overlayed on the user. 4.4 The software shall choose the best-fitting item according to the user's measurements to overlay it on the user. **Non-Functional Requirements:** 4.5. The output should be shown to the user within 30 seconds. 4.6 Downtime for REFLECT with normal user request traffic shall not exceed 5 seconds.
5. REFLECT users shall pay for their clothes without worrying about their information being compromised.	**Functional Requirements:** 5.1 All financial dealings will be managed and tracked. **Non-Functional Requirements:** 5.2 The app shall have some kind of rudimentary verification method so that no one else can use it to make purchases in the user's name. 5.3 Sensitive information (such as credit card data) sent to the server by Web clients is usually encrypted using secure socket layers (SSL) or another encryption method.
6. REFLECT users should be able to rate and review each product to help other customers make wise purchases.	**Functional Requirements:** 6.1 Users should be able to quickly identify products with excellent reviews or specific features using the software's search and filter capabilities. 6.2 The software should provide reporting and moderation features to make sure the user's reviews are acceptable. **Non-Functional Requirements:** 6.3 The software should be able to handle large amounts of user reviews and ratings without affecting its performance.

SYSTEM MODELS & ARCHITECTURAL DESIGN

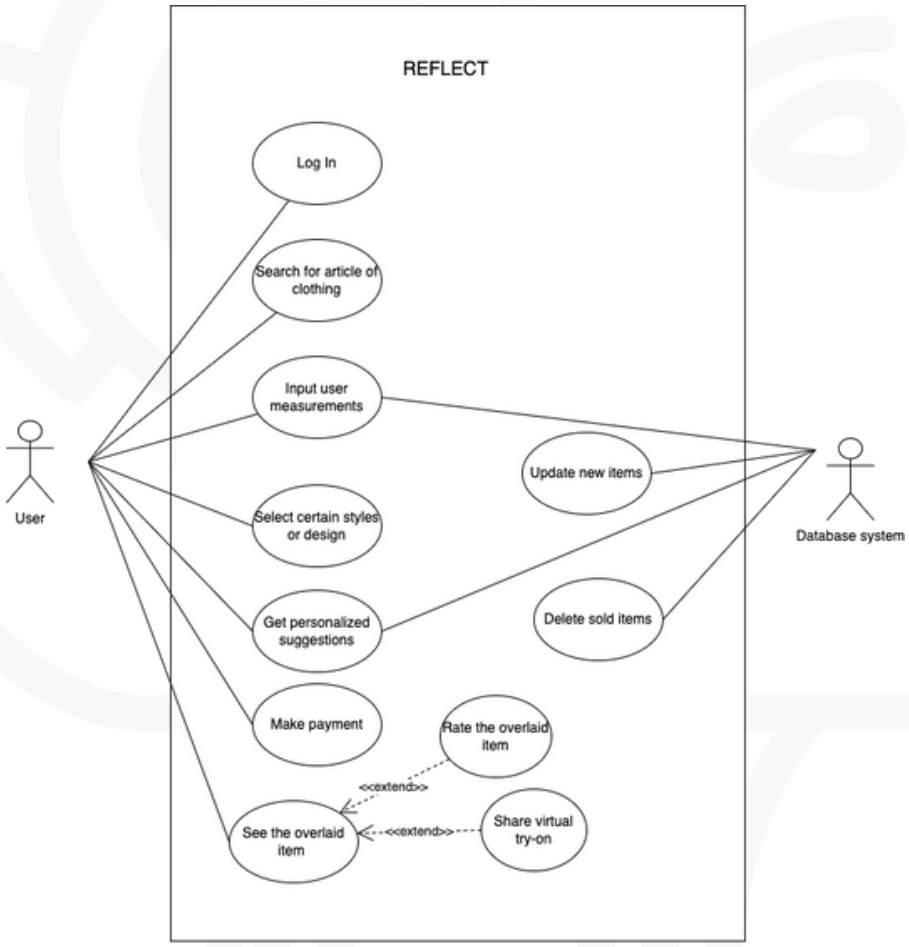
CONTEXT MODEL:



Reflect will use the following systems to its benefit:

- **Item Database:** which will store all available clothing items for REFLECT. It will maintain and store the database depending on consumer needs.
- **User Database:** for storing user information such as usernames and passwords. This database willl also monitor user preferences and update its suggestions for the user.
- **Payment Management System:** this will help the user make secure payments reliably and efficiently.
- **Augmented Reality System:** this system will be composed of AR SDKs, and will tremendously help in REFLECT's functioning as a system.

USE CASE DIAGRAM- INVOLVING THE ROLE OF THE USER AND DATABASE SYSTEM:



Reflect user(primary actor) is allowed to:

- **Log in:** The user will have an interface that will allow them to enter their information (username, password).
- **Search for an article of clothing:** Items REFLECT can be searched for by the user, and they will be looked for in the 'item database'.
- **Input user measurements:** The interface will allow the user to input their measurements, either the user enters it or the AR algorithm will approximately guess.
- **Select certain styles or designs:** The user can select a certain style to shop for, as the system divides the items into categories.
- **Get personalized suggestions:** Based on user interactions, REFLECT will offer some suggestions.

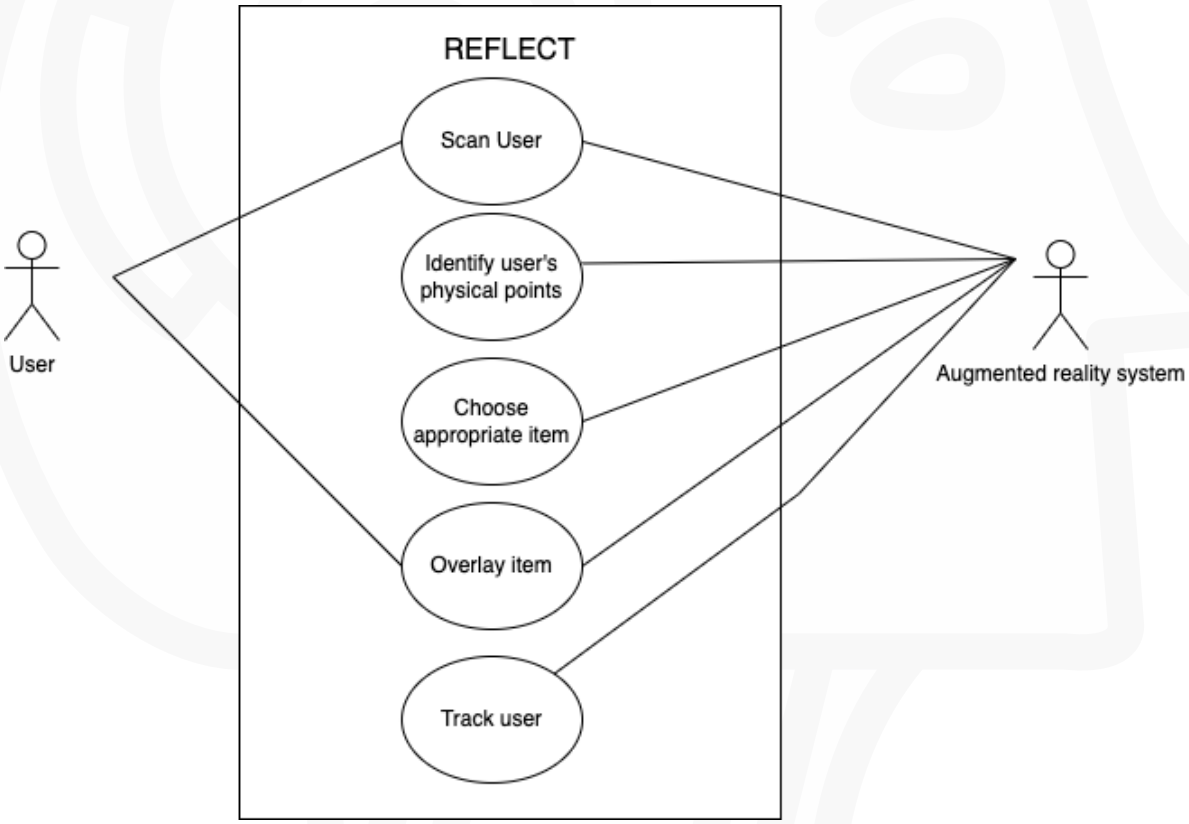
Reflect user(primary actor) is allowed to:

- **Make payment:** users will pay for orders securely
- **See the overlaid item:** The user will have an output visible to them. And they can also rate the item or share the output

The database system(secondary actor) will do the following:

- **Upload new items:** It will add and store new items.
- **Delete sold items:** As soon as an item is sold, the database will delete it. And will add it back when it is back in stock.
- **Get personalized suggestions:** The database will make use of the user's likes and dislikes.
- **Input user measurements:** The database will store the entered measurements. This will help the AR system in evaluating the correct item for the user.

USE CASE DIAGRAM- INVOLVING THE ROLE OF USER AND AUGMENTED REALITY SYSTEM:



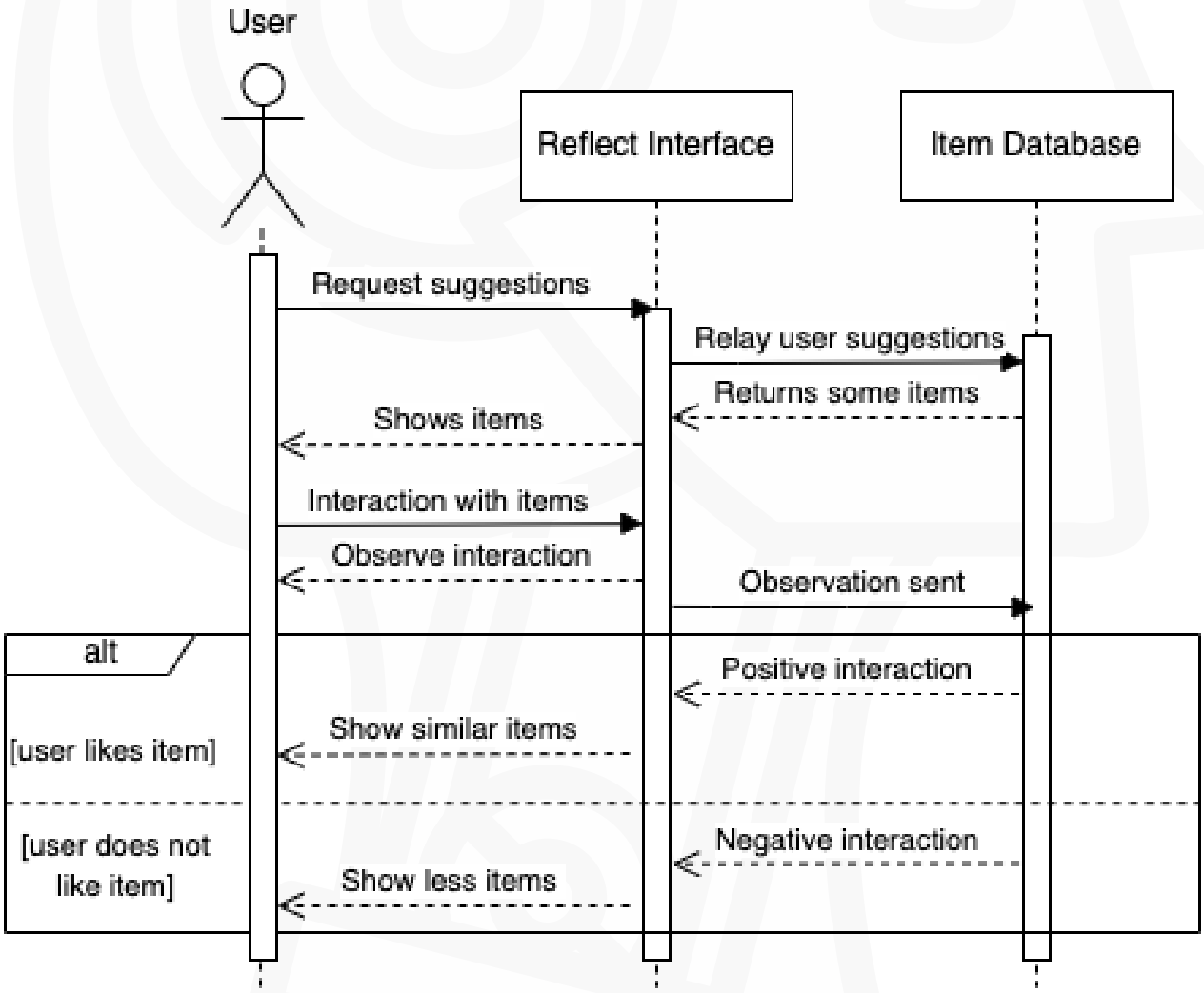
REFLECT user(primary actor) is allowed to:

- **Scan user:** The user will stand in front of a camera to scan their physique.
- **Overlay item:** The user will visually see the item on the screen.

Augmented reality system(secondary actor) is allowed to:

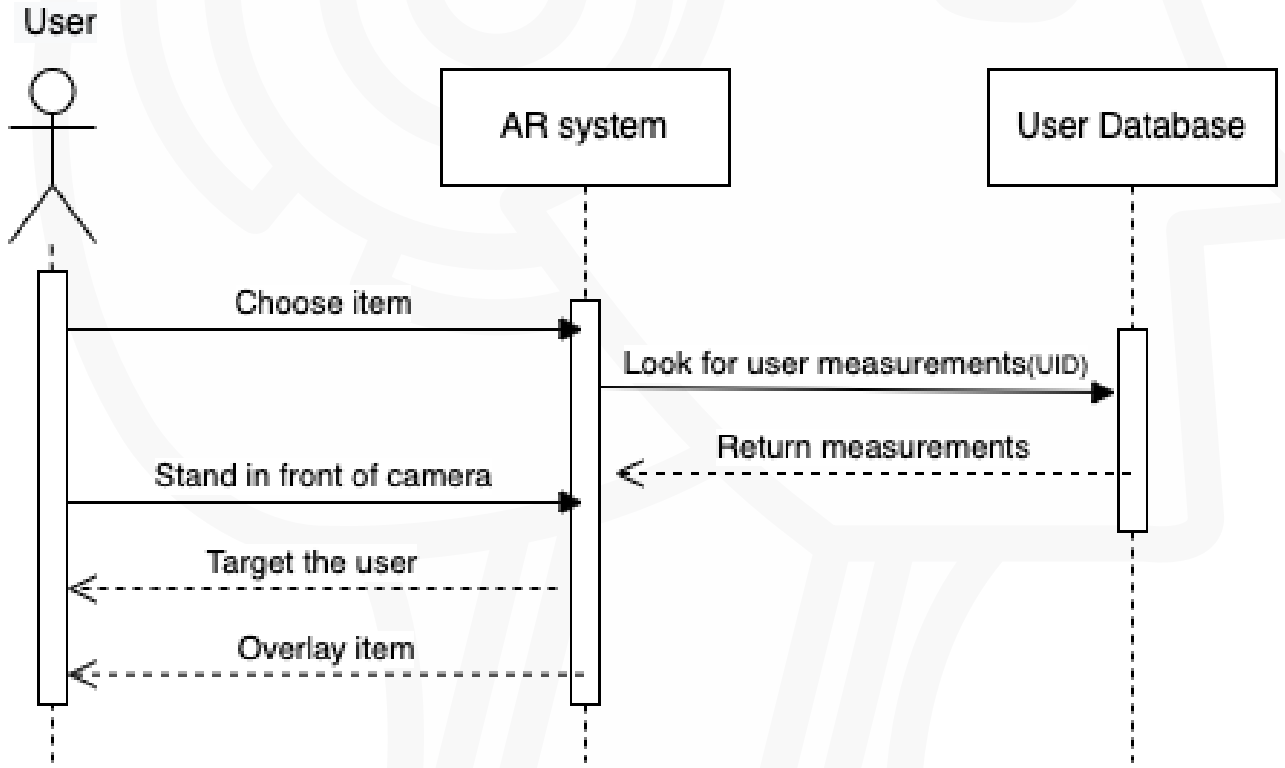
- **Scan user:** The system will capture the scanned image of the user and store it.
- **Identify the user's physical points:** Certain points such as the head, waist, hips, and legs are marked to accurately put the item on top of the user.
- **Choose the appropriate item:** Based on the user's measurements and preferences, the system will decide on the best-fitting item.
- **Overlay item:** The system will overlay the item on the user as accurately as possible.
- **Track user:** The User will be the target of the system, so when the user moves the item will move with them.

SEQUENCE DIAGRAM FOR 'GET PERSONALIZED SUGGESTIONS'



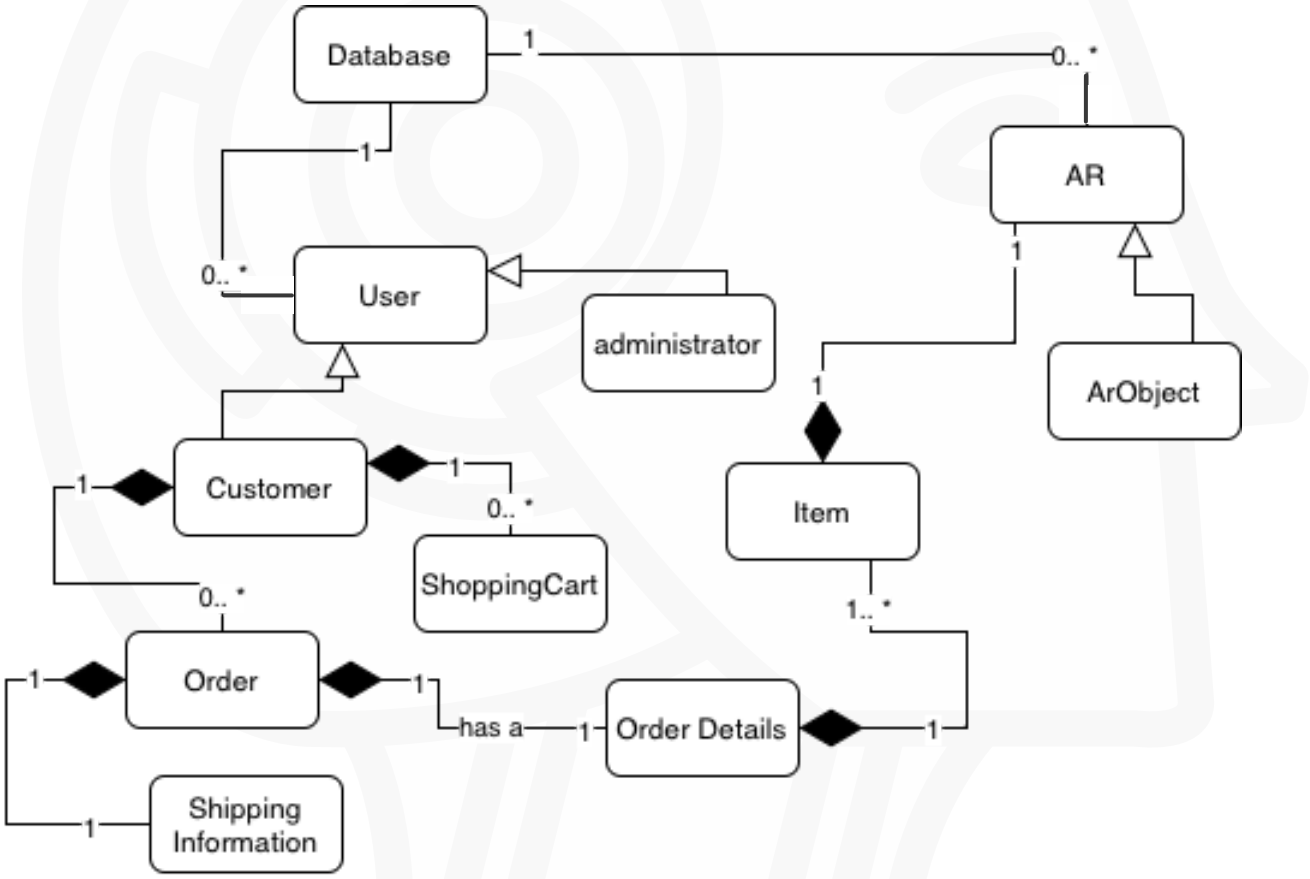
1. The user will request the system for personalized suggestions.
2. The system will check its database for the user's history, likes, and past orders.
3. The database will return a few suggestions to the user.
4. The user will interact with these items, they may add them to the cart, like them, swipe, or dislike the item.
5. The system will observe the user's behavior and store it in the database.
6. If the user showed a positive behavior i.e. liked the item, then the system will recommend similar clothes to the user
7. If the user exhibited negative behavior, then the system will make sure not to recommend such items in the future.

SEQUENCE DIAGRAM FOR 'OVERLAY ITEM':



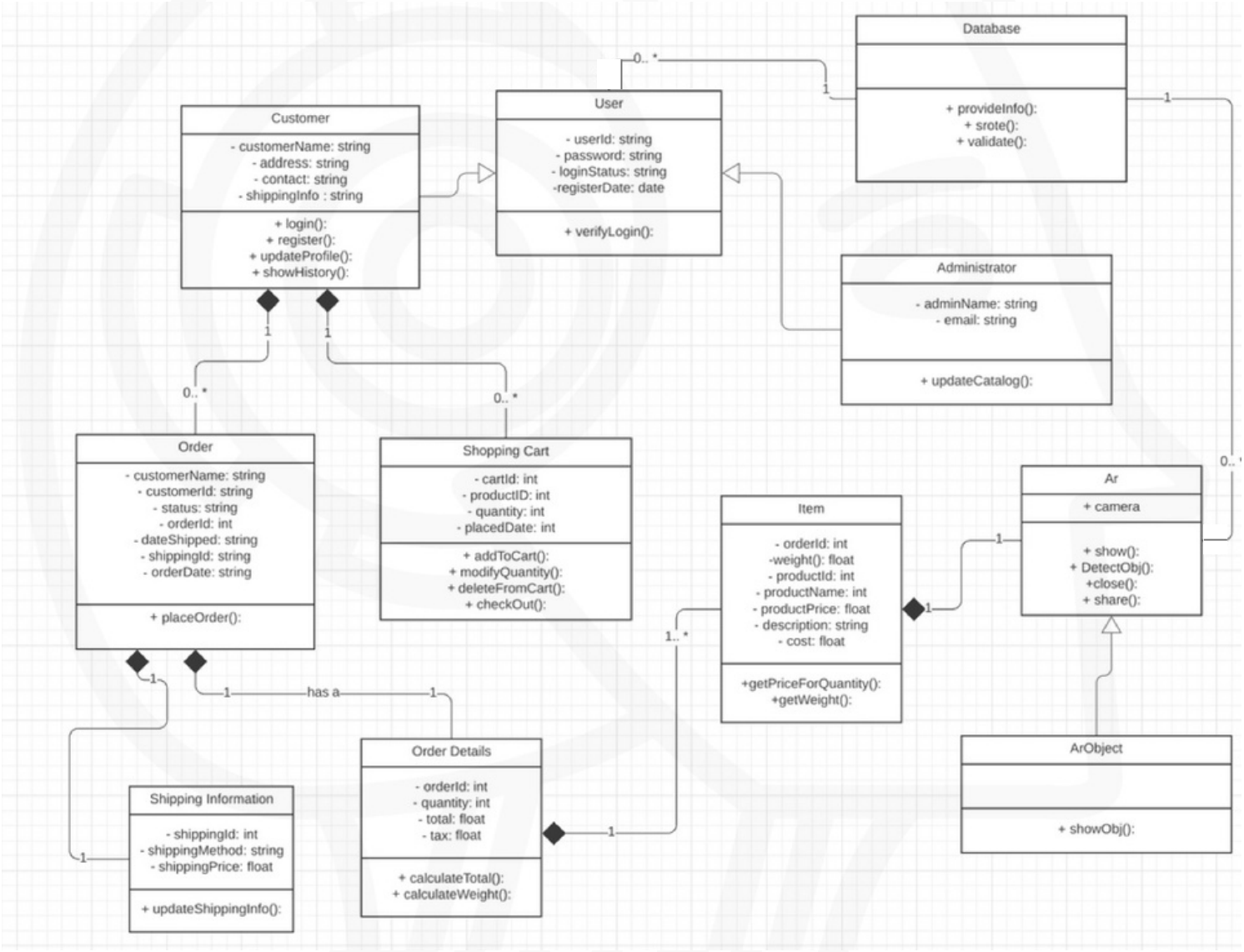
1. The user will choose an item to be overlaid with.
2. The AR system shall find the user's measurements by the User id (Uid).
3. It will take the measurements and choose the best item for the user.
4. The user then shall stand in front of the screen to be captured by the camera.
5. The system will target the user with AR technology and marks them as the target.
6. Finally, the system will output the item on top of the user.

HIGH LEVEL CLASS DIAGRAM OF REFLECT SYSTEM:



1. **Database class** will save all the users all information about users (admins, customers), and their access actions.
2. **User class** will control how users access the system and their accounts.
3. **Administrator class** will provide the admins' roles and define the entire system actions that they are allowed to perform.
4. **Customer class** will contain information on the users and define how they can interact with the system, including creating an account, logging in, seeing previous purchases, adding items to their cart, and recording their preferences and order history.
5. **Order class** will keep the customer updated on the order's progress throughout the process, from delivery to pick-up.
6. **The shopping cart class** will contain information about the items in the cart that are saved temporarily, such as the name, quantity, and price.
7. **The shipping information class** will contain information about the customer's shipping address, the method of shipping selected, the expected delivery date, and tracking information for the customer.
8. **Order details class** will contain information about the items that are ordered.
9. **The item class** will manage the item's availability and contain details about the item, like its name, description, and price.
10. **AR class** will monitor the position and orientation of the virtual object concerning the surrounding real-world environment.
11. **AR object class** will contain information about the object, such as its size, shape, and texture.

DETAILED CLASS DIAGRAM OF REFLECT SYSTEM:



1. Database class:

The Database class is responsible for managing the user's account, validating their information, and ensuring that any updates to their information are reflected in the database. Additionally, it is responsible for storing the contents of the Ar class for future use.

2. User class:

The User class stores the user's basic information, including their user ID, password, login status, and registration date. Users can log in to the system using their user ID and password, and their identity is verified by the verifyLogin() method, which calls on the Database class to confirm their information is correct and stored in the database.

3. Customer and Administrator classes:

Both the Customer and Administrator classes inherit from the User class. The Customer class contains the specific details of a customer, such as their name, address, contact information, and shipping details. Customers have access to methods such as login(), register(), updateProfile(), and showHistory().

On the other hand, the Administrator class includes the name of the administrator and a method of contact via email. Administrators have the privilege to update the catalog of the system by updateCatalog() method.

DETAILED CLASS DIAGRAM OF REFLECT SYSTEM:

4. Order class:

The Order class includes several attributes related to an order, such as the customer's name and ID, the order status (e.g. placed, shipped), the order ID, the date shipped, the shipping ID, and the order date. These attributes provide basic information about the order. Additionally, the placeOrder() method can be used by a customer to place multiple orders.

5. Shopping Cart:

The Shopping Cart class stores information about a customer's shopping cart, including the cart ID, product ID, quantity of each product, and the date the product was added to the cart. The class provides four methods:

1. addToCart() - adds a product to the customer's shopping cart.

2. modifyQuantity() - allows the customer to update the quantity of a product in their cart. 3. deleteFromCart() - allows the customer to remove a product from their shopping cart. 4. checkOut() - directs the customer to the payment page to complete their purchase.

6. Class for shipping information:

The Shipping Information class depends on the Order class since it offers all the information needed for shipment, such as the shipping ID and cost. Customers may select their preferred delivery method, and this class gives them the ability to use the updateShippingInfo() method to change their shipping information.

7. Order details class:

this class will include the attributes: order ID, quantity, total, and tax they describe the detailed information about the order, and the methods that allow the system to manage the order such as

calc total() will calculate the total cost of a customer's order, including any applicable taxes or shipping costs.

calc weight() will calculates the total weight for that quantity of the item.

8. Item class:

this class will include those attributes order ID, weight, product is, product name, product price, description, and cost which represent the characteristics of the item and the methods that allow the system to manage the item such as getting a price for quantity to() will calculate the total cost of a certain quantity of an item. get weight () will display the weight of the item to the customer**.**

9. Ar and ArObject class:

The Ar class is responsible for providing the necessary augmented reality technology for the system. Its primary attribute is the camera, which allows it to implement the following methods:

1. show() - displays the product in the picture taken.

2. DetectObj() - detects human body patterns to apply the technology.

3. close() - shuts down the camera and deletes any pictures taken.

4. share() - enables users to share pictures of products taken using various social media platforms. Information related to this functionality is stored in the Database class.

Additionally, the ArObject class is used to specify the properties of the augmented reality objects used in the system. The ArObject class is a child class that specializes in AR object recognition and manipulation of real- world elements, such as the human body, to create a virtual 3D model for product display. It also helps place the AR object in the correct location. This class includes only one method:

1. showObj() - recognizes and positions the object (product) in its appropriate location.

RISKS

1 - ALGORITHM ISN'T SUITABLE FOR EVERY PLATFORM

RISK DETAILS:

Developing an appropriate algorithm that works across all platforms will be hard. As we have to consider the different operating systems and capacities of different devices and browsers. This risk will pose a "Technology Risk" for the development of REFLECT.

RISK ANALYSIS:

Since we want to make sure that the software is accessible to every possible user, we have to work across all platforms and implement the software. The possibility of our algorithm not working for a certain platform will be moderate and its effects serious.

RISK MANAGEMENT:

Risk Avoidance Strategies:
Develop a solid architectural plan, such that it encompasses every platform. Use an SDK that is inclusive for every platform.

Impact Minimization:
Replace unsuitable parts of the algorithm with bought-in components of known-reliability.

RISK MONITORING:

Delivering software late is an indicator of possible risks. So we must test the software after major milestones to ensure that it runs properly on every device and platform.

2- SUDDEN GROWTH IN REQUIREMENTS

RISK DETAILS:

Augmented Reality is an area that experiences rapid change and growth. So, continuous development is inevitable. Also, changing customer requirements have to be met, because the application type is prone to change. This is a "Requirement Risk" for our team.

RISK ANALYSIS:

Growing requirements will lead to major rework. Hence, the probability of the risk is high and its effects are serious.

RISK MANAGEMENT:

Risk Avoidance Strategies:
Obtain traceability information to observe the requirements and to maximize information hiding in the design.
Keep the requirements as clear and updated as possible, so newer requirements can be adjusted accordingly.

Impact Minimization:
Arrange for regular meetings to asses changing requirements. Have great communication between team and client.

RISK MONITORING:

Delivering software late is an indicator of possible risks. So we must test the software after major milestones to ensure that it runs properly on every device and platform.

3 - FINANCIAL SHORTAGE

RISK DETAILS:

It's one of the most significant risks that can affect the delivery of our project. The whole software and its future depends on the budget, The risk is considered an "Organizational Risk".

RISK ANALYSIS:

Since our software works with Augmented reality technology it can cost a lot so the chances that the risk will happen is moderate which will lead to catastrophic effects.
If we don't consider a wise cost allocation we won't be able to complete our project.
Lack or unplanned budget can cause failure to reach the main goals of the project which will cause it to be undeveloped in the end.

RISK MANAGEMENT:

To avoid the chances of the risk:
Do full research on previous similar projects.
Take into account all the possibilities that the software could be modified and updated, as well as the associated costs.
To minimize the impact of the risk:
Use an incremental approach to reduce expenses so that each step is affordable and can be modified with the minimum amount of expenditure.

RISK MONITORING:

Plan a regular meeting that includes all the details about the budget, discuss any growth with the manager, and keep track of all the financial risks.

4-NOT MEETING DEADLINES

RISK DETAILS:

The risk that tasks in your project will take longer than expected. The type of risk is "Estimation risk".

RISK ANALYSIS:

The consequences of not meeting deadlines are significant and can often be very costly. The possibility of not meeting deadlines in our project is high and the effects serious.

RISK MANAGEMENT:

Risk Avoidance Strategies:
Set reminders for all your tasks. Create a daily planner.
Give each task a time planner.
To minimize the impact of risk :
Evaluate the risk, which entails calculating the degree of risk associated with the events noted. Measures, procedures, and controls are included in risk assessments in order to lessen the impact of risk. The development team should always have a plan, and try to allow buffer time should unexpected risks occur. Lastly, breaking down the project into mini-deadlines will help the team in having a clear goal and working towards it.

RISK MONITORING:

You must identify and track risks as part of risk monitoring, and keep an eye out for risk triggers, such as submitting deliverables late to the customer. As well as not keeping up with the deadline.

5-TECHNICAL DIFFICULTIES

RISK DETAILS:

It is a technical risk with AR technology that has an impact on the final product and its functionality. Technical challenges are associated with the software and hardware components that are utilized to link the various pieces of the software system and make it fully functioning.
The type of risk is "tool risk" and "technology risk".
Technological issues might arise in several aspects in our software such as: Limited hardware capabilities
-The capabilities of cameras vary amongst devices. The majority of them only capture 2D photos, which is not ideal for building a 3D hybrid reality.
Object Detection Challenges
-The most difficult issues in object detection are classifying items(such as humans) and identifying their position.

RISK ANALYSIS:

Technological issues are more likely to occur than any other risk, and they may lead to lowering the quality of the finished product and raising project costs. The chances that this risk will arise are high and its effects are serious.

RISK MANAGEMENT:

This risk can be prevented using risk avoidance strategies such as:
Developing the software incrementally.
Utilizing modern technology and being informed about future AR developments.
The consequences of this risk can be minimized using risk-minimizing strategies such as: As the software is developed incrementally, it may be updated at a minimal cost to accommodate new technology.
Providing a quick update in order to address the problem that occurred.

RISK MONITORING:

The following strategies can be used to monitor technical difficulties:
Frequent provision of prototypes for performance testing.
Proper documentation for the risk factors at every level of the development plan.

TASKS AND MILESTONES

TASKS BREAKDOWN:

T1.COLLECT REQUIREMENTS AND CONSTRAINTS:

It is a process that defines and manages the needs and requirements of the stakeholders to meet their demands to ensure project satisfaction.

T2.FINALIZE THE REQUIREMENTS:

We gather all of the requirements in one location after collecting them from the stakeholders, and before choosing a requirement for the project scope, we determine how important it is. Then we review, accept, reject, and sort the requirements we collected.

T3.MODEL THE REQUIREMENTS:

In this task, we construct a representation of the requirements that we collected, activities, and data, and define the connections between them.

T4.ALLOCATE THE ACTIVITIES TO THE TEAM:

Give tasks to the team members while carefully defining each person's role and responsibilities, considering their skills, knowledge, and experience.

T5.DESIGN THE STRUCTURE OF THE SOFTWARE:

The basic architecture and design of REFLECT will be designed in this stage. So we will represent the system graphically or through a document. This will help us understand more about the structure and requirements for RELFECT.

T6.CREATING A PROTOTYPE:

Before any development starts, it's important to provide a prototype of REFLECT to the clients. As the software is complex, so it's crucial that customers see the prototype. This will give our team valuable feedback on how to continue with the project.

T7.CONSTRUCT THE 'AR' ALGORITHM:

Since our software works with augmented reality for clothes, we have to create an algorithm that functions as accurately as it can. While also, making it work across all platforms with no problems. This task will be one of our main priorities during development.

TASKS BREAKDOWN:

T8.DEVELOPING THE SOFTWARE REQUIREMENTS:

This stage will pertain to developing the requirements the customers asked for. We will use the structural design of REFLECT and start developing it alongside the algorithm.

T9.DATABASE COLLECTION AND DEVELOPMENT:

Our team will gather the available database of the company and filter out the data based on how we'll use it. Later, we'll develop a database that contains all items(clothes) and constraints that match the algorithm functionalities. Since the database may be huge this will take some time.

T10.APPLYING AR TECHNOLOGY:

At this stage, we'll use available and known AR software development kits "SDK's" to implement our algorithm with the database made. This is an important part of the development. As it will show how our components will integrate together to use AR technology.

T11.SOFTWARE TESTING AND VALIDATION:

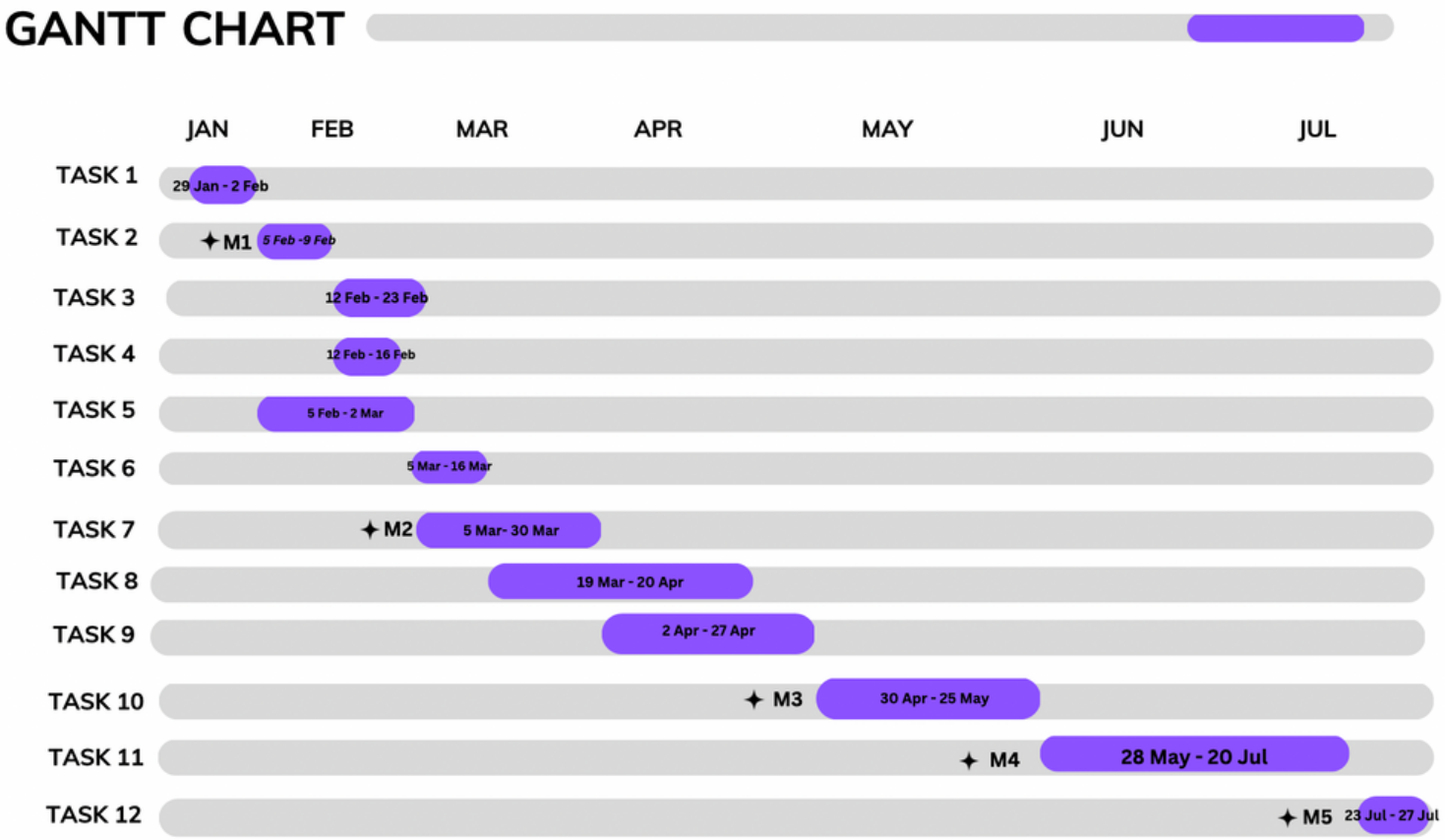
The product development team may conduct testing methods (unit testing, system testing, integration testing, and acceptance testing), and validation to ensure that the performance product meets its acceptance criteria.

T12.FINAL REVIEW OF THE SOFTWARE:

Our team will review the program to identify and address any problems or weaknesses by evaluating the testing methods in the task above, and we'll use those fixes to reduce any gaps in our software.

TASKS	DESCRIPTION	EFFORT (PERSON-WEEKS)	DURATION (WEEKS)	DEPENDENCIES
T1	Collect requirements and constraints from the client and important stakeholders	1	1 week	
T2	Finalize the requirements to see if they're suitable	4	1 week	T1 (M1)
T3	Model the requirements	2	2 weeks	T2
T4	Allocate the activities to the team	1	1 week	T2
T5	Design the structure of the software	3	4 weeks	T1
T6	Creating a prototype	1	2 weeks	T5
T7	Construct a specific algorithm suitable for the company and its platforms	4	4 weeks	T4,T5(M2)
T8	Developing the software requirements	4	5 weeks	T6
T9	Gather the database file containing the clothes and items of the company	2	4 weeks	T7
T10	Applying AR technology	4	4 weeks	T7,T9(M3)
T11	software testing and validation	2	8 weeks	T8,T10(M4)
T12	Final Review of the software	4	1 week	T11(M5)

GANTT CHART


GANTT CHART
JAN
FEB
MAR
APR
MAY
JUN
JUL
TASK 1
29 Jan - 2 Feb
TASK 2
M1
5 Feb -9 Feb
TASK 3
12 Feb - 23 Feb
TASK 4
12 Feb - 16 Feb
TASK 5
5 Feb - 2 Mar
TASK 6
5 Mar - 16 Mar
TASK 7
M2
5 Mar- 30 Mar
TASK 8
19 Mar - 20 Apr
TASK 9
2 Apr - 27 Apr
TASK 10
M3
30 Apr - 25 May
TASK 11
M4
28 May - 20 Jul
TASK 12
M5
23 Jul - 27 Jul

TEST CASES

TEST CASE ID:

TC01

TEST TYPE:

Use-case testing

TEST CASE DESCRIPTION:

An input request for getting personalized suggestions results in a list of items being generated that the user may like. This list will ultimately be returned from the request.

PRECONDITION:

The user must have previous interactions with REFLECT i.e. liked items, past orders, or history.

TEST STEPS:

- Open the application.
- Log in to your account.
- Click on the "Get personalized suggestions" option on the menu.
- Interact with the options available from the system.

TEST DATA:

Products the system assumes the user will like are sent.

EXPECTED RESULT:

The user should like the suggestions sent and have a positive reaction.

ACTUAL RESULT:

- As expected: Success
- Not as expected: Fail

TEST CASE ID:

TC02

TEST TYPE:

Performance testing

TEST CASE DESCRIPTION:

REFLECT will experience an incremental increase in workload, to check how well non-functional requirements like responsiveness, speed, and reliability uphold.

PRECONDITION:

The system has a set workload value that increases steadily until it cannot handle anymore load.

TEST STEPS:

- Identify the testing environment.
- Choose a performance metric.
- Plan and design performance tests.
- Run and configure the tests.
- Analyze and re-test.

TEST DATA:

Sample data which is close to 'real' data for users should be used.

EXPECTED RESULT:

REFLECT should be able to withstand the workload given. And developers should identify performance bottlenecks, if any, in the system.

ACTUAL RESULT:

- As expected: Success
- Not as expected: Fail

CONCLUSION

This concludes our report on the 'REFLECT' software. We have gone through all of the stages of software development ranging from specifying the requirements to modeling the requirements and the interactions that the user will have once they use the software. The risks that might befall REFLECT are also mentioned with ways to minimize and monitor said risks. The Gantt chart for the task breakdown and the test cases for our software are also explained. Lastly, REFLECT is an innovative software that merges the industries of fashion and technology to showcase a unique product, we hope that we have convinced you of the benefits of our software and the importance that it might give if implemented.

REFERENCES

- FOKINA, M. (2023). ONLINE SHOPPING STATISTICS: ECOMMERCE TRENDS FOR 2023. TIDIO. HTTPS://WWW.TIDIO.COM/BLOG/ONLINE-SHOPPING-STATISTICS/
- STATISTA. (2023, MARCH 15). GLOBAL CONSUMERS REASONS FOR RETURNING CLOTHES BOUGHT ONLINE 2021. HTTPS://WWW.STATISTA.COM/STATISTICS/1300981/MAIN-REASONS-RETURN-CLOTHES-BOUGHT-ONLINE/
- FRANKLIN, S. (2022, OCTOBER 7). THE TRUE COST OF ECOMMERCE RETURNS: STATS AND BEST PRACTICES FOR MINIMISING LOSS. LETSBLOOOM. HTTPS://WWW.LETSBLOOM.COM/BLOG/TRUE-COST-OF-ECOMMERCE-RETURNS/#:~:TEXT=RETURNS%20COST%20BUSINESSES%20ABOUT%2066,USUALLY%20SHOULDERS%20THE%20RETURN%20COST).
- MCDOWELL, M. (2021, JULY 26). WHY AR CLOTHING TRY-ON IS NEARLY HERE. VOGUE BUSINESS. HTTPS://WWW.VOGUEBUSINESS.COM/TECHNOLOGY/WHY-AR-CLOTHING-TRY-ON-IS-NEARLY-HERE
- (2021, JULY 13). BUILDING YOUR DREAM AUGMENTED REALITY APP: HARDWARE REQUIREMENTS. WEDESIGNVIRTUAL. HTTP://WEDESIGNVIRTUAL.COM/BUILDING-YOUR-DREAM-AUGMENTED-REALITY-APP- HARDWARE-REQUIREMENTS/