DASH-Fit

Software Requirement Specification (SRS)

GitHub Repository

BY

Team Lead: ***Adeen Atif (22999)***

Member(s):

M Saad Tariq (22947)

Mir Hamza Ali (22767)

Dania Ahmed (22795)

PROJECTS COMMITTEE (PC)

SUPERVISOR: ***Dr Syed Ali Raza***

INDUSTRY SUPERVISOR: ***Arsalan Rashid (Systems Ltd)***

SUBMITTED TO

PROJECTS Manager – FYP

ON

December 4th, 2023

IBA SMCS

School of Mathematics and Computer Science

Table of Contents

1. Problem Statement

1.1 Purpose

This application aims to provide users with a convenient and immersive virtual environment where they can try on clothing items before making a purchase during online shopping. The primary objectives of this project are:

- Offering a realistic virtual try-on experience.
- Bridging the gap between traditional in-store and online shopping.
- Reduced returns due to informed purchases.
- Versatility for retailers in terms of shopping platforms.

1.2 Intended Audience and Reading Suggestions

- End Users: The primary audience for DashFit includes online shoppers who are looking to try on clothing items virtually before making a purchase. This group comprises individuals of various demographics who engage in online shopping and seek a more accurate representation of how clothing items will fit and look on them.

- Retailers and Clothing Outlets: Retailers operating their own online stores or platforms, as well as clothing outlets interested in integrating the DashFit features will benefit from this document. They can use the application to showcase their clothing items more effectively and provide a unique shopping experience to their customers, by getting a sense of what the product offers and its limitations.

1.3 Product Scope

Inclusions*:*

Virtual Try-On Interface: The application will provide a user-friendly and immersive virtual environment where users can select clothing items and visualize how they would look when worn. The interface will utilize augmented reality (AR) technology to superimpose the selected clothing items onto the user's image in real time.

Clothing Item Selection: Users will be able to browse and select clothing items from a catalog stored on the cloud. The application will offer a variety of clothing categories, styles, and sizes to cater to different preferences.

Realistic Visualization: The application will accurately simulate how selected clothing items fit and look on the user's body. Users will be able to rotate and/or adjust the virtual clothing using their camera image on the screen through physical movements.

Exclusions*:*

Physical Measurements: The application will not take physical measurements of users to determine sizing. Users will need to provide their sizing information manually.

Compatibility Constraints: The application will initially only be supporting Android devices. The specific compatibility requirements will be outlined in the technical specifications.

2. System Requirement

2.1 List of Functional Requirements

Key:

H-High, M-Medium, L-Low

Index	Functional Requirement Description	Priority
REQ-A-1	The system shall provide a clothing catalog from which the desired item can be selected.	H
REQ-B-1	The system will allow the user to select an item from the catalog	H
REQ-C-1	The system shall render the clothing model of the chosen body type/size onto the person's body, automatically fitting according to the individual's position on the mobile screen.	H
REQ-D-1	The system shall present options to adjust the cloth's position horizontally or vertically for a better look, if necessary.	M
REQ-D-2	The system shall render multiple selected clothes from each of the two categories-tops and bottoms-at the same time	M
REQ-D-3	The system allows to switch between multiple clothes of the same type in the try room	L
REQ-E-1	The system shall provide options for clothing sizes from which the user can choose based on the size they usually wear	H
REQ-F-1	The system shall allow the user to select multiple items before the try-room button click.	L
REQ-G-1	The system shall enable admin access to add new clothing items to the existing catalog	L
REQ-H-1	The system shall enable admin access to remove existing clothing items from the catalog	L
REQ-I-1	The system shall take a screenshot of the screen upon the user's tap to capture an image of the item tried on.	M

2.2 List of Non-functional Requirements

2.2.1 Performance Requirements

The application's implementation should be sufficiently optimized to ensure smooth, even visuals while viewing the augmented reality clothing. Program should run in a resolution that provides accurate display, enabling all created models and images to be realistic. The program should promptly provide visual response to any input provided by the user.
The performance requirements need to be met in order to accurately render the clothing with as high accuracy as possible.

2.2.2 Safety Requirements

DashFit is software compatible with most low-end mobile phones capable of running a unity mobile application. However, it should be ensured that the software does not consume a lot of memory or compromise other operations of the mobile system. Excessive load on the system can potentially cause overheating of the system, which may result in hardware damage.

2.3 User Interface Requirements

- Easy-to-use, self-explanatory user interface that can be understood by users of all age groups and backgrounds.
- The application must provide a responsive and smooth user experience, with clothing items rendered on the user's image in real-time, ensuring minimal delay and lag.
- The application will work best in good lighting conditions and a mobile camera with decent specifications according to the current standards.
- The software should run adequately on low to mid-end systems but will give the most realistic results on devices with the appropriate resolution requirements.
- The application must adhere to the security requirements outlined earlier in this document to protect user data, privacy, and prevent unauthorized access.

3. Functional Requirement Specification

3.1 Functional Requirements Specification Use Cases

3.1.1 Browse Catalog

User Feature

3.1.1.1 Description and Priority

Browse Catalog is a basic function of the application. It is of High Priority since almost every other feature stems out from this one.

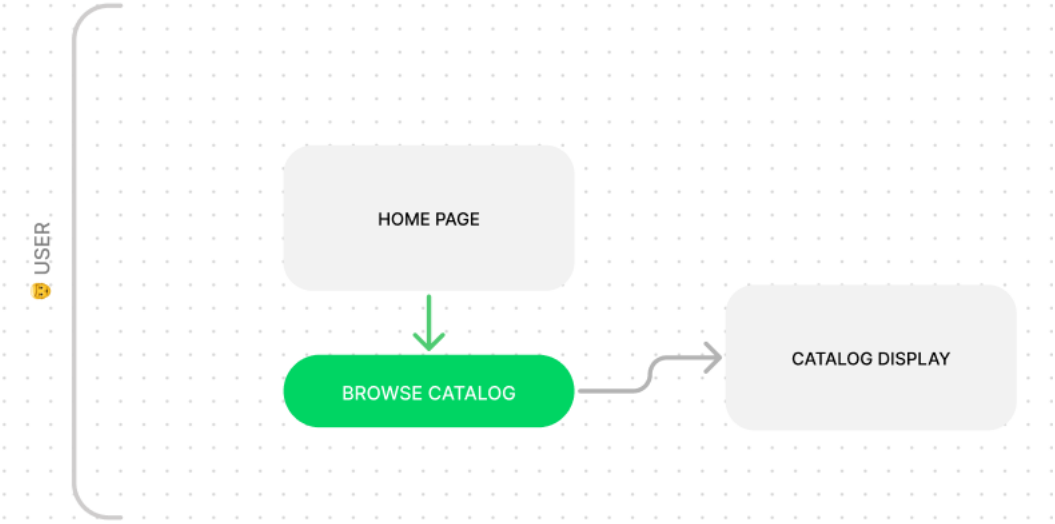
3.1.1.2 Functional Requirements

REQ-A-1

3.1.1.3 Stimulus/Response Sequences

Step 1: User opens the application DashFit

Step 2: Catalog is displayed



2.1.1.4 Dependency/Assumption

The primary condition to use this feature is to have the program running and to select the desired item.

3.1.2 Select Item

User Feature

3.1.2.1 Description and Priority

The DashFit application allows you to select an item from the catalog using an onscreen click. This feature has high priority.

3.1.2.2 Functional Requirements

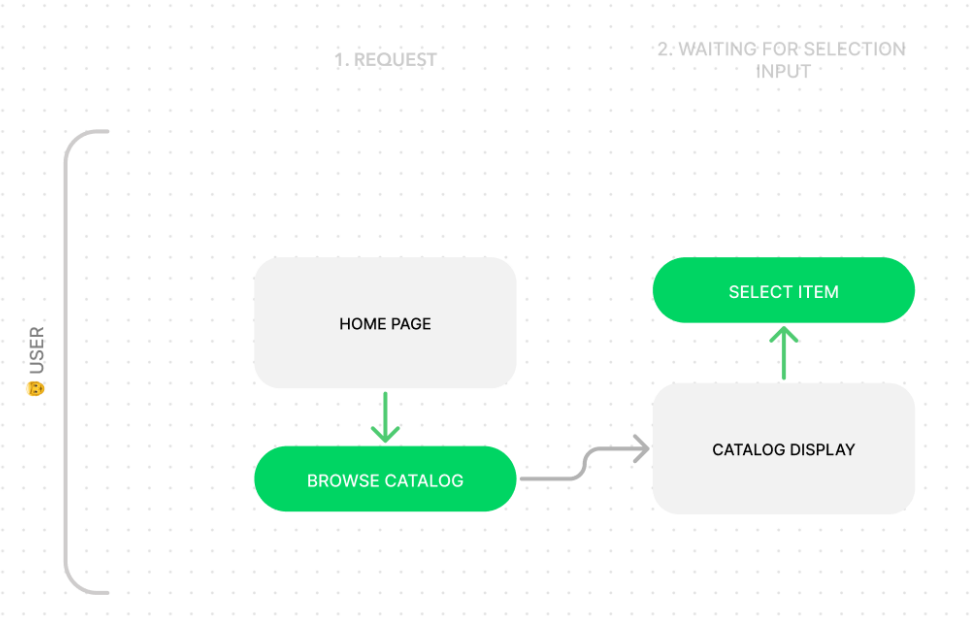
REQ-A-1
REQ-B-1

3.1.2.3 Stimulus and Priority

Step 1: User opens the program DashFit

Step 2: The user browses the catalog using a swiping motion

Step 3: User clicks on a particular item



3.1.2.4 Dependency/Assumption

The primary condition to use this feature is to have the program running and to be present in the browsing catalog screen view. The user then selects the item they want to try on.

3.1.3 View Virtual Clothing

User Feature

3.1.3.1 Description and Priority

When all preliminary conditions are met, the selected clothing of the selected body type will appear on the person and automatically fit according to where the individual is on the screen. You can point the camera towards a person, or you can point towards a mirror. This feature has high priority.

3.1.3.2 Functional Requirements

REQ-B-1
REQ-E-1
REQ-C-1

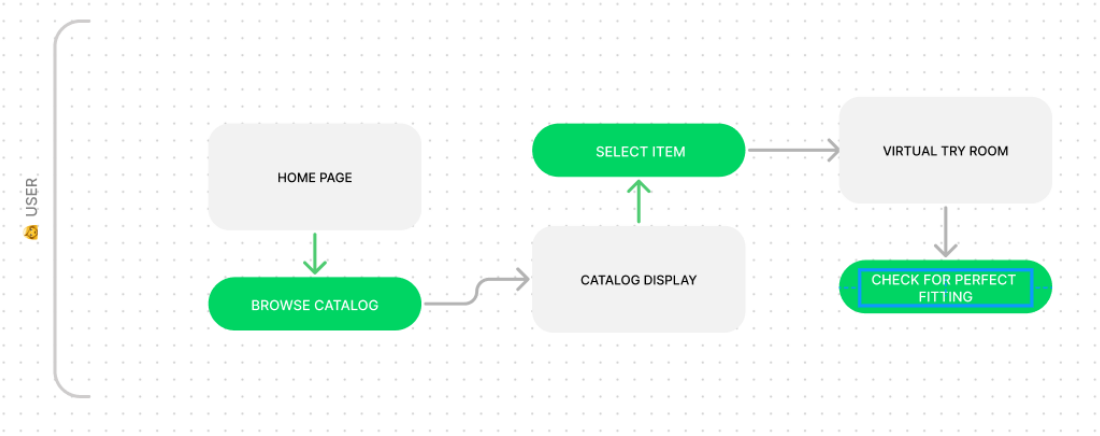
3.1.3.3 Stimulus/Response Sequences

Step 1: User opens the program DashFit

Step 2: Look for clothing item and select size

Step 3: User opens the try room and moves the camera around to calibrate

Step 4: View clothing item in augmented reality.



3.1.3.4 Dependency/Assumption

The primary condition to use this feature is to have the program running and to click on and select all necessary requirements for the clothing display.

3.1.4 Adjust Position

User Feature

3.1.4.1 Description and Priority

When the clothing item appears on the screen, the user has the functionality to adjust the cloth's horizontal and vertical positioning if the need arises for a slightly better accurate look. This feature has medium priority.

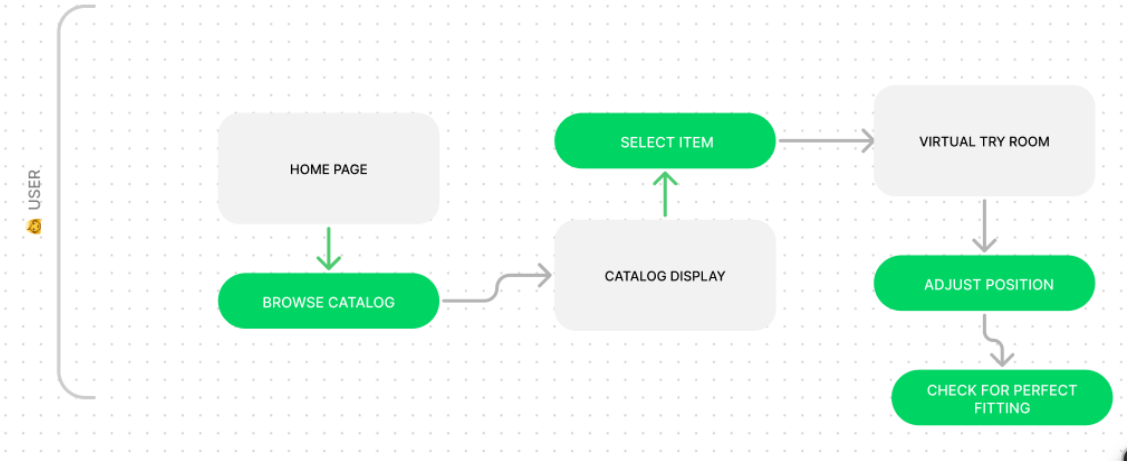
3.1.4.2 Functional Requirements

REQ-B-1
REQ-E-1
REQ-C-1
REQ-D-1

3.1.4.3 Stimulus/Response Sequences

Step 1: User opens the program DashFit

Step 2: Look for clothing item and select



3.1.4.4 Dependency/Assumption

The primary condition to use this feature is to have the program running and to click on and select all necessary requirements for the clothing display and then adjust the clothing item according to the user's needs with the button functionality provided.

3.1.5 Select Size

User Feature

3.1.5.1 Description and Priority

The DashFit application allows you to select a size category for an item from the catalog using an onscreen click. This feature has high priority.

3.1.5.2 Functional Requirements

REQ-B-1
REQ-F-3

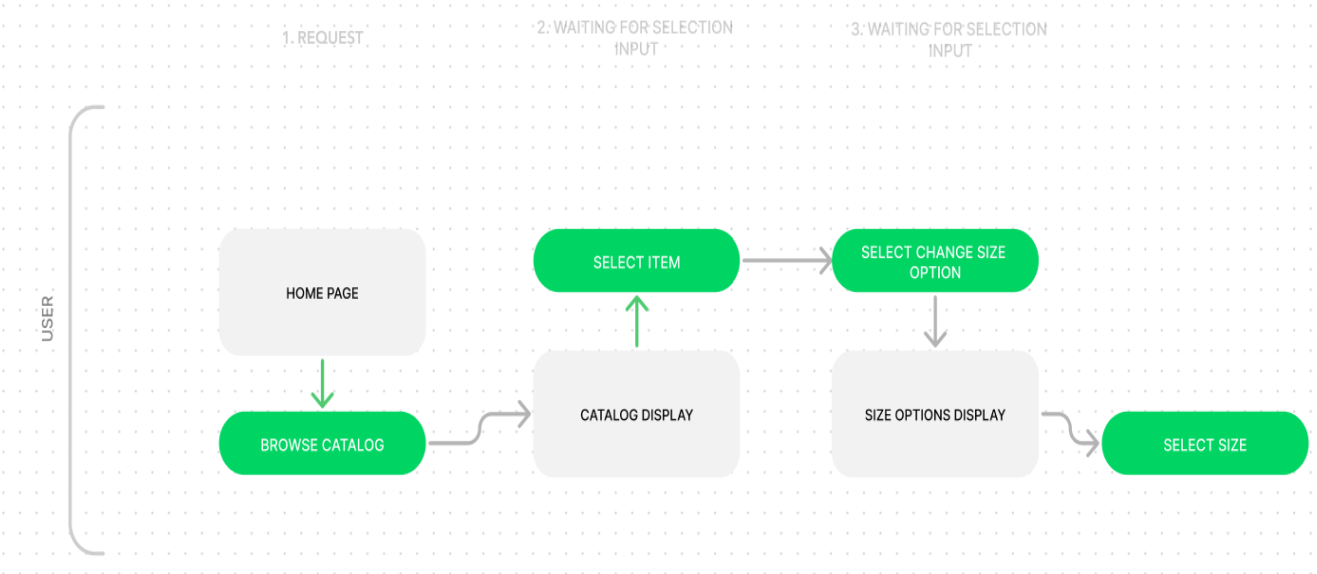
3.1.5.3 Stimulus/Response Sequences

Step 1: User opens the program DashFit

Step 2: Select an item from the catalog

Step 3: Select a body type from the on-screen view

Step 4: Select the size of the item from the on-screen view



3.1.6 Select Multiple Items

User Feature

3.1.6.1 Description and Priority

The DashFit application allows you to select multiple clothes before entering the try room to allow for a better user experience. This feature has low priority.

3.1.6.2 Functional Requirements

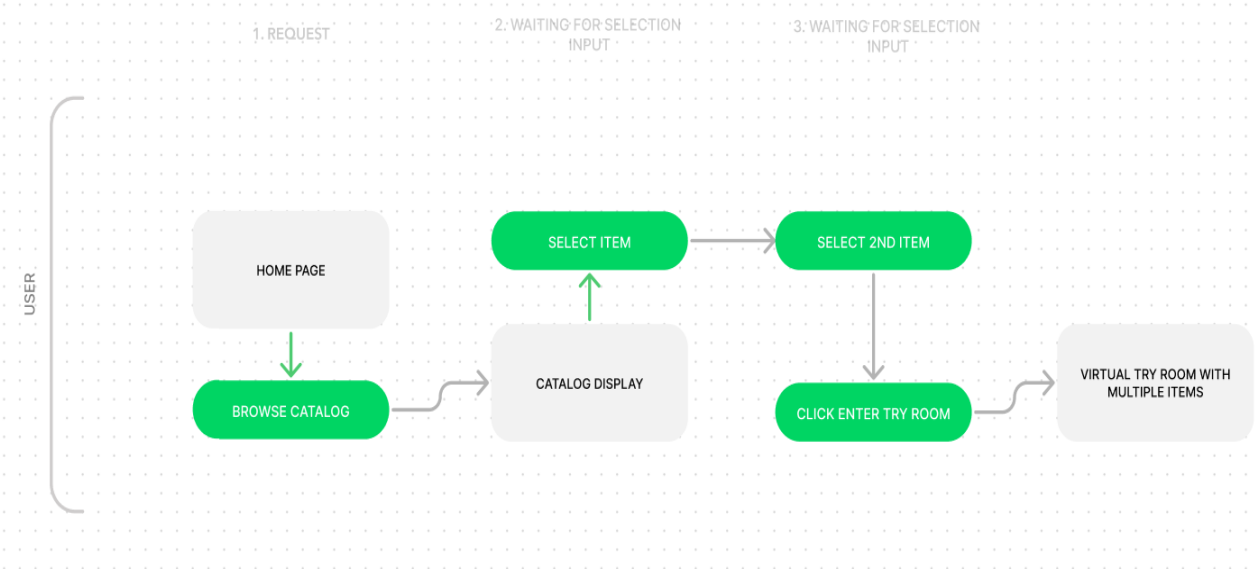
REQ-B-1
REQ-F-1
REQ-D-2

3.1.6.3 Stimulus/Response Sequences

Step 1: User opens the program DashFit

Step 2: Select an item from the catalog

Step 3: Select more items before clicking on try room button



3.1.7 Switch Between Items

User Feature

3.1.7.1 Description and Priority

The DashFit application allows you to switch between the selected clothes in the AR try room to give a more realistic user experience. This feature has low priority.

3.1.7.2 Functional Requirements

REQ-A-1
REQ-B-1
REQ-F-1
REQ-D-3

3.1.7.3 Stimulus/Response Sequences

Step 1: User opens the program DashFit

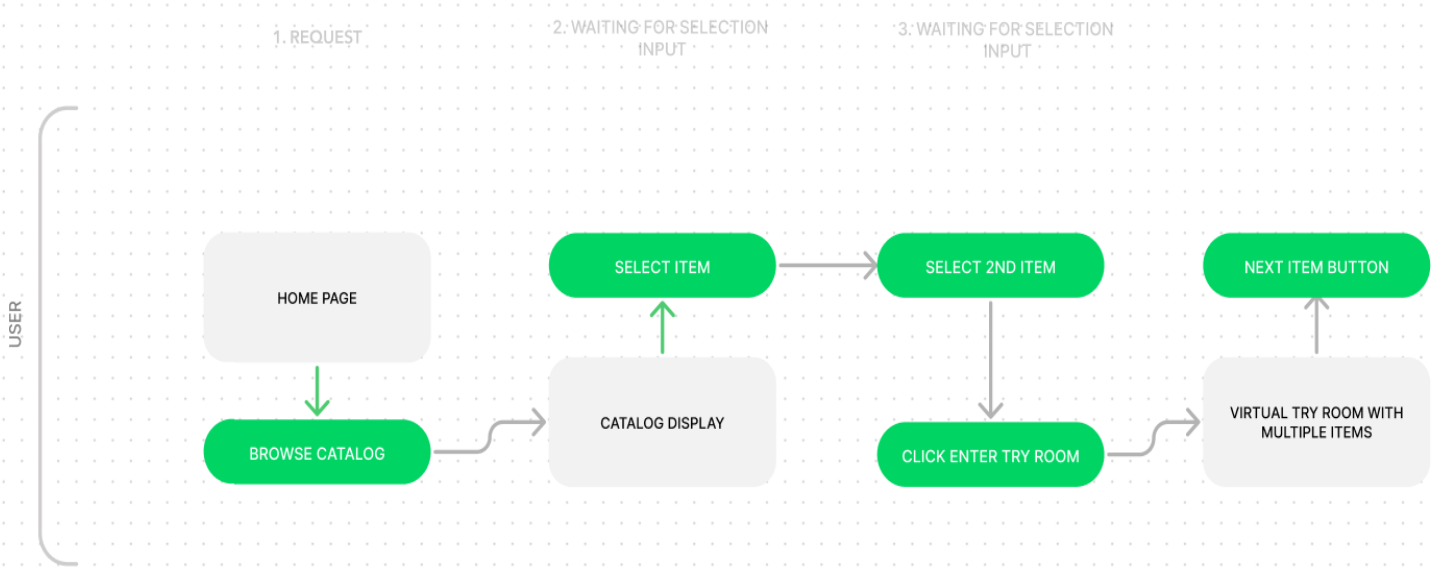
Step 2: Select multiple items from the catalog

Step 3: Select appropriate body type

Step 4: Select appropriate sizes for each clothing

Step 5: Select try room button

Step 6: Click on change item button to iterate between clothes



3.1.8 Screenshot the final look

User Feature

3.1.8.1 Description and Priority

Once the user is satisfied with how the item looks on them, they can take a screenshot to capture their final look. This feature has medium priority.

3.1.8.2 Functional Requirements

REQ-B-1
REQ-E-1
REQ-C-1
REQ-I-1

3.1.8.3 Stimulus/Response Sequences

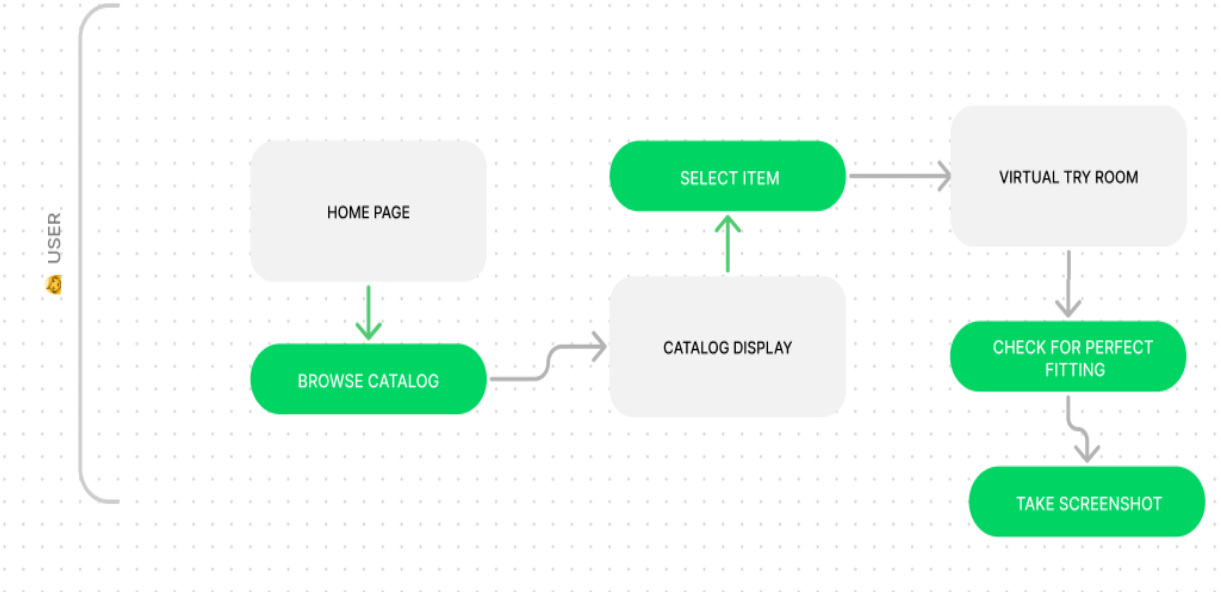
Step 1: User opens the program DashFit

Step 2: Look for clothing item and select size

Step 3: User opens the try room and moves the camera around to calibrate

Step 4: View clothing item in augmented reality.

Step 5: User taps the screenshot button on the screen



3.1.9 Add Item

Admin Feature

3.1.9.1 Description and Priority

The DashFit application allows the admin to add catalog items which will then automatically appear in the defined sizes in the user catalog view. This feature has low priority given that some items are already present in the catalog.

3.1.9.2 Functional Requirements

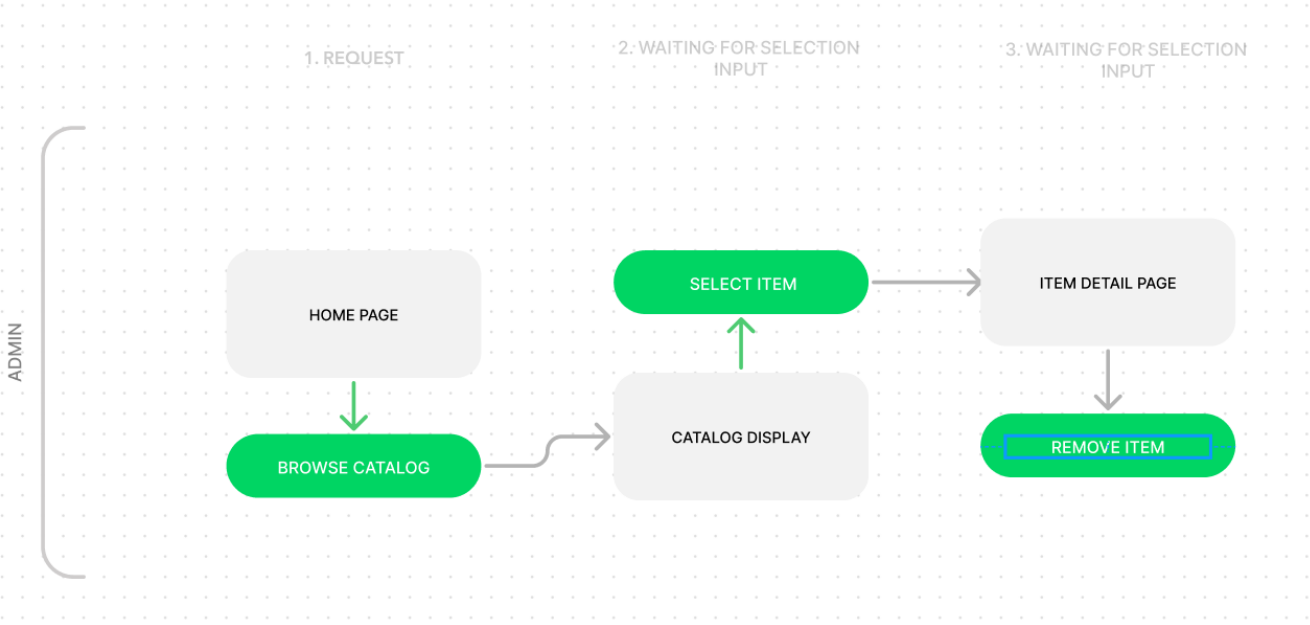
REQ-A-1
REQ-G-1

3.1.9.3 Stimulus and Priority

Step 1: Admin opens the program DashFit

Step 2: Admin goes to the catalog screen

Step 3: Admin selects add item option with the relevant details



3.1.9.4 Dependency/Assumption

The primary condition to use this feature is to have the program running and to be present in the browsing catalog screen view. The admin then selects the add item option and proceeds with filling in the appropriate details for the new item.

3.1.10 Remove Item

Admin Feature

3.1.10.1 Description and Priority

The DashFit application allows the admin to remove catalog items which will then automatically disappear from the user catalog view. This feature has low priority.

3.1.10.2 Functional Requirements

REQ-A-1

REQ-H-1

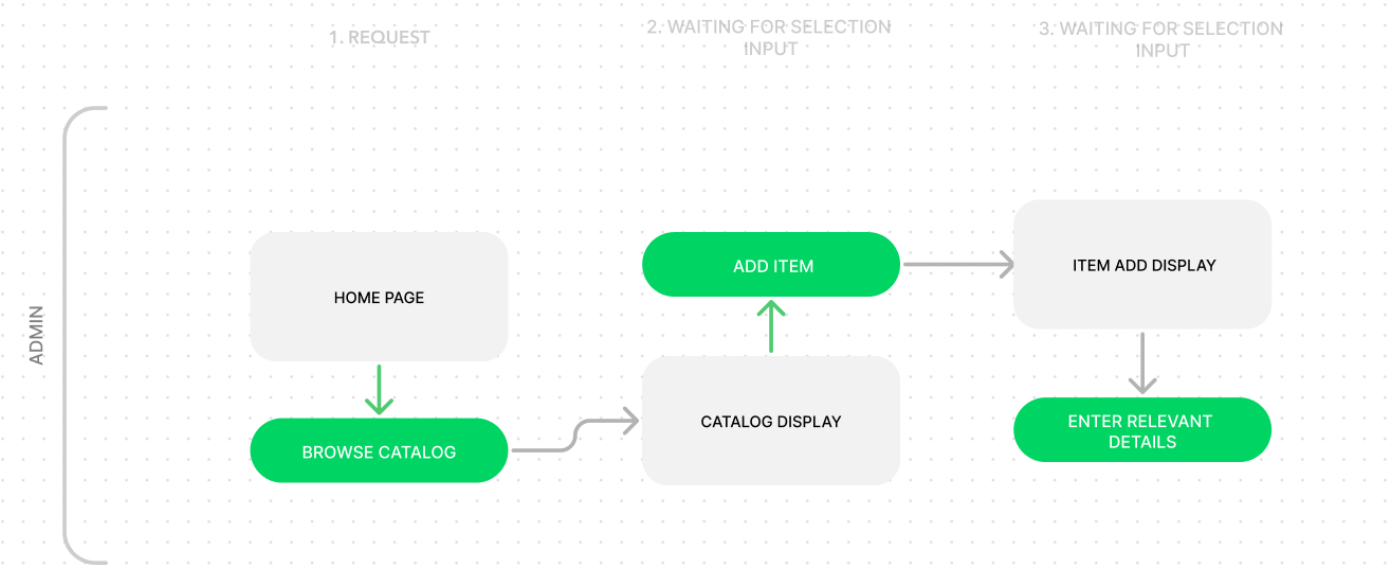
3.1.10.3 Stimulus and Priority

Step 1: Admin opens the program DashFit

Step 2: Admin goes to the catalog screen

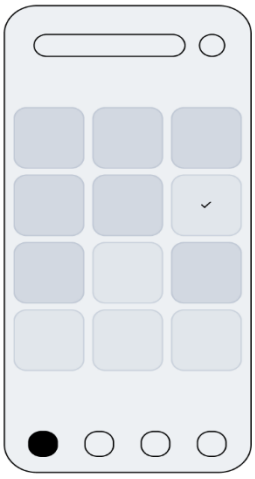
Step 3: Admin selects item

Step 4: Admin selects remove item option

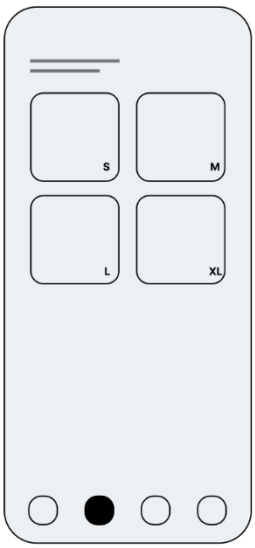


4. User Interface Specification

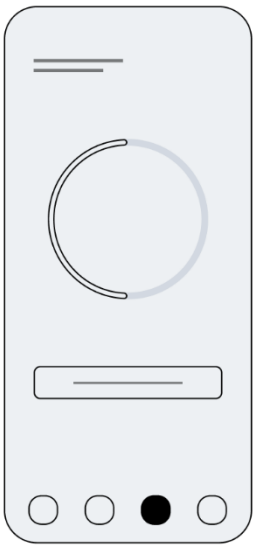
Upon opening the app a landing page appears. Once the user clicks the arrow at the bottom of the page, the home



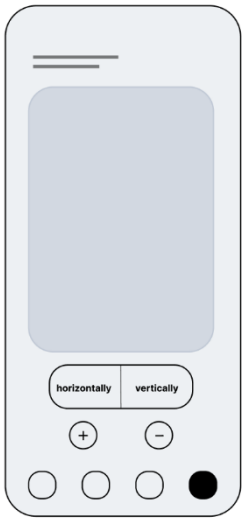
The home page shows a catalogue of all the clothing items available. User can click on items to try them on.



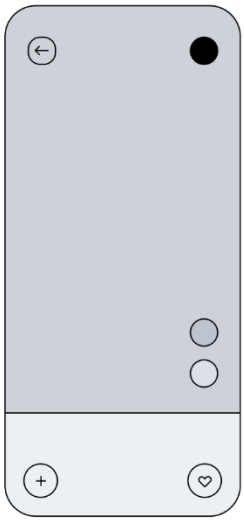
After choosing the item the user is redirected to another screen where they can select the size they want from the given options to try on



Upon choosing next the screen displays the loading page while the system detects the user and overlays the clothing item onto him/her



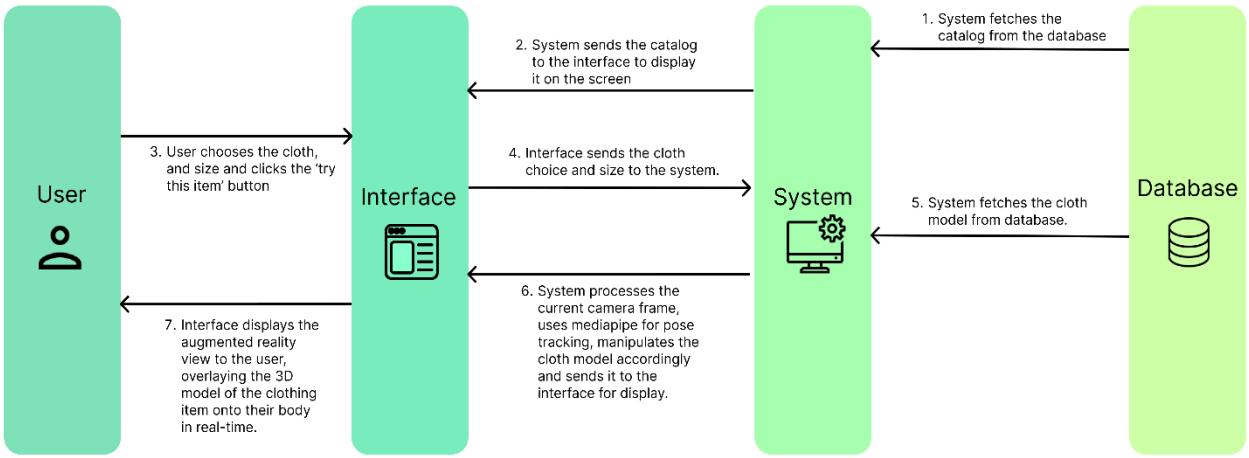
After a waiting period of 5-10 seconds, the camera adjustment screen appears which gives a preview of how the item looks on the user while allowing them to make minor adjustments (move the clothing model up/down, left/ right) to get an accurate fit



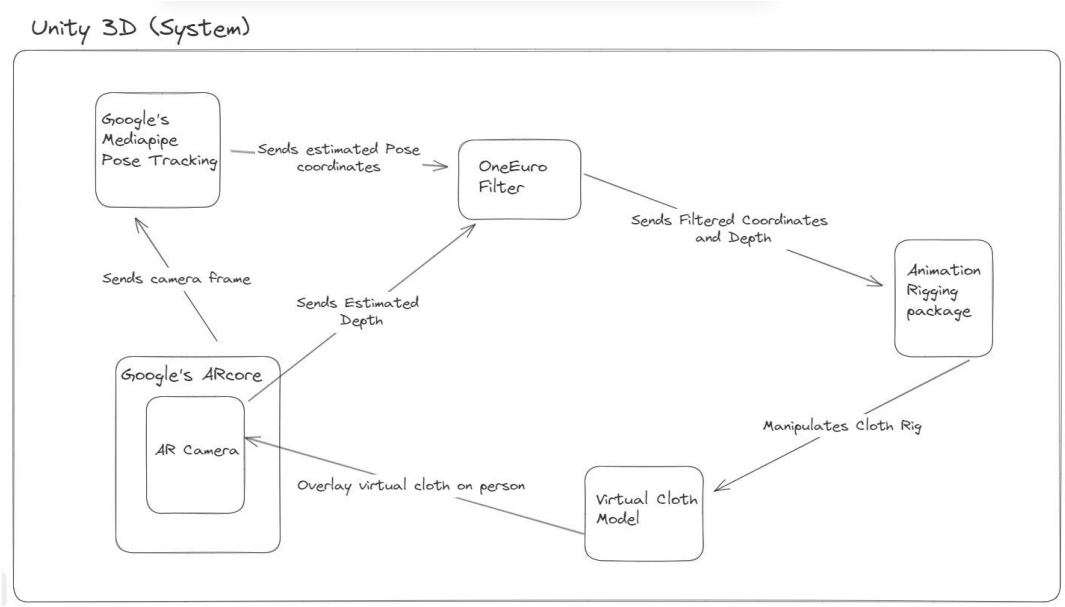
Once the user has finalized the adjustments by clicking on the tick button on the previous screen, they can now see the final fit of the clothing item on themselves

5. Domain Analysis

5.1 Domain Model



5.1.1 System (Internal Working and Flow)



Diagrammatic Representation

5.1.1.1 Google's ARcore

ARcamera

- Shows user the Augmented Reality Experience
- Tracks Orientation and Position
- Estimates Lighting conditions
- Manages Occlusion and Environment Depth calculation using pre-trained models

5.1.1.2 Google's Mediapipe Pose Tracking

Takes Camera Frame and uses Pose Tracking Graph/Model to estimate human pose

5.1.1.3 OneEuroFilter

Filters the Pose coordinates and estimated Depth to minimize Jitter and Lag

Algorithm 1: 1€ filter

EXT: First time flag: *firstTime* set to *true*
Data update rate: *rate*
Minimum cutoff frequency: *mincutoff*
Cutoff slope: *beta*
Low-pass filter: *xfilt*
Cutoff frequency for derivate: *dcutoff*
Low-pass filter for derivate: *dxfilt*
IN : Noisy sample value: *x*
OUT: Filtered sample value

```
if firstTime then
    firstTime ← false
    dx ← 0
else
    dx ← (x - xfilt.hatxprev()) * rate
end
edx ← dxfilt.filter(dx, alpha(rate, dcutoff))
cutoff ← mincutoff + beta * |edx|
return xfilt.filter(x, alpha(rate, cutoff))
```

Algorithm 2: *Filter* method of Low-pass filter

EXT: First time flag: *firstTime* set to *true*
IN : Noisy sample value : *x*
Alpha value : *alpha*
OUT: Filtered value

```
if firstTime then
    firstTime ← false
    hatxprev ← x
end
hatx ← alpha * x + (1 - alpha) * hatxprev
hatxprev ← hatx
return hatx
```

Algorithm 3: Alpha computation

IN : Data update rate in Hz: *rate*
Cutoff frequency in Hz: *cutoff*
OUT: Alpha value for low-pass filter

```
tau ← 1.0 / (2*π*cutoff)
te ← 1.0 / rate
return 1.0 / (1.0 + tau/te)
```

OneEuro Filter Algorithm

5.1.1.4 Unity's Animation Rigging Package

- Used for Rigging the Virtual Cloth Model
- Manipulates the Virtual Cloth Rig at runtime according to the coordinates received using inverse kinematics and bone manipulation

5.1.1.5 Virtual Cloth Model

3D model(rigged) of a cloth consisting of base model + texture, of a specific size (small, medium, or large)

6. References

Géry Casiez, Nicolas Roussel, Daniel Vogel. 1€ Filter: A Simple Speed-based Low-pass Filter for Noisy Input in Interactive Systems. CHI'12, the 30th Conference on Human Factors in Computing Systems, May 2012, Austin, United States. pp.2527-2530, ff10.1145/2207676.2208639ff. ffhal-00670496f