

Project Motivations

- Canadians are feeling more in debt than ever, with 90% claiming to have more debt today than they did five years ago [1].
- Only 47% of Canadians plan their spending based on actual budgets due to how tedious bookkeeping could be [1].
- Accounting is an essential method to enhance the personal experience of financial management.

- Canadians are feeling more in debt than ever, with 90% claiming to have more debt today than they did five years ago [1].
- Only 47% of Canadians plan their spending based on actual budgets due to how tedious bookkeeping could be [1].
- Accounting is an essential method to enhance the personal experience of financial management.

Project Objectives

- Implement a mobile application that optimizes the process of expense management by automating tedious aspects of bookkeeping.
- Provide a personalized experience and an overview of the user's financial situations by conducting statistical analysis.
- Apply Computer Vision techniques to recognize details of the receipts to record purchase history.

- Implement a mobile application that optimizes the process of expense management by automating tedious aspects of bookkeeping.
- Provide a personalized experience and an overview of the user's financial situations by conducting statistical analysis.
- Apply Computer Vision techniques to recognize details of the receipts to record purchase history.

Major Advantages

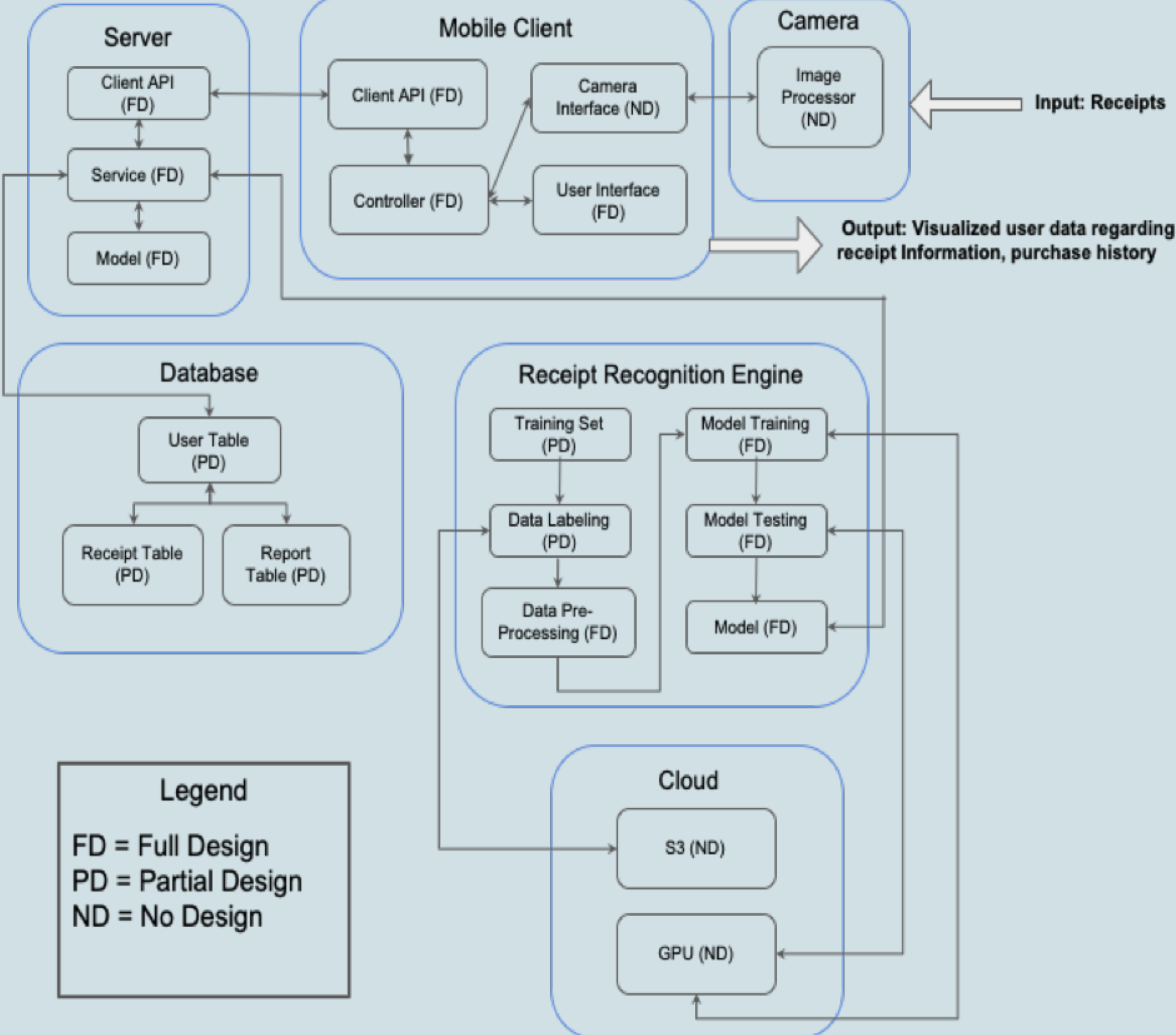
- Provide a single source of user purchase history and robust yet scalable data security.
- Display overviews of expenses over arbitrary periods of time to help users monitor their financial status.
- Accessible from multiple platforms including websites and android applications.
- Provide the solution to longer-than-usual receipts by allowing taking multiple pictures and stitching them together.

Acknowledgements & References

[1] Visa Canada, “Budgeting,” *Practical Money Skills*, Jun. 2019; <https://practicalmoneyskills.ca/personalfinance/savingspending/budgeting/>

[2] H. Saleh, "MVVM architecture, ViewModel and LiveData (Part 1)," Medium, 13-Oct-2018: <https://proandroiddev.com/mvvm-architecture-viewmodel-and-livedata-part-1-604f50cda1>.

Block Diagram



Design Alternatives

Subsystem	Option 1	Option 2	Reasoning
User Interface	MVVM	MVC	<ul style="list-style-type: none">• High degree of decoupling [2].• Easy to reuse.
Database	MySQL	PostgreSQL	<ul style="list-style-type: none">• Lightweight and memory-efficient.• Excellent scalability.
Receipt Recognition Engine	DNN	CNN	<ul style="list-style-type: none">• More suitable for image processing.• Higher accuracy and stability.
Backend	Node.js	Ruby on Rails	<ul style="list-style-type: none">• Lower difficulty of deployment.• Better performance.

Implementations

- User interface uses the Model-View-ViewModel architecture to ensure high maintainability and extensibility.
- Backend is written in Node.JS and APIs are implemented according to the requirements of REST architecture. All access to APIs are required to contain a valid JSON Web Token to prevent hacking attempts.
- The image recognition engine uses artificial neural network that applies convolutional operations for image processing tasks, applied to character classification to transfer images to text.
- Transformation functions such as Gaussian are applied to detect edges and locate characters.

Analysis Results

- 90%+ Recognition Accuracy

```
Data processing finished
Train on 50596 samples, validate on 5622 samples
Epoch 1/100
50596/50596 [=====] - 6s 112us/step - loss: 1.5028 - acc: 0.6151 - val_loss: 0.4173 - val_acc: 0.8598
Epoch 2/100
50596/50596 [=====] - 5s 101us/step - loss: 0.4597 - acc: 0.8347 - val_loss: 0.3088 - val_acc: 0.8903
Epoch 3/100
50596/50596 [=====] - 5s 101us/step - loss: 0.3573 - acc: 0.8707 - val_loss: 0.2667 - val_acc: 0.9006
Epoch 4/100
50596/50596 [=====] - 5s 107us/step - loss: 0.3066 - acc: 0.8851 - val_loss: 0.2553 - val_acc: 0.8983
Epoch 5/100
50596/50596 [=====] - 5s 105us/step - loss: 0.2753 - acc: 0.8931 - val_loss: 0.2329 - val_acc: 0.9055
Epoch 6/100
50596/50596 [=====] - 5s 103us/step - loss: 0.2525 - acc: 0.8994 - val_loss: 0.2284 - val_acc: 0.9091
Epoch 7/100
50596/50596 [=====] - 5s 104us/step - loss: 0.2371 - acc: 0.9046 - val_loss: 0.1994 - val_acc: 0.9216
Epoch 8/100
```

- < 10s Recognition Time

Experiment No.	Total Character Count	Total Recognition Time (s)
1	385	6.5
2	277	4.38
3	482	10.3
4	420	9.8
5	318	6.7
Average	N/A	7.536

- 99% Segmentation Accuracy

Experiment No.	Total Character Count	Segmented Count	Segmenting Accuracy
1	385	381	0.9896
2	277	277	1.0000
3	482	479	0.9938
4	420	417	0.9929
5	318	310	0.9748
Average	1882	1864	0.9904