

# Functional Requirements Draft

## Weather to Wear

### F-1 Clothing-Catalog Management

- **F-1.1** The system shall allow a user to add a wardrobe item by taking a photo and completing a short form (name, category, material, weather-appropriateness, tags).
- **F-1.2** The system shall let the user edit or delete any wardrobe item they own.
- **F-1.3** The system shall provide search, filter and sort capabilities over the wardrobe (e.g., by category, colour, material).
- **F-1.4** Wardrobe data and images shall be persisted so that a user's catalog is available across devices.
- **F-1.5** Each wardrobe item shall support tagging for weather conditions (e.g., "hot", "rainy") and occasions (e.g., "work", "casual").
- **F-1.6** The system shall enforce a maximum file size and accepted image types (JPEG, PNG) during wardrobe uploads.

### F-2 Weather-Data Integration

- **F-2.1** The system shall retrieve a location-specific weather forecast (min/max temperature, precipitation, wind, humidity, condition) from an external API.
- **F-2.2** If the primary provider fails, the system shall automatically fallback to a secondary API without user intervention.
- **F-2.3** Weather data shall be refreshed at a configurable interval (default: every 6 hours) so that recommendations stay current.
- **F-2.4** The system shall attempt to automatically detect the user's location using IP-based geolocation.
- **F-2.5** If automatic detection fails, the system shall allow manual location input from the user.
- **F-2.6** The weather module shall log each API call's success/failure and selected provider for debugging and analytics.

### F-3 Wardrobe-Recommendation Engine

- **F-3.1** Given a forecast and the user's wardrobe, the system shall produce at least one ranked outfit recommendation for a selected day.
  - **F-3.1.1** The ranking shall consider temperature range, weather condition, and user preferences.
- **F-3.2** The engine shall learn from user feedback captured through a 5 star rating system.
- **F-3.3** Recommendations shall include layering advice when forecasts indicate significant temperature variation.
- **F-3.4** The engine shall exclude items marked unsuitable for the predicted conditions.
- **F-3.5** The system shall avoid recommending identical outfits across consecutive days, where alternatives exist.

- **F-3.6** The engine shall take time-of-day into account (e.g., colder mornings vs. warm afternoons) if available in the forecast.

## F-4 Social-Sharing Platform

- **F-4.1** Users shall be able to publish an outfit post (image + caption + list of items) to a social feed.
- **F-4.2** Followers shall be able to like and comment on a post in real time.
  - **F-4.2.1** Likes and comments shall update asynchronously without page reload.
- **F-4.3** The system shall filter or block offensive text or images before a post becomes visible.
- **F-4.4** Privacy controls shall let the author restrict visibility to:
  - **F-4.4.1** Public
  - **F-4.4.2** Followers only
  - **F-4.4.3** Private (visible only to user)
- **F-4.5** The system shall allow users to follow and unfollow others.
  - **F-4.5.1** A users feed shall show posts of those who they follow

## F-5 User-Feedback & Learning

- **F-5.1** The system shall record each outfit rating action against the corresponding outfit version.
- **F-5.2** The user's preferences shall be updated after each outfit rating action.
  - **F-5.2.1** The recommendation engine shall incorporate user's preference and weather conditions into future ranking within 24 hours or upon user refresh.
- **F-5.3** The system shall allow users to view and manage their past feedback history.
- **F-5.4** The engine shall increase outfit diversity if consistent negative feedback is received for similar combinations.

## F-6 [Optional] Virtual-Closet Analytics

- **F-6.1** The system shall display usage statistics (e.g., most-worn items, cost-per-wear).
- **F-6.2** Users shall be able to tag clothing by occasion and filter their closet on those tags.
- **F-6.3** The system shall visualize wardrobe insights using graphs (e.g., pie chart for usage distribution).

## F-7 [Optional] Emergency-Outfit Response

- **F-7.1** When the forecast changes beyond a configurable threshold (e.g.,  $\geq 5$  °C drop or rain chance > 60%), the system shall:
  - **F-7.1.1** Notify the user
  - **F-7.1.2** Suggest a new outfit suitable for the new forecast
- **F-7.2** Notifications shall be sent via in-app alert or push notification if enabled.

## F-8 [Optional] Fashion Time-Machine Planner

- **F-8.1** The system shall allow users to attach events or trips to future calendar dates.
- **F-8.2** For each future date with an event, the system shall:
  - **F-8.2.1** Pre-compute an outfit based on long-range forecast
  - **F-8.2.2** Update the outfit automatically if the forecast changes

## F-9 [Optional] Virtual Try-On

- **F-9.1** The app shall provide an AR preview that overlays the recommended outfit on:
  - **F-9.1.1** A live camera view
  - **F-9.1.2** A customizable 3D avatar
- **F-9.2** The try-on experience shall function in modern browsers that support camera access.

## F-10 Cross-Cutting Functionalities

- **F-10.1** Authentication & Authorisation
  - **F-10.1.1** All user-modifiable endpoints shall require authentication.
  - **F-10.1.2** Each user shall only be able to access and modify their own wardrobe, preferences, and posts.
- **F-10.2** Offline Mode
  - **F-10.2.1** The PWA shall cache the latest wardrobe and recommendations.
  - **F-10.2.2** The app shall queue user actions (e.g., ratings) and sync them once back online.
  - **F-10.2.3** Weather forecasts shall be cached to allow the recommendation engine to generate outfits while offline
- **F-10.3** Multiplatform Delivery
  - **F-10.3.1** The app shall work in modern browsers (Chrome, Safari, Firefox).
  - **F-10.3.2** The app shall support installation as a home screen shortcut on desktop.
- **F-10.4** Logging & Monitoring
  - **F-10.4.1** The backend shall log API failures, user actions, and system warnings.
  - **F-10.4.2** Admins shall be able to review logs via the dashboard.
- **F-10.5** Testability
  - **F-10.5.1** Each module (auth, weather, recommendation) shall include unit and integration tests.
  - **F-10.5.2** The system shall be CI/CD integrated, running tests on each publish.