Mental Load and AI:
The Foundations for a Fair Delegation System

Alessandra Zamora

February 1, 2026

Contents

1 Introduction

This project is conducted under **CM3020 Project Template #4: Orchestrating AI Models to Achieve a Goal**. The aim of this template is to design a system that combines multiple pre-trained artificial intelligence models into a coordinated workflow that can address a real world problem. In this project, the domain is **domestic labor and financial management**, and the goal is to reduce cognitive burden and imbalanced workload distribution within households.

Domestic responsibilities and financial management are fundamental aspects of daily life, yet the decision making required to coordinate, monitor and maintain them can create significant cognitive exertion. This problem, commonly known as *mental load*, is described by Daminger (2019) as the invisible labor of managing domestic work, including anticipating other people's needs, tracking deadlines, remembering exceptions, and ensuring that household standards are met, regardless of who performs the physical tasks.

Numerous studies show that this burden is rarely distributed equally within households with women disproportionately assuming the role of the "default organizer", carrying not only visible labor but also the hidden cognitive and emotional effort required to sustain it (E., 2025; Haupt, 2024). Importantly, this imbalance persists even in modern dual income relationships and in couples who self identify as egalitarian: responsibility may be shared in principle, but the cognitive overhead of planning, monitoring, and emotional regulation remains uneven. Over time, this has measurable consequences, including **decision fatigue, psychological distress, reduced leisure and recovery time, spillover into paid work performance, and lower relationship satisfaction** (Craig & Brown, 2017; E., 2025; Haupt, 2024; Yavorsky et al., 2015).

The motivation behind this project is therefore not simply to build a productivity tool, but to investigate whether artificial intelligence can reduce the *thinking behind the tasks*. Existing household management and budgeting tools largely function as passive repositories: they store expenses, list chores, or trigger reminders. While useful, they do not have decision making capacity. As a result, typically the same default organizer must still decide what matters, when it matters, and who should act. The central hypothesis of this project is that a coordinated AI system could do more: it could assume parts of the planning, monitoring, and delegation process itself, thereby reducing cognitive and emotional load.

To test this hypothesis, this project implements a smart household assistant named *Otto*, designed as a pipeline of interacting models rather than a single monolithic system. In line with template' requirements, Otto orchestrates several pre-trained and fine-tuned models that operate on different data types and feed into a unified decision process. The system consists of three main subsystems:

1. **Receipt Understanding**: An *image-to-text* and language processing pipeline that extracts structure expense information from photographed receipts.
2. **Spending Forecasting**: A time-series model that predicts next month spending patterns and identifies potential budget risks.
3. **Delegation and Fairness Modeling**: A deep neural network trained to recommend chore assignments and detect imbalance in household effort using imitation learning from historical data.

Each subsystem solves a distinct subproblem, but they are not independent. The receipt understanding module feeds structured financial data into the forecasting model; both financial and chore logs feed into the delegation model; and the outputs of all models are combined into a real time dashboard that presents fairness metrics, predictions, and recommended actions.

In this way, Otto functions as a closed loop AI system that both interprets states and produces actionable guidance. Rather than focusing on isolated model performance, the emphasis is on system-level behavior: how perception models, prediction models, and decision models interact to create a practical assistant capable of supporting human life at home.

2 Literature Review

To understand why everyday responsibilities become cognitively demanding, it is first necessary to clarify the different forms of labor that shape domestic life. The literature identifies several forms of labor that operate simultaneously in the home: **mental load**, **cognitive labor**, **emotional labor** and **domestic labor**. Together, these concepts help explain why household coordination is difficult, and why inequalities persist even when responsibilities appear evenly split.

2.1 Conceptual Definitions

Mental load refers to the continuous and largely invisible effort involved in keeping track of tasks, anticipating needs, and ensuring that responsibilities are completed. In research, this is described as the mental cost imposed by task management and the limits it places on an individual's capacity to meet daily demands (Gopher, 1994). In the domestic context, Daminger (2019) defines mental load as responsibility for outcomes: maintaining standards, coordinating interdependent activities, and carrying long-term accountability regardless of who performs the physical task. As Walzer (1998) notes, mental load is rarely recognized as labor because it happens internally, is unpaid, and is often framed as care or commitment rather than work.

Cognitive labor captures the mental processes that translate household needs into organized action. Morini (2007) describes cognitive labor as the use of mental abilities for coordination and relational maintenance. Within households, this includes planning, prioritizing, scheduling, and monitoring tasks. Daminger (2019) identifies four core components: anticipation, identification, decision-making, and monitoring. These activities are unevenly visible, which helps explain why a household can appear organized while the cognitive burden remains concentrated on one person.

Emotional labor extends domestic responsibility beyond planning into managing emotions and relationships. Originally theorized in professional settings (Hochschild, 1983, 2003), emotional labor includes regulating one's own emotions and responding to the emotional needs of others. In domestic life, this often involves maintaining harmony, preventing conflict, and absorbing stress so daily routines can continue. Brotheridge and Lee (2003) show that emotional labor requires regulating the intensity, frequency, and duration of emotional expression, and that these demands are closely intertwined with mental load.

Domestic labor refers to the unpaid work required for household functioning, including core chores, episodic tasks, and childcare (Lee & Waite, 2005). Much of this work, especially its organizational and administrative aspects, has been described as “invisible work,” shaped by gender norms and expectations of care (Daniels, 1987; DeVault, 1991). These norms influence whose work is recognized and whose organizational labor remains unseen.

2.2 Mental Load, Cognitive Labor, and Gendered Inequality

Research consistently shows that domestic labor extends far beyond physical chores to include continuous management, anticipation, and emotional regulation (Daminger, 2019; DeVault, 1991; Hochschild, 1983; Walzer, 1998). Mental load and cognitive labor help explain why divisions of labor that appear equal can still feel deeply unequal. Even when chores are split, one partner may remain responsible for noticing what needs to be done, remembering timelines, monitoring progress, and handling breakdowns. This organizational responsibility consumes attention and creates persistent background pressure (Offer, 2014).

Recent studies show that these patterns persist in contemporary dual income households. Women continue to carry a disproportionate share of cognitive household labor, even when both partners are employed full time and describe their relationships as egalitarian (E., 2025; Haupt, 2024). This imbalance is associated with higher work–family conflict, reduced wellbeing, and chronic stress. These findings suggest that the problem is not primarily a lack of reminders or task lists, but an unequal distribution of responsibility for planning and coordination. From a technical perspective, this reframes the challenge: **the goal is not simply to track tasks, but to redistribute decision making itself.**

Beyond subjective experience, the literature links mental load to concrete consequences. Unequal cognitive responsibility is associated with **decision fatigue and reduced leisure time** (Craig & Brown, 2017), **spillover into professional work and reduced productivity** (Yavorsky et al., 2015), and **lower relationship satisfaction** driven by perceived unfairness and unacknowledged effort (E., 2025; Haupt, 2024). These findings call for interventions that go beyond simply tracking chores and expenses, instead directly easing the planning and oversight load that leads to these outcomes.

2.3 Existing Digital Tools and Their Limitations

A wide range of consumer tools exist for managing chores and household finances. However, most focus on documentation, reminders, or visualization rather than decision making. While these tools can improve transparency, they rarely reduce the cognitive burden of planning and coordination, and may even reinforce mental load by requiring users to structure information before the tool becomes useful.

Table 1: Comparison of existing household management tools

Tool	**Primary Function**	**Strengths**	**Limitations for Mental Load**
Goodbudget, YNAB	Budget tracking	Clear expense visualization; shared access	Requires manual categorization and planning; no forecasting or decision support
Splitwise	Expense splitting	Transparency in shared costs	Does not model fairness over time or non-financial effort
Sweepy, OurHome	Chore scheduling	Routine building; reminders	Users still decide assignments; planning burden remains
Cozi	Family organization	Centralized calendar and lists	Focuses on coordination, not responsibility redistribution
Alexa, Google Home	Voice reminders	Low-friction reminders	Reactive only; no context, fairness, or learning

Across these tools, a common limitation emerges: responsibility for planning remains with the user, particularly the ”default organizer”. Tasks must be defined, assigned, and negotiated before the system can assist. As a result, the cognitive labor of anticipating needs and deciding who should act is preserved rather than reduced. This aligns with findings in the sociological literature, which suggest that tools focused solely on execution can unintentionally reinforce mental load instead of alleviating it (E., 2025; Haupt, 2024).

This gap motivates an alternative approach that treats the household as a decision environment rather than a static checklist. Such a system requires perception (capturing expenses), forecasting (anticipating budget risk), and decision making (delegating tasks under fairness constraints). The remainder of this review examines technical approaches for each of these layers and situates the models implemented in this project within that landscape.

3 AI Techniques and Design Rationale

Before defining the system design, it is useful to examine which AI techniques are commonly applied to each subproblem in similar real world technologies, and what tradeoffs they introduce under limited data and MVP constraints. Many applied systems combine pretrained components with *task-specific* learning and explicit domain constraints, rather than relying on a single *end-to-end* model. This section synthesizes candidate approaches for the three functional layers of the proposed system: **perception, forecasting, and decision making**.

3.1 Perception: Receipt Understanding

The perception task consists of converting a receipt image into a structured transaction record containing a total amount and an expense category and subcategory. In the literature and in industry systems, three technical approaches are commonly used.

The first is a **classic OCR-based pipeline with post-processing rules**. In this setup, a generic OCR engine such as Tesseract extracts raw text, which is then parsed using regular expressions and heuristics to identify fields such as totals and dates (Smith, 2007). The strength of this approach is its simplicity: it is lightweight, transparent, and easy to debug. Its main weakness is sensitivity to layout variation and OCR noise, which limits generalization across receipt formats.

A second option is the use of **cloud-based receipt parsing APIs**, such as Google Document AI or Amazon Textract. These systems can achieve strong accuracy, but they are costly and provide limited control over model behavior. For an academic project focused on understanding model behavior and evaluation rather than API usage, these tradeoffs are significant.

A third approach involves **pretrained document understanding transformers**, such as LayoutLMv3 and Donut, which jointly model text content and visual layout (Huang et al., 2022; Kim et al., 2022). These models can generalize well across formats and can outperform *OCR-plus-rules* systems when sufficient labeled data and compute are available. However, they require GPU resources, careful fine-tuning, and non-trivial post-processing to adapt outputs to a custom schema.

3.2 Forecasting: Spending Prediction and Budget Risk

The forecasting component aims to estimate future household spending and flag whether a user is likely to exceed a planned budget. Technically, this can be framed either as a time-series forecasting problem or as a supervised learning problem over engineered temporal features.

Traditional statistical models such as ARIMA and exponential smoothing are well established in financial forecasting and perform well under stable seasonal patterns (Box et al., 2015). However, household spending is often irregular and influenced by behavioral factors such as salary timing, one-off purchases, and category-specific volatility, which limits the usefulness of purely statistical approaches.

Neural time-series models, including LSTMs and transformer-based architectures, can capture long-range dependencies and complex interactions (Lim et al., 2021). While powerful, these models typically require longer histories and careful tuning to avoid overfitting.

An alternative approach is feature-based forecasting, where transaction histories are converted into structured attributes (lags, rolling averages, volatility measures, and seasonal encodings) and learned using a non-linear model such as gradient boosting (Chen & Guestrin, 2016). This approach is well suited to small datasets, remains interpretable, and allows domain-specific signals such as budget thresholds to be incorporated directly.

3.3 Decision-Making: Task Delegation and Fairness Metrics

The delegation component assigns household tasks to partners while balancing effort over time under availability and feasibility constraints. This problem is closer to constrained allocation than to standard prediction.

Rule-based scheduling and optimization methods, such as integer linear programming, can encode fairness and feasibility constraints explicitly and are widely used in workforce scheduling (Pinedo, 2016). However, these methods can become brittle if preferences and capacities evolve frequently, and they typically require an explicit objective that may not capture household-specific tradeoffs.

Reinforcement learning offers a framework for long-horizon decision-making with delayed rewards (Sutton & Barto, 2018), but it requires extensive interaction data or simulation and is difficult to evaluate reliably at an early stage. In practical deployments, reinforcement learning in human environments also raises additional concerns around safety, exploration, and user trust.

Imitation learning provides a practical compromise. A teacher policy encodes constraint handling and fairness logic, and a supervised model learns to approximate these decisions across many scenarios (Osa et al., 2018). This approach allows the system to behave consistently from the outset while remaining adaptable as real household data becomes available. Imitation-based systems are also compatible with staged deployment: initial behavior can be generated by a transparent teacher policy, and the learned model can later be refined as real usage data accumulates.

3.4 Summary and Design Implications

The reviewed literature suggests that supporting domestic life requires more than simple tracking or reminders. Effective systems must combine perception, prediction, and decision-making in order to reduce the cognitive burden of planning, monitoring, and coordination.

Together, studies of mental load, domestic labor, and existing digital tools motivate a layered AI architecture in which distinct models operate over different types of data and time horizons. Receipt understanding enables perception of financial activity, spending forecasting supports anticipation of future risk, and task delegation models provide decision support under fairness and feasibility constraints. Structuring the system as a modular, multi-stage pipeline allows each component to be evaluated, improved, and replaced independently, while still supporting a coherent end-to-end workflow.

This design directly supports the project goal of redistributing cognitive responsibility from the user to the system itself: instead of merely storing information, the assistant can reason over household data and provide actionable recommendations that reduce the need for continuous human oversight.

4 Design

4.1 Project Overview and System Goal

This project is an intelligent system whose main objective is to minimize the cognitive load associated with managing domestic life. The system is framed as an assistant that can take on parts of the organizer role: capturing financial information from receipts, predicting budget risk before it occurs, and recommending household task assignments that remain fair over time.

The assistant is implemented as three core subsystems:

1. **Receipt Understanding:** transforms receipt images into structured transactions using OCR, parsing, and categorization.
2. **Spending Forecasting:** predicts next-month spending and flags whether the household is likely to exceed its budget.
3. **Task Delegation:** recommends which partner should perform tasks while balancing effort and constraints over a monthly horizon.

4.2 Target Users

The primary users of this system are cohabiting couples and shared households who struggle with coordinating domestic responsibilities and shared finances. This includes couples in which one partner experiences a disproportionate share of planning, monitoring, and follow-up, as well as partners who are newly living together and wish to establish balanced routines from the outset rather than resolving conflicts after they emerge.

The system is designed to support a wide range of household structures, including modern dual-income couples, roommates, and other co-living arrangements. Mental load is not limited to traditional family models or single-earner households; it persists wherever multiple people must coordinate shared resources, tasks, and expectations. By treating the household as a collaborative decision environment, the system aims to support fairness and sustainability across diverse living situations.

4.2.1 Additional Users

The system can also be useful for individuals who struggle with home administration alone, including those with execute dysfunction (e.g. ADHD) who need more than reminders. The emphasis is on reducing the burden of planning and adapting rather than adding another checklist. Secondary users include researchers and practitioners interested in modeling domestic fairness, cognitive labor, and AI assisted home management.

4.3 User Requirements

4.3.1 Functional Requirements

Receipt Understanding

- Allow a user to capture or upload a receipt image and store it as input artifact.
- Convert the receipt image into a machine readable text using OCR, with preprocessing to reduce noise.
- Extract key fields (at minimum: total amount; optionally: merchant and date) and output a structured transaction record.
- Predict a spending category from the extracted text using a supervised model and output a confidence score.
- Produce a structured JSON transaction object that can be logged and later aggregated for forecasting.

Spending Forecasting

- Aggregate transactions into monthly totals and category-level summaries.
- Forecast next-month household spend as a numeric estimate.
- Compute and overspend risk flag for the *current budgeting period*.
- Provide interpretable signals that justify warnings.

Task Delegation and Fairness

- Represent partners with capacity constraints and track cumulative effort.
- Represent tasks with duration, frequency, priority, and feasibility.
- Recommend an assignee for each task occurrence while balancing effort over time.
- Maintain an interpretable fairness target.
- Output both recommendations and balance indicators.

4.3.2 Non-Functional Requirements

- **Interpretability**
- **Modularity**
- **Feasibility**

4.4 System Architecture

The system is designed as a layered data flow, where information progresses from raw input to decision outputs through preprocessing, feature extraction, and modeling stages. (Architecture Diagram here)

4.5 Model Orchestration and Data Flow

The three AI subsystems form a decision pipeline rather than operating independently. Receipt understanding generates structured financial records, which are aggregated and passed into the forecasting system. Forecast outputs modify the fairness targets used by the task delegation system, creating a closed feedback loop between perception, anticipation, and action.

Receipt understanding produces:

$$(\text{amount}, \text{currency}, \text{category}, \text{timestamp})$$

Forecasting produces:

$$\hat{S}_{t+1}, \quad P(\text{overspend})$$

Delegation uses these to update fairness targets and assign tasks.

Each subsystem operates on a different time horizon (event, monthly, weekly), allowing fast perception and slower planning to coexist.

4.6 Evaluation Criteria

Because the system consists of three technically distinct subsystems, evaluation is defined separately for each component using metrics that are appropriate to its task and data constraints. The goal is not to optimize a single number, but to verify that each model behaves reliably in the role it plays within the overall pipeline.

All evaluations are performed offline using held-out or synthetic data, reflecting the current MVP scope.

4.6.1 Receipt Understanding

The receipt understanding model is evaluated as a text classification problem, complemented by qualitative *end-to-end* checks of the full *OCR-to-output* pipeline.

Metrics

- Overall accuracy on a held-out test split.
- Per-class precision, recall, and F1-score.

- Macro-averaged F1-score to account for class imbalance.
- Qualitative inspection of the full pipeline (OCR → amount extraction → category prediction).

Rationale Receipt categories are heavily imbalanced, with common expenses (e.g., groceries, shopping) dominating the dataset. Accuracy alone would therefore overstate performance. Macro F1 is used to verify whether rare categories improve, while per-class precision and recall expose specific failure modes.

End-to-end qualitative checks are included because OCR noise and regex-based extraction can affect user-visible outputs even when text classification metrics appear acceptable. This ensures the pipeline is evaluated as it would be experienced in practice.

4.6.2 Spending Forecasting

Spending forecasting produces both a numeric prediction and a binary overspend warning. These outputs are evaluated separately.

Metrics

- Mean Absolute Error (MAE) for monetary predictions.
- Evaluation on both a hold-out split and a walk-forward rolling window.
- Overspend classification accuracy, precision, and recall.

Rationale MAE is chosen because it is directly interpretable in monetary terms, which is critical in budgeting contexts. Because the model operates sequentially over time, walk-forward evaluation is preferred over random cross-validation, as it better reflects real deployment.

Overspend warnings are evaluated explicitly as a classification task, since the primary user-facing risk is failing to flag an upcoming overspend rather than producing a slightly inaccurate dollar estimate.

4.6.3 Task Delegation Model

The task delegation model is evaluated both as a prediction problem and as a fairness control mechanism.

Metrics

- Classification accuracy, macro F1-score, and ROC-AUC on a household-level test split.
- Fairness deviation, measured as the absolute difference between each partner's assigned labor minutes and the target share derived from financial contribution.

- Summary statistics of fairness deviation (mean and 90th percentile).

Rationale Predictive metrics (accuracy, F1, ROC-AUC) measure how well the model replicates the teacher's assignment decisions across diverse household scenarios. However, predictive performance alone is insufficient: the core objective is to maintain equitable labor distribution over time.

The fairness deviation metric directly evaluates whether the model keeps cumulative labor within a small margin of the target share, ensuring that recommendations align with the intended fairness logic rather than drifting over the course of a month.

4.7 Technology and Key Methods

4.7.1 Model 1: Receipt Understanding

The receipt understanding subsystem transforms a raw receipt image into a structured transaction record by combining OCR, lightweight parsing, and text classification. The goal of the MVP is not full semantic understanding of receipts, but a reliable *end-to-end* pipeline that extracts the total amount and assigns the transaction to one of several expense categories.

1. OCR and preprocessing Receipt images are preprocessed (resizing, grayscale conversion, thresholding) before OCR to improve text consistency. OCR is performed using Tesseract (Smith, 2007). The pipeline returns raw and cleaned text outputs, and applies regex based heuristics to extract the total amount from common receipt patterns.

2. Text normalization and dataset construction Training data is constructed from multiple sources: personal labeled examples, synthetic category snippets that simulate merchant language variation, and optional receipt text samples from public sources when available. Normalization includes case folding, punctuation stripping, and removal of common OCR artifacts (e.g., spacing and symbol noise). This stage is treated as essential because OCR errors propagate into downstream classification.

3. Category classification Receipt categorization is implemented using a scikit-learn pipeline (Pedregosa et al., 2011):

$$\text{TfidfVectorizer}_{\text{char n-grams}} + \text{LogisticRegression}.$$

Character-level n-grams improve robustness to OCR inconsistencies and partial matches. Logistic regression is chosen for interpretability and stable training under small datasets, while still outputting probabilistic confidence scores.

4.7.2 Model 2: Spending Forecasting

The spending forecasting subsystem predicts next-month household expenditure and flags whether the household is likely to exceed its budget. It is implemented as a feature-based time series learning pipeline with two learners: a regressor for amount prediction and a classifier for overspend risk.

1. Aggregation and preprocessing Transactions are parsed and filtered to construct monthly spending totals and category level sums. This step uses structured data processing (e.g., pandas) (McKinney, 2010) to convert raw records into a forecasting table.

2. Temporal feature engineering To compensate for limited real history, the model relies on engineered features:

- lagged spending values $(\text{lag}_1, \text{lag}_2, \text{lag}_3)$,
- rolling means and trend deltas,
- volatility (recent standard deviation),
- seasonality using sine/cosine month encodings,
- budget features (rolling median baseline, safety factor, budget gap, spend-to-budget ratio).

3. Forecast and overspend models Two gradient boosting models are trained:

$$\hat{y}_{t+1} = f_{\text{reg}}(X_t), \qquad \hat{p}_{t+1}(\text{overspend}) = f_{\text{clf}}(X_t).$$

This structure produces both a continuous estimate and a discrete warning signal, which is closer to how users experience budgeting: a number is useful, but the decision to intervene requires an interpretable alert.

4.7.3 Model 3: Task Delegation and Fairness

The delegation subsystem models equitable distribution as a decision problem over time. Rather than assigning chores manually, the system recommends an assignee per task occurrence while tracking cumulative monthly effort and constraints.

Implemented approach The MVP uses **supervised imitation learning** (Osa et al., 2018): a synthetic data generator produces multi-month household logs, a teacher allocation function assigns tasks under fairness and feasibility rules, and a neural network learns to approximate these decisions. The model is implemented in PyTorch (Paszke et al., 2019).

Effort representation The conceptual model defines contribution as a weighted sum:

$$C_i(t) = w_T T_i(t) + w_F F_i(t) + w_E E_i(t) + w_M M_i(t),$$

where:
$T_i(t)$ is task effort,
$F_i(t)$ is financial contribution,
$E_i(t)$ represents fatigue, and
$M_i(t)$ represents cognitive/emotional load.

In the MVP, the representation is restricted to measurable or synthesize-able variables (task duration, availability, fatigue flags, efficiency multipliers, and cumulative monthly labor minutes). Emotional load is treated as a future extension because it is difficult to label reliably.

Fairness target Instead of an index based in variance, the MVP implements an interpretable complement rule:

$$\text{target\_share}_i = 1 - \text{financial\_share}_i.$$

This encodes a practical fairness assumption: if one partner pays less of the shared budget, they carry more labor, and vice versa. The teacher attempts to keep monthly labor minutes close to this target while honoring constraints.

Teacher logic For each task, infeasible partners are filtered (insufficient availability, temporary fatigue reductions, or feasibility issues). For feasible partners, the teacher scores assignment based on movement toward the target share:

$$\Delta_i = |(s_i + v) - t_i| - |s_i - t_i|,$$

where:
s_i is current labor share,
t_i is the target share, and
v is the normalized effort value of the incoming task.

The partner minimizing Δ_i is chosen, with an additional recent-overload penalty to avoid unfair clustering.

Learning objective The neural model predicts an assignment probability via binary classification:

$$\mathcal{L} = \mathcal{L}_{BCE} + \lambda \cdot \left| \hat{s}_i - t_i \right|,$$

where $\hat{s}_i$ is the batch-level implied labor share under predictions. This regularization encourages alignment with fairness rather than only label imitation.

4.8 Visual Interface (Supporting Prototype)

(. . . your UI description and screenshots . . .)

4.9 Application Stack and Supporting Infrastructure

(... your Firebase, Expo, Firestore section ...)

4.10 AI–Application Integration Architecture

Although the three AI subsystems can be evaluated offline, the MVP requires an integration layer that enables real user interaction, reliable storage of inputs/outputs, and controlled inference costs. To support this, the system adopts a modular cloud architecture in which the mobile application communicates with Firebase services, and inference is executed through HTTP endpoints that forward requests to the deployed ML services.

4.10.1 Integration Goals

The integration layer was designed with four objectives:

- **Separation of concerns:** the mobile application remains lightweight and does not embed model code; instead it calls stable APIs.
- **Security and access control:** user authentication and authorization is handled by Firebase Authentication and validated at the gateway layer.
- **Traceability and learning loop:** predictions, confidence scores, and user corrections are stored in Firestore so the system can later be evaluated and improved using real feedback.
- **Cost control:** inference calls are centralized so rate limiting, caching, and usage monitoring can be enforced in one place.

4.10.2 Core Components

The deployed integration pipeline consists of the following components:

- **Mobile client (React Native / Expo):** captures user actions (logging chores/expenses, uploading receipts) and renders model suggestions and snapshots in the dashboard UI.
- **Firebase Authentication:** provides identity for each request (UID) and enables household-level access control.
- **Firebase Storage:** stores receipt images as immutable input artifacts. The app uploads a receipt image and passes its download URL to the receipt understanding endpoint.
- **Firebase Cloud Functions (HTTP gateway):** exposes stable HTTP endpoints (e.g., `/receiptAnalyze`,`/forecastPredict`, `/delegationPredict`) that validate requests, normalize payloads, call the ML service, and persist results in Firestore.

- **ML service (Cloud Run / Vertex AI):** hosts the trained models behind REST endpoints. This layer executes inference and returns structured JSON outputs (predictions, probabilities, recommended assignments).
- **Firestore (system-of-record):** stores households, tasks, expenses, model runs, and user corrections. It also stores the latest delegation snapshot so the dashboard can display suggestions without recomputing on every screen load.

4.10.3 End-to-End Data Flow

Figure 1 illustrates the request and data flow. Receipt processing is representative: the client uploads a receipt to Storage, obtains a download URL, calls the HTTP gateway, and receives a structured transaction prediction. The gateway stores the ML run and optionally writes derived artifacts (e.g., expense logs) to Firestore. Similar flows exist for forecasting and delegation, which operate on historical Firestore data and write back a prediction snapshot.

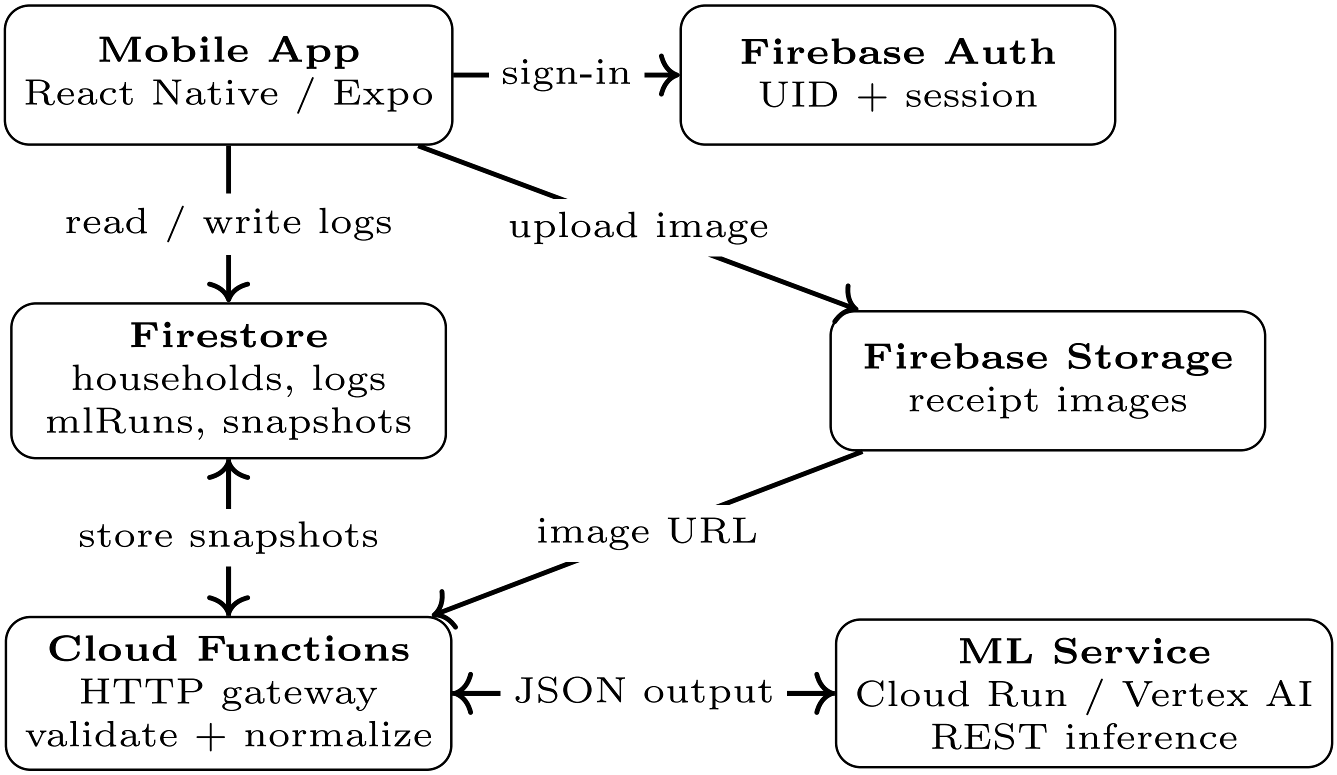

Figure 1: AI–application integration architecture (collision-safe swimlane layout).

4.10.4 Persisted ML Outputs and Feedback Loop

To support iteration and evaluation, each inference request is stored as a structured `mlRuns` record containing:

- **input:** e.g., receipt image URL, currency, household ID, time window, or delegation feature payload;
- **output:** predicted category / amount extraction, forecast values, recommended assignees, probabilities, and fairness indicators;
- **metadata:** timestamps, model version, latency, and execution source.

Additionally, when the user corrects a prediction (e.g., category override), the correction is stored alongside the original output. This creates a practical learning loop for future supervised refinement and allows longitudinal evaluation of error patterns under real usage.

4.10.5 Pragmatic MVP Strategy

For an MVP, delegation suggestions are exposed as a *cached snapshot* rather than computed on every dashboard render. The gateway writes the most recent delegation snapshot to Firestore (e.g., `households/{hid}/delegationSnapshots/latest`). The dashboard reads this document and displays suggestions above the activity feed. This reduces latency and cost, while enabling near real-time updates whenever the underlying household logs change.

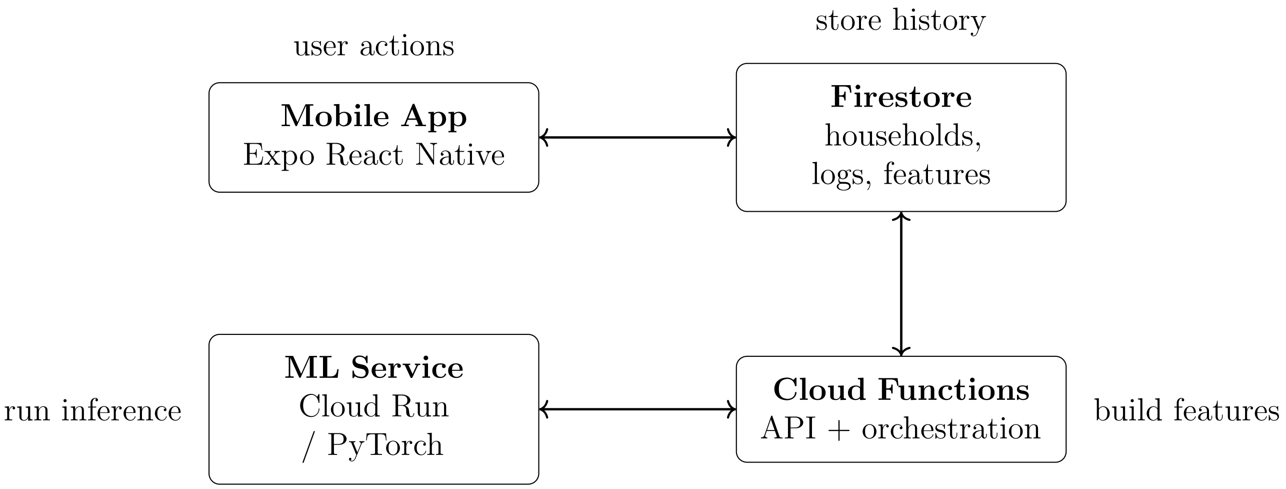

Figure 2: Live orchestration loop between the mobile app, backend services, and ML models.

5 Implementation

5.1 System Overview

The implemented system consists of three machine learning pipelines and a production grade application backend that are loosely coupled through a REST orchestration layer. The goal of this architecture is to allow each model to be developed, trained, evaluated, and deployed independently while still functioning as a coherent decision making system for home administration. All machine learning models are trained offline using Python, and are serialized as model artifacts. These artifacts are deployed behind a REST interface using Cloud Run and Vertex AI, allowing the mobile application to access model inference through standard HTTP requests.The application itself is implemented in React Native with Firebase as its backend for authentication, storage, and persistent data. The system therefore operates in three layers:

- **Client:** The mobile application, which captures user actions (expenses, tasks, receipts) and displays AI recommendations.
- **Backend:** Firebase Cloud Functions, which validate requests, construct feature payloads, call machine learning endpoints, and store inference results.
- **Machine learning:** Cloud Run and Vertex AI services that host the receipt understanding, spending forecasting, and delegation models.

This separation allows the system to be extended or retrained without breaking the application, while also enabling low-latency inference and reproducible evaluation of each model. The architecture closely follows modern machine learning system design, where inference services operate independently from application logic and persistent storage.

5.2 Receipt Understanding Pipeline

The receipt understanding pipeline transforms raw images of shopping receipts into structured financial records that can be used by the downstream forecasting and fairness model. This module is responsible for converting noisy, unstructured visual data into reliable numerical and categorical features.

The pipeline is composed of four stages: OCR and text extraction, amount parsing, category classification and model training and persistence.

5.2.1 OCR and Text Extraction

Receipts are first captured as images through the app and transmitted to the backend for processing. The system applies an Optical Character Recognition engine to convert the visual content into raw text.

Due to the variability of receipt layouts, fonts, lighting conditions, and camera angles, the extracted text often contains noise, misaligned tokens, and irrelevant symbols. To handle this, the output is normalized through a preprocessing pipeline that removes non-alphanumeric artifacts, merges fragmented tokens, and reconstructs coherent text lines.

The result of this stage is a cleaned textual representation of the receipt that includes merchant names, dates, line items, and monetary values, which serves as the input for subsequent financial parsing and classification.

5.2.2 Amount Parsing

After extraction, the system identifies and extracts the monetary value associated with each transaction. Unlike simple keyword matching, this stage applies rule-based filtering combined with numeric pattern recognition to detect valid currency amounts.

Candidate numbers are detected using regular expressions and contextual cues (e.g. proximity to terms such as ”total”, ”amount”, or currency symbols). When multiple candidates are present, a ranking heuristic is applied based on numeric magnitude, text position, and semantic markers to select the most likely transaction amount. This approach allows the system to reliably extract correct totals even when receipts contain multiple prices, taxes, discounts, or partial subtotals.

5.2.3 Category Classification

To assign semantic meaning to each expense, the system employs a supervised text classification model. The cleaned receipt text is encoded using a transformer-based language model, which captures semantic relationships between merchants, item names, and spending categories.

The classifier is trained to map receipt text to predefined financial categories such as groceries, transport, dining, utilities, etc. This allows heterogeneous and noisy merchant descriptions to be mapped into consistent, structured labels.

The resulting category is attached to each transaction and used by the financial forecasting and fairness models to understand spending patterns and allocate financial responsibility across household members.

5.2.4 Model Training and Persistence

The receipt classifier is trained on a labeled dataset of receipt texts and associated spending categories. Training data includes both real receipts and synthetically augmented examples to improve robustness to spelling errors, formatting variation, and merchant name diversity.

The trained model is serialized and stored in persistent storage and deployed as a cloud-hosted inference service. This allows the app to submit receipt images and receive categorized financial records in real time.

By persisting the model and decoupling training from inference, the system supports feature training as new data is collected, enabling continuous improvement without disrupting app availability.

5.3 Spending Forecasting Pipeline

The spending forecasting pipeline is responsible for transforming historical transaction data into predictive signals about future financial behavior. While the receipt understanding module structures individual purchases, this component models temporal household dynamics, enabling the system to anticipate cash-flow patterns and detect financial risk.

5.3.1 Transaction Aggregation

All categorized transactions produced by the receipt understanding pipeline are stored in the household database and aggregated over time. Transactions are grouped by temporal windows (e.g., daily, weekly, and monthly) and by semantic categories such as groceries, transportation, and discretionary spending.

This aggregation step produces time series representations of household spending, allowing the model to observe trends, seasonality, and changes in consumption behavior. Aggregated features form the numerical backbone of the forecasting model and allow the system to compare recent activity against long-term baselines.

5.3.2 Feature Engineering

From the aggregated transactions, the system computes a set of financial features designed to capture household behavior. These include: • Total spending per period • Category-level spending proportions • Rate of spending change over time • Budget utilization ratios • Short-term vs long-term spending averages

These features allow the model to distinguish between normal variation (e.g., weekly grocery cycles) and anomalous patterns (e.g., sudden increases in discretionary expenses). By encoding both absolute and relative spending signals, the feature set enables robust generalization across different household sizes and income levels.

5.3.3 Regression Model

A supervised regression model is trained to predict future household spending based on historical features. Given a sequence of past financial states, the model estimates the expected total expenditure for upcoming periods.

This forward-looking prediction allows the system to anticipate whether the household is on track to remain within budget or is trending toward financial stress. The regression output is continuously updated as new transactions arrive, ensuring that predictions adapt to evolving household behavior.

5.3.4 Overspend Classification

In addition to predicting continuous spending values, the system includes a binary overspending classifier that evaluates whether the predicted future expenditure exceeds safe budget thresholds.

This classifier takes as input the regression output and contextual financial features, and outputs a risk signal indicating whether the household is likely to overspend. This signal is used by the task delegation model to adjust workload distribution and by the application interface to trigger warnings and recommendations.

By combining numerical forecasting with discrete risk classification, the system is able to provide both precise financial projections and actionable alerts, enabling proactive household financial management.

5.4 Task Delegation and Fairness Model

The task delegation model ensures that household duties are distributed efficiently, adaptively, and in a socially fair way. Unlike conventional task schedulers, this model combines financial contributions, task effort, priority, frequency, and personal availability to optimize both the distribution of work and the financial participation of all household members.

The model is trained using a combination of synthetic data generation, rule based supervision, and fairness optimization.

5.4.1 Synthetic Household Generator

To train the delegation model, a synthetic household generator is used to produce diverse simulated household scenarios. Each generated household consists of two agents with attributes such as available time, income, energy levels, task preferences, and historical task completion patterns.

A predefined catalog of household responsibilities is sampled, with each task characterized by effort, frequency, and priority. Financial contributions are incorporated to represent varying levels of economic pressure.

This simulation framework allows the system to observe a wide range of household configurations and ensures that the model generalizes beyond the limited data available during early deployment.

5.4.2 Rule-Based Teacher Policy

A *rule-based* teacher policy is used to generate high quality supervisory signals for the learning model. This teacher encodes human designed principles of fairness and practicality, such as:

- Tasks should be distributed in proportion to available time
- Financial contributors should not be disproportionately burdened with unpaid labor
- High effort or high frequency tasks should be balanced across agents

Given a synthetic household state, the teacher computes an idealized task allocation based on these rules. These allocations serve as target outputs for the neural network, enabling the model to learn how to replicate and generalize human fairness heuristics.

5.4.3 Neural Network Architecture

The delegation model is implemented as a neural network that maps household state representations to task assignment probabilities. Inputs include financial contributions, partner availability, task attributes, and historical workload distributions.

These features are embedded into a shared latent space and processed through multiple fully connected layers to capture nonlinear interactions between money, time, and labor. The output layer produces a distribution over which household member should perform each task.

This architecture enables the model to adapt to complex tradeoffs, such as assigning more tasks to a partner during periods of lower financial contribution or shifting workload when availability changes.

5.4.4 Fairness-Aware Loss Function

To ensure that the model produces socially acceptable and stable allocations, training is guided by a fairness-aware loss function. In addition to minimizing prediction error against the teacher policy, the loss incorporates penalties for imbalance in workload and misalignment between financial contribution and assigned labor.

This encourages the network to converge toward solutions that are not only accurate but also equitable. As a result, the system avoids degenerate solutions in which one household member is systematically overburdened, even when short term optimization pressures exist.

By embedding fairness directly into the learning objective, the model produces task schedules that are both mathematically optimized and socially sustainable.

5.5 AI + Application Integration

The three machine learning pipelines should be integrated into a live mobile application through a cloud-based service architecture. This allows the system to operate continuously with real household data while maintaining scalability, reliability, and responsiveness.

Each model is deployed as an independent inference service and orchestrated through a central application backend, enabling the household agent to act as a unified decision making system.

5.5.1 Request–Response Flow

The application is designed so that all household intelligence is derived from a shared, continuously updated state composed of financial transactions, member availability, and the household task catalog.

When receipts are processed, the resulting structured transactions are stored as part of the household's financial history. These records are intended to be used by the task delegation model to estimate each household member's financial contribution and economic context when generating task assignments. This allows responsibility allocation to reflect not only time and availability, but also the real balance of monetary input between partners.

The spending forecasting model is architecturally designed as a complementary budgeting and risk-prediction layer. Although it is not yet active in the live request flow, it is planned to consume the same transaction history in order to generate cash-flow projections and overspending alerts. These signals will inform both users and the task delegation model, enabling proactive financial and workload adjustment.

When a task schedule is requested, the application will gather the current household state—including financial transactions, member availability, and the predefined task catalog—and send it to the task delegation service. The model will return a proposed allocation of responsibilities, which will then be rendered in the user interface.

This planned request–response architecture ensures that perception (receipts), economic context (transactions and forecasts), and decision-making (task delegation) operate over a unified household state, enabling the system to function as a coherent intelligent agent once fully integrated.

5.5.2 Snapshot Storage and Caching

To ensure consistency and reduce redundant computation, the system stores periodic snapshots of household state, including financial aggregates, forecasts, and task assignments. These snapshots are cached in the backend and referenced by subsequent model calls.

By persisting intermediate results, the application avoids unnecessary recomputation while ensuring that users always interact with a stable, explainable view of the household's current status. This also enables historical analysis, allowing users to review how financial and workload decisions evolved over time.

5.5.3 Real Time User Interface Updates

The application uses real time data synchronization to reflect AI decisions immediately in the user interface. When new receipts are processed or new task assignments are generated, the updated state is pushed to all connected devices.

This allows both partners to see the same financial forecasts, task schedules, and fairness metrics simultaneously, eliminating ambiguity or conflicting views of household responsibilities.

6 Evaluation

6.1 Evaluation Framework

The evaluation of the proposed system is conducted in two complementary stages: **offline model validation and deployment oriented generalization analysis**. This structure reflects how real world systems are developed and assessed, where models are first validated under controlled conditions and later adapted to live environments.

In the offline phase, each machine learning component is evaluated independently using held-out test sets and synthetic simulations. This includes the receipt understanding model, the spending forecasting model and the task delegation model. These evaluations measure predictive accuracy, robustness, and fairness using standard quantitative metrics such as accuracy, F1 score, mean absolute error (MAE), ROC-AUC, and fairness deviation. Offline evaluation ensures that each model meets a baseline level of performance before being integrated into a live application.

In parallel, the system is evaluated at the architectural level as multi-component intelligent agent. Rather than treating the models as isolated predictors, the evaluation considers how their outputs propagate through the full pipeline: OCR and classification feed financial forecasts, which in turn inform task delegation and fairness calculations. This reflects the operational reality of the system, where small errors in perception or prediction can influence downstream household decisions.

Because the system is designed for deployment in real households, the evaluation framework explicitly distinguishes between training and test distributions and future real-world usage. The models are trained primarily on labeled data and synthetic simulations to ensure coverage across diverse household scenarios. As a result, offline metrics primarily measure generalization across simulated environments rather than memorization of a small number of real users.

Finally, the evaluation framework includes an analysis of expected behavior after deployment. This examines how performance may change when the system transitions from synthetic and curated data to real household inputs, accounting for cold-start effects, data drift, and evolving user behavior. By combining controlled evaluation with deployment-aware analysis, the framework provides a realistic and scientifically grounded assessment of the system's performance and reliability.

6.2 Receipt Understanding Evaluation

The receipt understanding pipeline is evaluated in terms of its ability to accurately convert real-world receipt images into structured financial records. This evaluation focuses on both semantic classification performance and the robustness of the OCR-based text extraction, as errors in either stage directly impact downstream financial forecasting and task delegation.

The evaluation is conducted on a held-out validation set of labeled receipt data, ensuring that the reported results reflect generalization to unseen receipts.

6.2.1 Classification Metrics

The receipt category classifier is evaluated using accuracy and macro-averaged F1 score. On the held-out validation set, the model achieves an overall accuracy of 0.867 and a macro F1 score of 0.678.

Performance is strongest for frequent categories such as groceries and shopping, where the model benefits from a larger number of labeled examples and more consistent merchant patterns. Lower scores are observed for sparse categories such as rent and utilities; however, this is primarily due to class imbalance rather than instability in the learned representations.

The relatively high macro F1 score indicates that the model maintains reasonable performance across categories rather than simply exploiting dominant classes, which is critical for reliable downstream financial aggregation.

6.2.2 OCR Error Analysis

To evaluate the robustness of the OCR pipeline, a manual end-to-end assessment was performed on receipt images from the SROIE dataset. Extracted totals and merchant information were compared against human-labeled ground truth.

The OCR pipeline accurately recovers total transaction values for standard receipt formats and remains robust under moderate image noise and layout variation. In more ambiguous cases, such as receipts with overlapping prices or unconventional layouts, the system produces lower confidence scores, typically around 0.6, reflecting appropriate uncertainty rather than confident misclassifications.

Despite OCR noise, the downstream category classifier remains stable, as the transformer-based embeddings are able to recover semantic meaning even when individual tokens are imperfectly recognized. This demonstrates that the receipt understanding pipeline is resilient to realistic image quality variations commonly encountered in mobile applications.

6.3 Spending Forecasting Evaluation

The spending forecasting pipeline is evaluated in terms of both numerical accuracy and its ability to detect financial risk. Because the system is designed to support household decision-making rather than only produce point estimates, the evaluation focuses on predicting spending trends and identifying overspending conditions.

Evaluation is performed using a combination of held-out time periods and rolling walk-forward validation to simulate realistic deployment conditions.

6.3.1 MAE and Trend Accuracy

On a held-out period, the regression model achieves a Mean Absolute Error (MAE) of approximately $82,916. Inamorestringentwalk-forwardevaluationacross25rollingsplits, themodelyieldsanaverageMAEof$

While these absolute values appear large, they are driven by synthetic months in which total household spending exceeds 1 million. When normalized by monthly spending, the relative error remains within the 1–5

In addition to point-wise accuracy, the model consistently predicts the correct direction of spending change (increase or decrease), demonstrating that it learns meaningful behavioral patterns rather than simply regressing to the mean.

6.3.2 Overspend Detection Performance

The overspending classifier is evaluated on its ability to identify months in which projected spending exceeds safe budget thresholds. On a single held-out period, the classifier achieves an accuracy of 66.7

The improvement under rolling evaluation indicates that the model becomes more reliable as more historical context is available, reflecting realistic deployment behavior in which forecasting improves over time.

These results show that the forecasting pipeline provides a reliable early-warning signal, allowing the system to adjust task delegation and alert users before financial instability occurs.

6.4 Task Delegation and Fairness Model

The task delegation model is evaluated in terms of both predictive accuracy and fairness. Because the system is designed to produce socially sustainable task allocations rather than merely replicate a teacher policy, evaluation considers not only correctness but also alignment between financial contribution and assigned labor.

Evaluation is performed on a household-wise held-out test set of synthetic household scenarios that were not seen during training.

6.4.1 Prediction Accuracy

The neural delegation model is compared against the rule-based teacher policy used during training. On the held-out test set, the model achieves an accuracy of 0.815, an F1 score of 0.815, and a ROC-AUC of 0.899.

These results indicate that the network successfully learns to reproduce and generalize the fairness principles encoded in the teacher policy, while maintaining strong discrimination between alternative task assignments.

The high ROC-AUC score demonstrates that the model produces well-calibrated probability distributions over task assignments, allowing the system to reason about uncertainty and trade-offs when multiple allocations are plausible.

6.4.2 Fairness Deviation

Fairness is evaluated by measuring the deviation between each household member's predicted share of labor and their target share derived from financial contribution and availability.

The model achieves a mean absolute deviation of approximately 2.5 minutes between predicted and target monthly labor allocation. This represents a substantial improvement over a logistic regression baseline, which exhibits significantly larger deviations and systematic imbalance.

By minimizing fairness deviation in addition to prediction error, the system ensures that task schedules remain stable, balanced, and socially acceptable over time.

6.5 System Level Performance

The proposed system is evaluated not only as a collection of individual models, but as a coherent, multi-stage decision pipeline. System-level performance focuses on stability, consistency, and suitability for real-time household use.

Each machine learning component is deployed as an independent cloud-hosted service and orchestrated through a central backend. This modular architecture ensures that failures or delays in one component do not compromise the integrity of the entire system. Snapshot caching further guarantees that downstream models always operate on a consistent view of the household state, preventing race conditions or contradictory task assignments.

Inference latency is bounded by the slowest model in the pipeline, but because receipt processing, forecasting, and task delegation are invoked asynchronously and cached, the system remains responsive for user interaction. Financial forecasts and task schedules are reused across sessions until new transactions or availability updates occur, allowing the system to scale efficiently without recomputing every model call.

At the system level, the architecture supports continuous operation, real-time synchronization across devices, and explainable state transitions, which are essential for household coordination rather than one-off predictions.

6.6 Failure Modes and Limitations

Despite strong offline performance, the system has several inherent limitations. The receipt understanding pipeline depends on image quality and receipt formatting, and extreme OCR errors can propagate into financial forecasts. While downstream models are robust to moderate noise, very poor image capture may still require manual correction.

The spending forecasting model is sensitive to limited historical data. In newly created households, financial predictions may initially exhibit higher variance due to the absence of long-term behavioral patterns. Similarly, the task delegation model relies on accurate estimates of availability and preferences, which may be noisy or incomplete when users first join the system.

Another limitation arises from the use of synthetic data for training the delegation model. While synthetic households provide broad coverage of possible scenarios, they cannot fully capture the complexity of real human relationships, emotional labor, or unexpected life events.

Finally, the system assumes cooperative household members. Strategic manipulation of reported availability or financial input could degrade fairness estimates, although this risk is mitigated through transparency and shared state visibility.

6.7 Expected Performance After Development

Once the system is connected to a live application and begins receiving real household data, its behavior is expected to evolve in a predictable way.

Initially, the forecasting and delegation models will operate in a cold-start regime, where predictions are primarily guided by learned priors from synthetic and historical data. During this phase, the models may exhibit mild underfitting, producing conservative or generic recommendations until sufficient household-specific data is collected.

As real transaction histories, availability patterns, and task completion data accumulate, the models are expected to rapidly specialize to each household. Forecasting accuracy should improve as seasonal trends and spending habits are learned, while the delegation model will adapt to the unique balance of income, effort, and preferences within each household.

Importantly, the system is designed to avoid overfitting by maintaining regularization through fairness constraints and by continuously retraining on a mixture of real and synthetic data. This ensures that learned household-specific patterns do not destabilize long-term fairness or financial safety.

Overall, deployment is expected to shift the system from a general household coordinator to a personalized household intelligence, improving both accuracy and user trust over time.

7 Conclusion and Future Work

7.1 Summary of Contributions

This paper presented a novel AI-driven household coordination system that integrates computer vision, natural language processing, financial forecasting, and fairness-aware learning into a unified intelligent agent. Unlike traditional budgeting tools or task managers, the proposed system models the household as a dynamic socioeconomic system, where money, time, effort, and fairness are jointly optimized.

The work contributes three major machine learning pipelines: a receipt understanding model that converts real-world purchases into structured financial data, a spending forecasting model that anticipates future financial risk, and a fairness-aware task delegation model that allocates domestic labor in a socially sustainable way. These components are deployed through a modular cloud-based architecture designed for real-time use in a mobile application.

7.2 System Capabilities and Achievements

The system demonstrates that it is possible to automate large portions of household administration that are typically handled through manual tracking, emotional negotiation, and invisible labor. By combining perception, prediction, and decision-making into a single pipeline, the system is able to continuously adapt to changing household conditions.

Offline evaluation shows that the receipt classifier achieves high accuracy, the forecasting model reliably predicts spending trends and overspending risk, and the task delegation model produces fair and balanced workload distributions. Together, these results indicate that the system is capable of supporting both financial stability and social fairness within a household.

Beyond technical performance, the system represents a new class of domestic AI: one that does not merely provide recommendations, but actively coordinates shared responsibilities in a transparent and data-driven manner.

7.3 Limitations

Despite its strong offline performance, the system has several limitations. The receipt understanding pipeline depends on image quality and may be affected by poorly captured or unusually formatted receipts. The financial forecasting model requires sufficient historical data to produce reliable predictions, making early-stage households more prone to uncertainty. The task delegation model is trained primarily on synthetic household data, which, while diverse, cannot fully capture the complexity of real human relationships and life events.

In addition, the system assumes cooperative participation. Misreporting availability, ignoring task assignments, or withholding financial information can reduce the effectiveness of fairness and forecasting mechanisms.

7.4 Future Improvements

Future work will focus on transitioning the system from offline validation to continuous online learning. As real households use the application, models can be fine-tuned on live data to improve personalization, reduce bias, and better capture individual behavioral patterns.

Additional improvements include reinforcement learning for long-term household optimization, integration of external financial data sources such as bank feeds, and expansion to multi-household and family-scale coordination. More advanced fairness metrics and conflict-resolution strategies could further enhance social stability and user trust.

References

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: Forecasting and control* (5th ed.). Wiley. https://onlinelibrary.wiley.com/doi/book/10.1002/9781118619193

Brotheridge, C. M., & Lee, R. T. (2003). Development and validation of the emotional labour scale. *Journal of Occupational and Organizational Psychology*, *76*(3), 365–379. https://doi.org/10.1348/096317903769647229

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Craig, L., & Brown, J. E. (2017). Feeling rushed: Gendered time pressure and leisure inequality. *Journal of Marriage and Family*, *131*(2), 225–242. https://doi.org/10.1111/jomf.12320

Daminger, A. (2019). The cognitive dimension of household labor. *American Sociological Review*, *84*(4), 609–633. https://doi.org/10.1177/0003122419859007

Daniels, A. K. (1987). *Invisible work*. University of California Press. https://doi.org/https://doi.org/10.2307/800538

DeVault, M. L. (1991). *Feeding the family: The social organization of caring as gendered work*. University of Chicago Press. https://onlinelibrary.wiley.com/doi/abs/10.1525/si.1992.15.4.529

E., A. (2025). Cognitive household labor: Gender disparities and consequences for maternal mental health and wellbeing. *Arch Womens Ment Health*. https://doi.org/10.1007/s00737-024-01490-w.

Gopher, D. (1994). Analysis and measurement of mental load. In R. Parasuraman & D. R. Davies (Eds.), *Handbook of perception and action, volume 2: Cognitive processes and performance* (pp. 265–291). Academic Press.

Haupt, A. (2024). The gendered division of cognitive household labor, mental load, and family–work conflict in european countries. *European Societies*, *26*(3), 828–854. https://doi.org/https://doi.org/10.1080/14616696.2023.2271963

Hochschild, A. R. (1983). *The managed heart: Commercialization of human feeling*. University of California Press.

Hochschild, A. R. (2003). *The commercialization of intimate life: Notes from home and work*. University of California Press.

Huang, Y., Lv, T., Cui, L., Lu, Y., & Wei, F. (2022). LayoutLMv3: Pre-training for document AI with unified text and image masking. *arXiv*. https://arxiv.org/abs/2204.08387

Kim, G., Hong, T., Kang, B., Kim, J., & Roh, S. (2022). Donut: Document understanding transformer without OCR. *Advances in Neural Information Processing Systems*. https://doi.org/https://doi.org/10.48550/arXiv.2111.15664

Lee, Y.-S., & Waite, L. J. (2005). Husbands' and wives' time spent on housework: A comparison of measures. *Journal of Family Issues*, *26*(6), 328–336. https://doi.org/10.1111/j.0022-2445.2005.00119.x

Lim, B., Arik, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, *37*(4), 1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012

McKinney, W. (2010). Data structures for statistical computing in python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th python in science conference* (pp. 51–56). https://doi.org/10.25080/Majora-92bf1922-00a

Morini, C. (2007). The feminization of labour in cognitive capitalism. *Feminist Review*, *87*, 40–59. https://www.jstor.org/stable/30140799

Offer, S. (2014). The costs of thinking about work and family: Mental labor, exhaustion, and gender inequality. *Social Forces*, *93*(1), 163–191. https://doi.org/10.1111/socf.12126

Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., & Peters, J. (2018). An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, *7*(1–2), 1–179. https://doi.org/10.1561/2300000053

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*. https://papers.nips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*, 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html

Pinedo, M. (2016). *Scheduling: Theory, algorithms, and systems* (5th ed.). Springer. https://www.researchgate.net/publication/225017349_Scheduling_Theory_Algorithms_And_Systems

Smith, R. (2007). An overview of the Tesseract OCR engine. *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, 629–633. https://doi.org/10.1109/ICDAR.2007.4376991

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press. https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf

Walzer, S. (1998). *Thinking about the baby: Gender and transitions into parenthood*. Temple University Press. https://doi.org/10.2307/2654405

Yavorsky, J. E., Kamp Dush, C. M., & Schoppe-Sullivan, S. J. (2015). The production of inequality: The gender division of labor across the transition to parenthood. *Journal of Marriage and Family*, *77*(3), 662–679. https://doi.org/10.1111/jomf.12189