

Final Report Cover Sheet

Subject Code: CSIT321

Subject Name: Project

Submission Type: Final Report Specifications

Project Title: FaceCard

Student/Team Name: Digital Solutions Group

Student Number: 7309314, 7422878, 8108328, 7329672, 7395656

Student Phone/Mobile No. 0562541888

Student E-mail: Yakik660@uowmail.edu.au

Lecturer Name: Dr. Haitham Yaish

Due Date: March 27th,2025

Date Submitted: March 27th,2025

<p>PLAGIARISM:</p> <p>The penalty for deliberate plagiarism is FAILURE in the subject. Plagiarism is cheating by using the written ideas or submitted work of someone else. UOWD has a strong policy against plagiarism.</p> <p>The University of Wollongong in Dubai also endorses a policy of non-discriminatory language practice and presentation.</p> <p>PLEASE NOTE: STUDENTS MUST RETAIN A COPY OF ANY WORK SUBMITTED</p>	<p>DECLARATION:</p> <p>I/We certify that this is entirely my/our own work, except where I/we have given fully-documented references to the work of others, and that the material contained in this document has not previously been submitted for assessment in any formal course of study. I/we understand the definition and consequences of plagiarism. We/I declare that the project proposal has not been used in any UOWD courses before and that this project idea is a total new idea of the project team</p> <p>Signature of Student:</p>
--	--

Optional Marks:

Comments:

Lecturer Project Proposal Receipt (To be filled in by student and retained by Lecturer upon return of assignment)

Subject:

Project Title:

Student / Team Name:

Student Number:

Due Date:

Date Submitted:

Signature of Student:

Student Project Proposal Receipt (To be filled in and retained by Student upon submission of assignment)

Subject:

Project Title:

Student/Team Name:

Student Number:

Due Date:

Date Submitted:

Signature of Lecturer:

Introduction.....	5
Purpose:.....	5
Scope:.....	5
FEASIBILITY ANALYSIS AND PLANNING.....	7
Competitor Analysis.....	7
Our Target niche.....	10
SWOT Analysis.....	12
System Prerequisites:.....	13
System Limitations:.....	13
SOFTWARE AND HARDWARE REQUIREMENTS.....	14
External Interface Requirements:.....	14
User interface:.....	14
Hardware interface:.....	15
Software interface:.....	15
Communication interface:.....	15
Libraries and Languages Required:.....	16
DESIGN OF THE SYSTEM.....	17
System Architecture.....	17
Business Workflow Diagram.....	18
Activity Diagram.....	19
High Level Use Case Diagram.....	20
Domain Model Diagram:.....	24
UI Flow Diagram:.....	25
Implementation.....	32
TEST PLANS.....	35

(Github Link: <https://github.com/Yahia-K/FaceCard> invitation was sent to drhyaish)

Introduction

Purpose:

Payments in schools are an untapped field and have not seen much innovation since the idea of adding funds to an ID card was implemented. However, this suffers from the same issue as cash: being physically tangible, which runs the risk of being lost and/or stolen.

The FaceCard project will make school cafeteria transactions easier and more secure; therefore, upon purchasing something from the cafeteria, it identifies the students (mainly kindergarten to primary school, ages 3–9) and approves the payment via facial recognition. The important objectives of the projects are reducing cash handling, giving parents real control over spending in real-time, and making the cafeteria experience a whole lot better for the students. One of our other main goals is to reduce overall parental anxiety regarding their children's spending in schools and ensuring they make correct decisions regarding their dietary preferences and allergies.

Scope:

The FaceCard system allows for a biometric, cashless payment in the school cafeteria; proposed to let students make purchases by recognizing their faces. The advanced computer vision combined with a web application to grant control to parents leaves the results of cash handling coupled with no insights for parents far behind.

The web application allows parents to sign in and add their children, along with an initial scan of their face to save it, which can then be used at school cafeterias to authorize and ensure their identity. It is at this point from the employee's point of

view from their login page that they will be able to view the child's information retrieved from the scan, and add items to their basket. These items will then be checked with the child's preferences to ensure that they do not purchase any products that include any ingredients they are allergic to. Parents will receive a live update of any purchases made by the children they have added to their list.

FEASIBILITY ANALYSIS AND PLANNING

Competitor Analysis

	PayFace	PayByFace	Fujitsu	Liv. Lite	Traditional cash
Target niche	General Stores	Partner merchants	healthcare, finance, public infrastructure (ATMs, payment kiosks, secure login systems)	kids and teens making purchases in variable markets	majority of the world accepts cash
Biometric focus	Facial features	Facial Features	Palm vein and facial features	none, tangible	none, tangible

Country limitations	Brazil	Romania and Bulgaria	Global	UAE	Global
----------------------------	--------	----------------------	--------	-----	--------

Payface

Payface originated in Brazil with the aim of improving payment processes by means



of facial recognition. Customers can make payments in brick-and-mortar stores solely through facial recognition, eliminating the need for cards or phones. They focus on security by using encryption and adhering to data privacy rules like LGPD (Brazilian General Data Protection Law). The company has participated in international expansion opportunities, including a program in Qatar, and is actively expanding throughout Latin America. They are also focused on increasing mobile and digital payment functionalities.

Limitations

Payface struggles from certain limitations such as limited merchant availability and adoption, as it is heavily reliant on partnering with retailers, meaning it is limited to Brazil.

Payface's Target Niche

Payface primarily focuses on Brazilian supermarkets and pharmacies. However, they have been expanding throughout Latin America and Qatar.

PayByFace



PayByFace was established in 2018 and started operations primarily in Romania and Bulgaria. It offers contactless biometric payment solutions, enabling users to link their face to a payment account. Once registered, users can make payments by simply scanning their face at partner merchants. PayByFace focuses on customer loyalty and aims to reduce checkout friction.

Limitations

Much like Payface, PayByFace also struggles with regional restrictions as the system is not publicly available for public use, as it is mainly used in Romania and Bulgaria. The system also struggles with a major security risk as scans are done by means of phone and tablet cameras, which can be accessed by various users, including those of ill intent.

PayByFace's Target Niche

The company has collaborated with payment platforms like SafexPay to expand into India and other regions. It also participated in Visa Innovation Programs, helping to refine its technology. PayByFace is conducting pilot projects across Europe, the UAE, and India to further its reach.

Fujitsu

Fujitsu has created a range of biometric solutions, such as palm vein recognition and facial recognition systems. Biometric technologies are frequently incorporated into ATMs, payment kiosks, and secure login systems. The palm vein system, in particular, has become



popular due to its high accuracy and contactless operation, addressing hygiene concerns post-pandemic.

Limitations

Fujitsu is highly dependent on its biometric scanning hardware which is expensive to deploy and highly specialized. The use of palm vein and facial recognition can also cause challenges as businesses will need to invest and pay for training and infrastructure.

Fujitsu's Target Niche

Fujitsu's solutions extend beyond retail payments into sectors such as healthcare, finance, and public infrastructure. They focus heavily on enterprise-level deployments, offering robust security measures and scalable technology.

Our Target niche

What sets FaceCard apart from its competitors is the market it is set in. As stated, we are aiming the product at parents with children aged 3 to 9 in schools. More specifically, those who are paying for lunches and snacks during their recess times.

Our target audience consists of:

Students - As our biggest priority and the largest part of our niche, we aim to make the students' experience smooth and efficient. They also would not have to worry about losing any cash which would ease their anxiety and stress levels.

Guardians/ Parents - The second largest part of our niche is the guardians which will provide the funding for the students. They will have their own web application in

which they could link their Stripe account which will allow for the students to pay for their food and beverages. They will also be able to set their own limits to monitor expenditure and select any allergies or dietary preferences.

Schools – the niche that includes all of the students is the school the aforementioned students go to. The system will be installed in their cafeteria and their database can be used for integration purposes. They would also be receiving payment from the cafeteria purchases.

SWOT Analysis

<u>STRENGTHS</u>	<u>WEAKNESSES</u>
<p>Preference management - The ability to update and manage preferences for each student such as allergies will be an option for guardians.</p> <p>Transaction Logging - A table will be stored containing purchase history of students</p> <p>Expenditure Limiting - Guardians will be able to limit the amount of money a child can spend</p>	<p>Hardware Dependent - FaceCard heavily relies on biometric scanning equipment (cameras) and cannot function without it</p> <p>Training Costs- The introduction of a new system would mean that employees would require training and there will also be maintenance costs.</p>
<u>OPPORTUNITIES</u>	<u>THREATS</u>
<p>Expansion into other niches - This technology could also be implemented in other areas such as gyms, supermarkets, pharmacies, etc., which is dominated by competitors.</p> <p>Introducing a wallet - A proposed improvement on the system involves the usage of a standalone e-wallet within the guardians' application</p> <p>Governmental Support - Potential support from a local or international government to deploy the system on a larger scale in public schools.</p>	<p>Privacy Concerns - The usage of facial recognition for students could raise some concerns with guardians as they may not consent to having their child's face in the system, raising ethical concerns.</p> <p>Law Changes - General Data Protection Laws and such may change over time, as AI laws change and vary vastly as the systems improve and become more advanced.</p> <p>Security Threats - Valuable information such as face encodings and account information could be subject to a data breach. This may be concerning as parental accounts are linked to a payment gateway. However, this is not much of a concern with a web application.</p>

System Prerequisites:

- **High-quality Cameras:** Cameras with high resolution to capture faces in various lighting conditions.
- **Server Infrastructure:** servers to handle facial recognition processing and ensure real-time response.
- **Parental Consent:** Schools and parents would be required to give consent for facial recognition usage to ensure privacy and ethical concerns have been addressed.
- **School Collaborations:** Agreements with schools to implement the system. Access to their existing database in order for admins to approve and double check any student additions. Note that FaceCard's database does not need to be part of the school database.
- **Payment Processors:** Integration with payment service provider (Stripe) to ensure secure transaction processing.
- **Network Infrastructure:** High-speed, secure network connections to enable real-time communication between devices and the system.

System Limitations:

- **Data Security Concern:** Biometric data is very sensitive, any potential breach could have serious consequences. The system's design must include robust security measures in order to prevent unauthorized access or misuse.
- **Facial Recognition Accuracy:** Children's facial features evolve more rapidly, and movement during scanning could cause identification errors. This might lead to incorrect transactions or failure in identification. A solution to this could be the use of a high definition camera with lighting, and updating student's facial embeddings scannings every semester (or every 6 months).

- **Hardware and Network Requirements:** High-quality cameras and a robust network are essential for the system to function efficiently. Schools with inadequate infrastructure may struggle to implement the system.
Network outages or slow connections could cause delays in processing payments or verifying students which would affect overall usability.
- **User Adoption:** Getting schools, parents, and students to adopt a new payment method would be challenging. Some may prefer traditional methods due to concerns over privacy, security, or technical failure.

- **Time:** Having to complete the project within a certain time frame requires cutting down on several features and work in order to present the prototype on time.

SOFTWARE AND HARDWARE REQUIREMENTS

External Interface Requirements:

User interface:

The user interface (UI) of the system software provides parents with services such as recharging for their children, obtaining their children's purchase information, and making special notes for their children. On the other hand, during the purchase stage, the employee obtains facial recognition authentication regarding the child and their information through the camera and completes the purchase.

Hardware interface:

The system not only needs a high-definition camera to obtain the child's facial information, it also needs a high-configuration computer. The high-definition camera assists the system to complete the rapid scanning and obtain information. Preventing the computer from being overloaded by the database and preventing the computer from giving wrong recognition matches is key. To help resolve the wrong recognition issue, a high definition camera with lighting attached would be ideal to ensure the most accurate results possible.

Software interface:

Facial recognition payment requires the database subsystem to communicate with the payment gateway. This step can be achieved by sending a command of user ID and recognition parameters through the module. The database interacts with the neural network and completes the verification, items are added to the basket, and finally the payment gateway confirms to complete the shopping. The database returns information including the deducted amount, remaining amount and time, student ID and connection to the parent, which is stored in the user account.

Communication interface:

The facial recognition payment system uses a secure HTTPS protocol for data transmission to protect the user's personal data and payment information. The payment gateway and database communicate with the system through an encrypted API interface to ensure the security of payment information transmission. As a result, the payment result and account information are displayed immediately after the user scans the face, achieving a fast and smooth payment process.

Libraries and Languages Required:

Python – well-suited for quickly writing and testing code, also it can be used for payment verification logic and interface interactions (such as connecting to payment platform APIs), providing simplicity and maintainability.

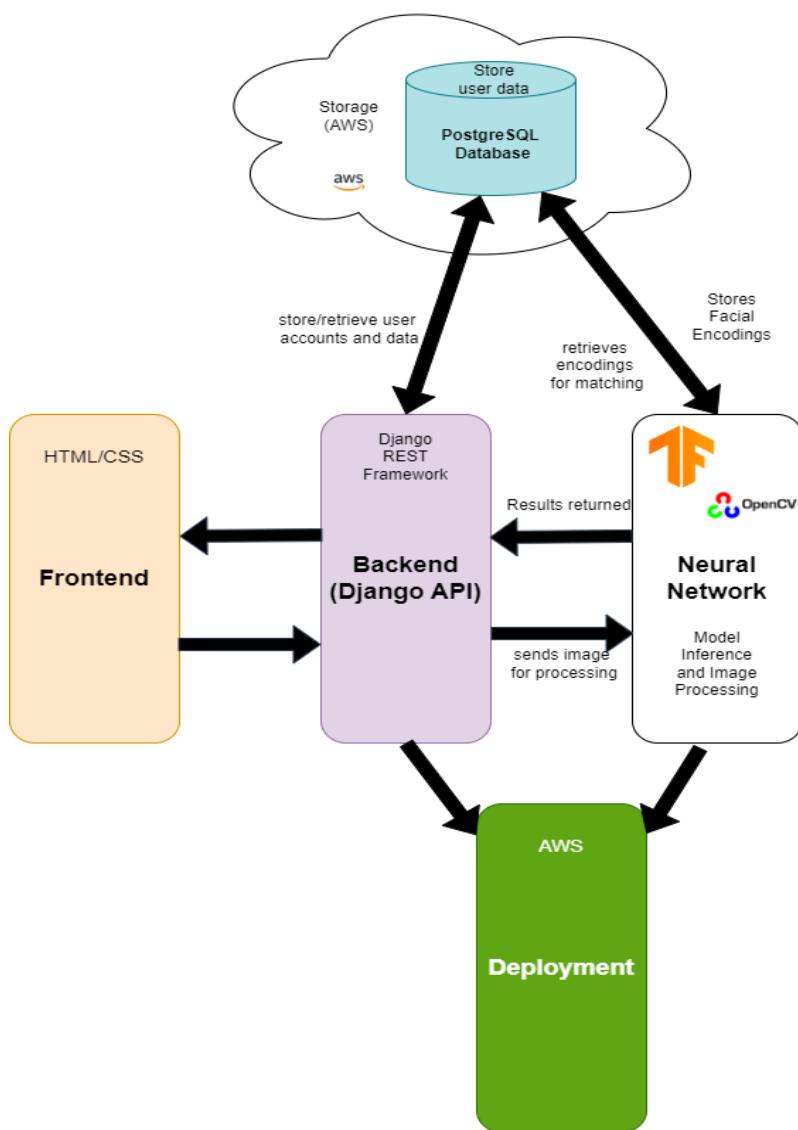
PostgreSQL – PostgreSQL can efficiently handle and query large volumes of user data while maintaining data security and integrity. It records each recognition and payment operation, ensuring comprehensive data logging for auditing and tracking purposes.

Django ORM – Using Django as an ORM streamlines backend development by enabling efficient database management through Python instead of raw SQL. It ensures data security and integrity while facilitating seamless querying. Django plays a crucial role in building the backend, integrating all modules, including the neural network for facial recognition and the payment gateway for secure transactions. It also logs and manages all database operations, ensuring reliable auditing and tracking.

OpenCV Library – Used to capture the user's facial image, process images (such as scaling and grayscale conversion), and detect facial key points. OpenCV can be combined with Python to implement core facial recognition features, supporting image recognition across different programming environments.

DESIGN OF THE SYSTEM

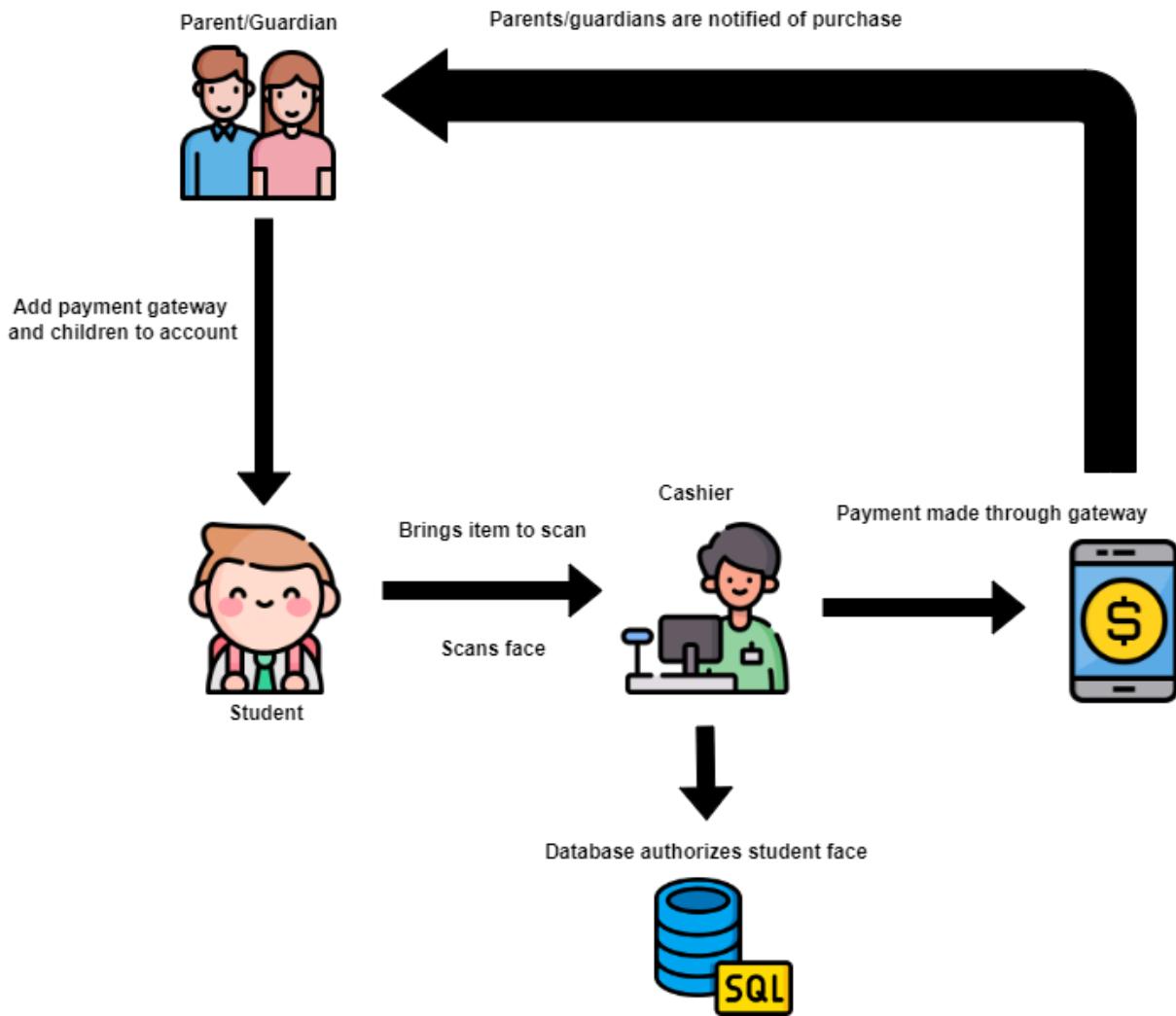
System Architecture



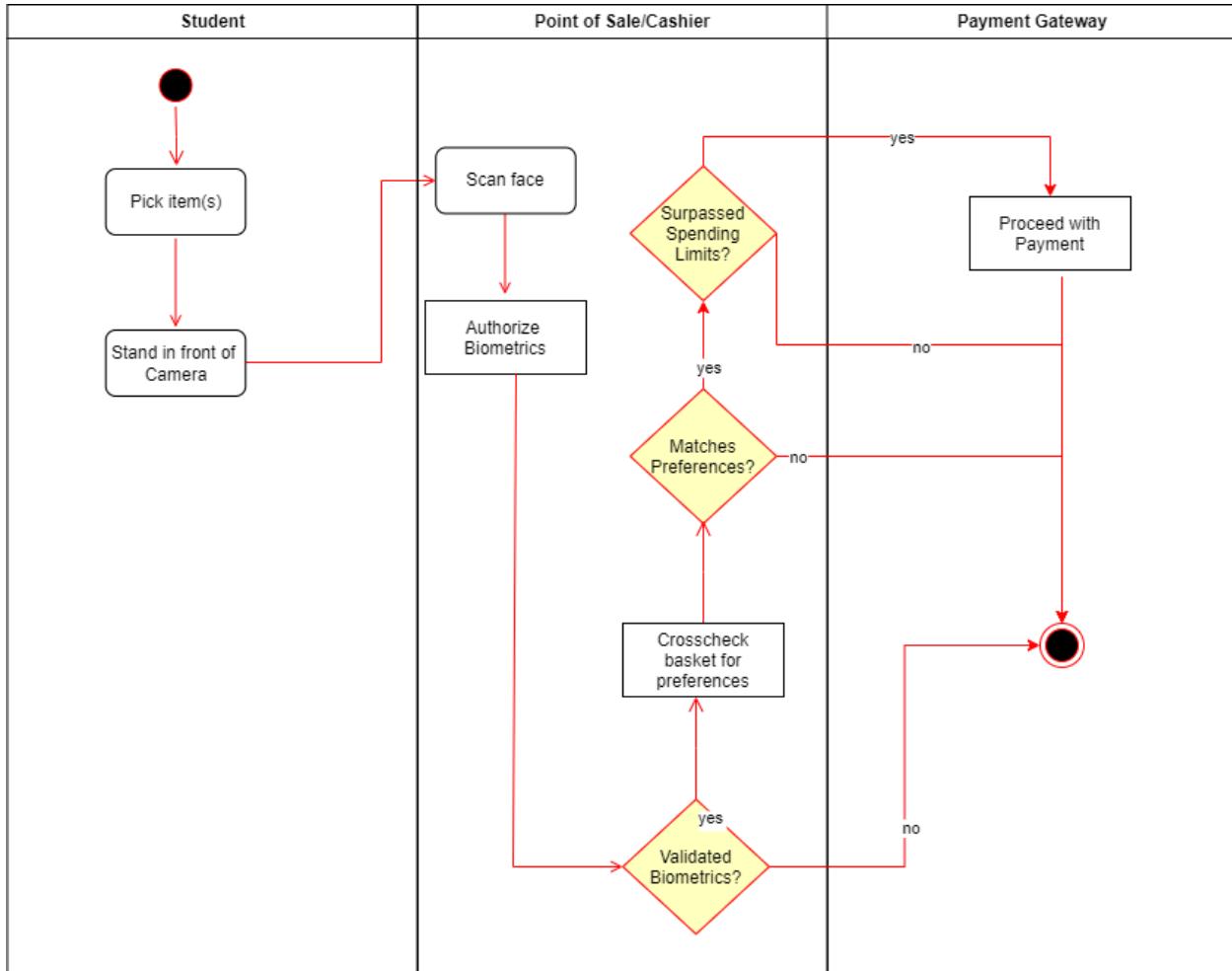
This is the finalized system architecture. The ORM used for this project is Django and a web application is created, so we have a frontend based off HTML and CSS which is connected to the backend that uses the Django REST framework. The backend is

connected to the Neural Network model which is responsible for facial recognition and image processing, and the PostgreSQL database which contains facial embeddings and user login information for parents and employees. In the future when deployment is done, it will be hosted on AWS, which will take from the backend and the neural network model.

Business Workflow Diagram

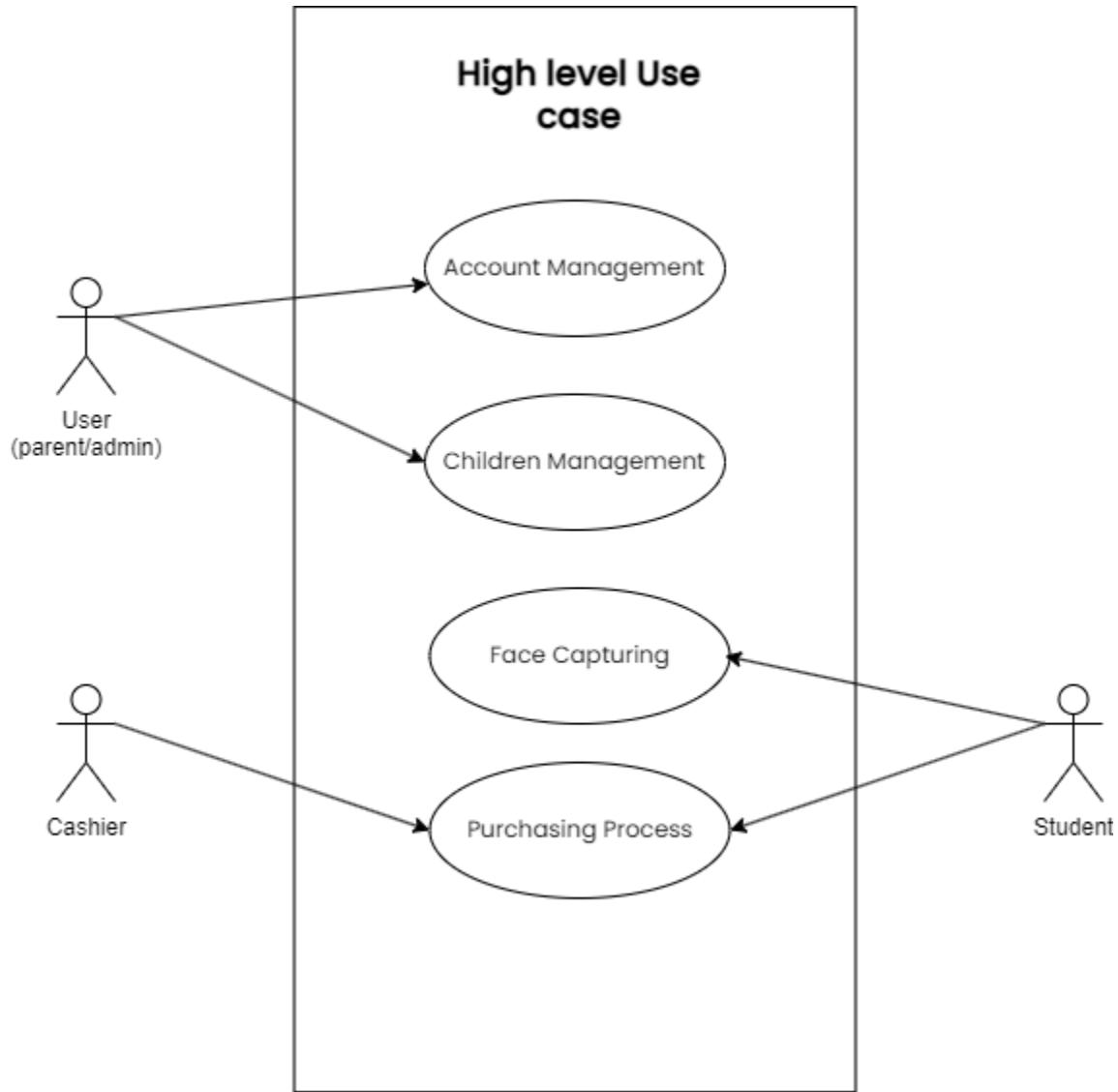


Activity Diagram



The activity diagram follows the series of events taken for students to pick an item that they want to buy, have their face scanned and make the purchase. At the point of sale, a series of checks to see if the face scan is authorized, if any allergies or dietary preferences are met, or if the spending limit has been reached. If all checks have been passed, the purchase is then made.

High Level Use Case Diagram



The diagram above explains the simpler use cases. However there are more that need to be taken into consideration as they are functions of the system:

- Create account (parent or worker)
- Delete account
- Update account (parent or worker)
- Account Login (parent or worker)

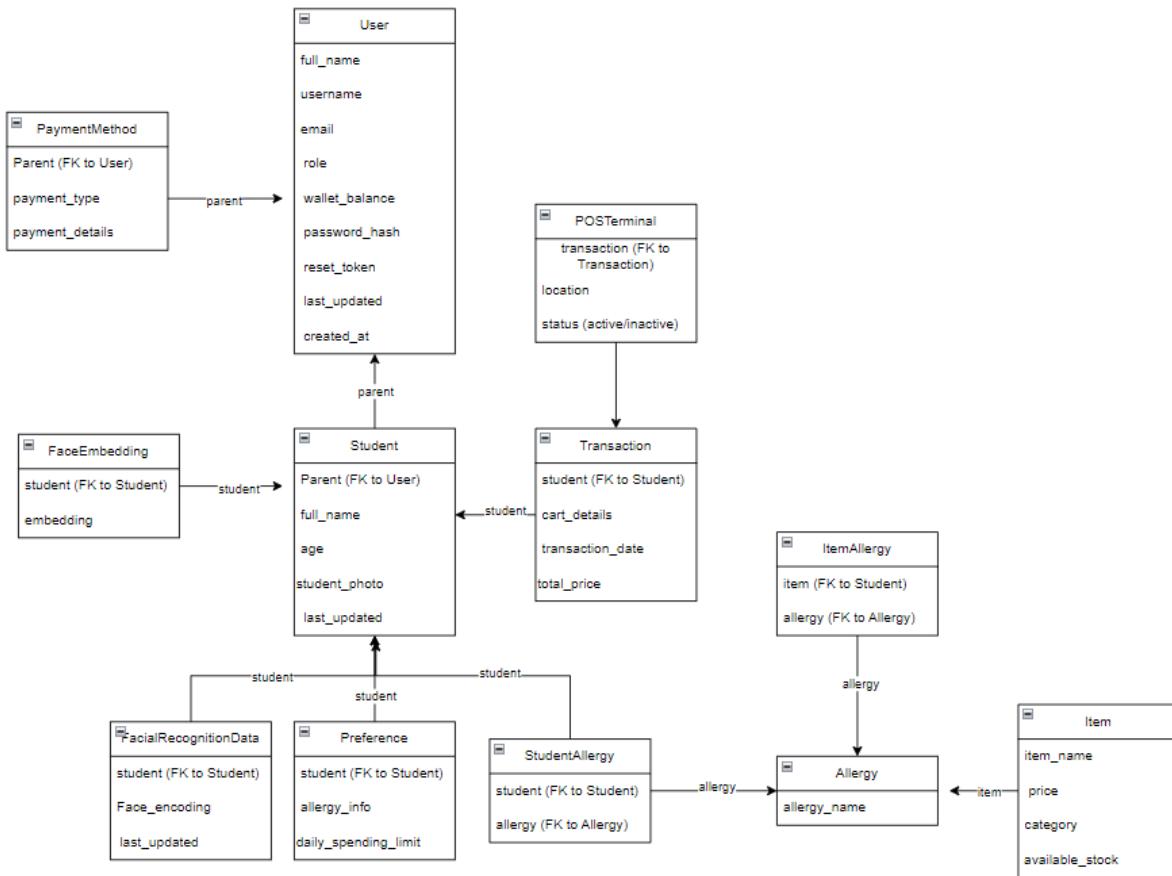
- Account Logout
- Add children/students
- Remove children/students
- List children/students
- Recharge account
- Manage preferences (allergies, food preferences etc.)
- Retrieve payment history
- Scan face
- Record face
- Authorize face
- Make payment
- Update biometric information
- Check credentials
- Add item to cart

Functions	Parent	Admin	Student	Cashier
<i>Accounts</i>				
Create account	*	*		*
Remove account	*	*		*
Update account	*	*		*
Check credentials	*	*		*

Login	*	*		*
Logout	*	*		*
Add student	*	*		
Remove student	*	*		
List students	*	*		
Recharge account	*	*		
Manage preferences	*	*		
List transaction history	*	*		*
<i>Facial Recognition</i>				
Scan face	*		*	*
Record face				
Authorize face				*

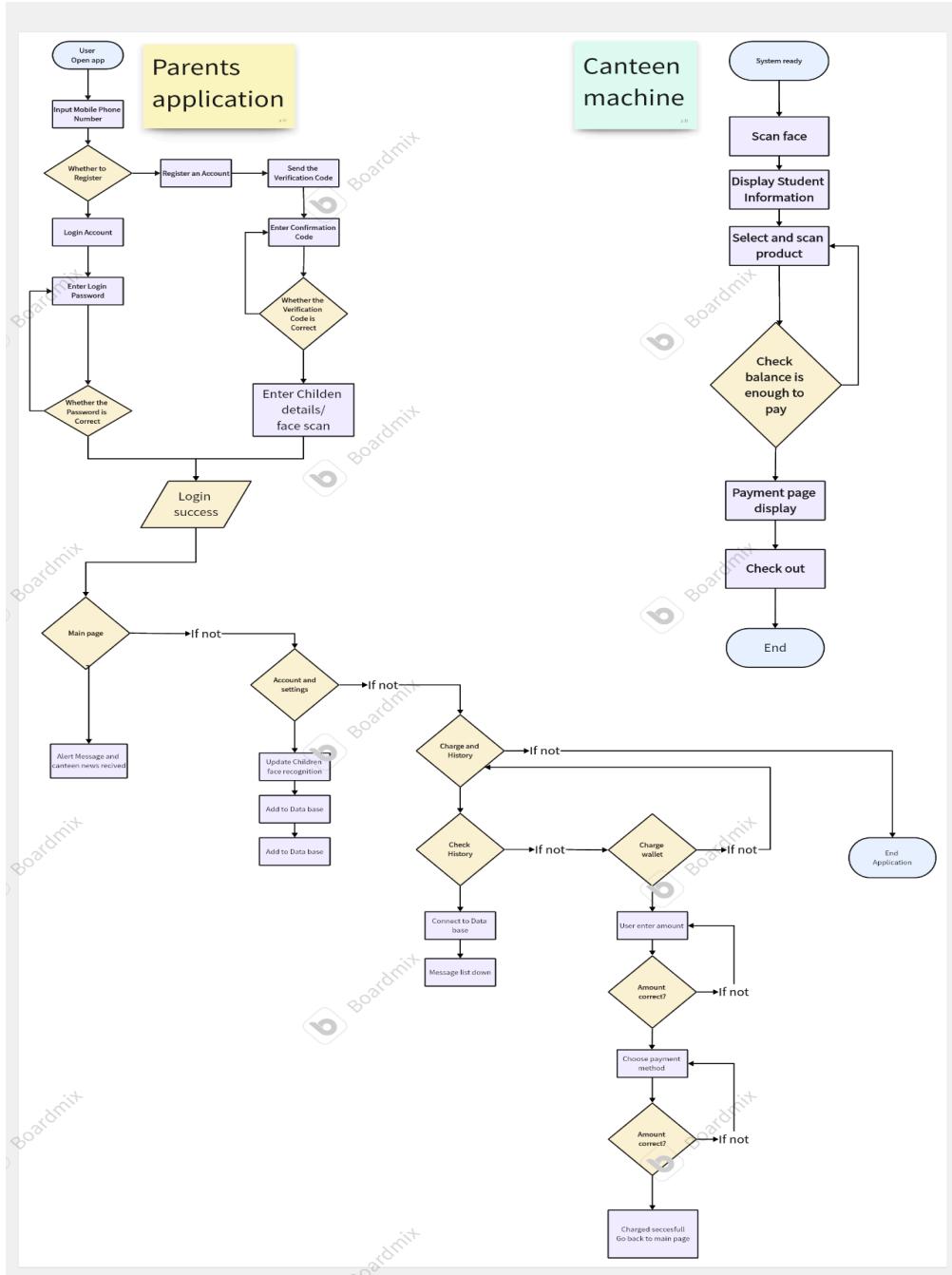
Update biometric info	*	*		
<i>Product related</i>				
Add item to cart				*
Make payment	*		*	*

Domain Model Diagram:



The above Model explains the tables used in the database along with their relations to each other. The allergies are connected to the items and students so they can be checked to see if they exist (in order to stop the students from buying products they are allergic to). Facial embeddings and the data from the neural network are related to the students, while the payment gateways are related to the users (parents in this case).

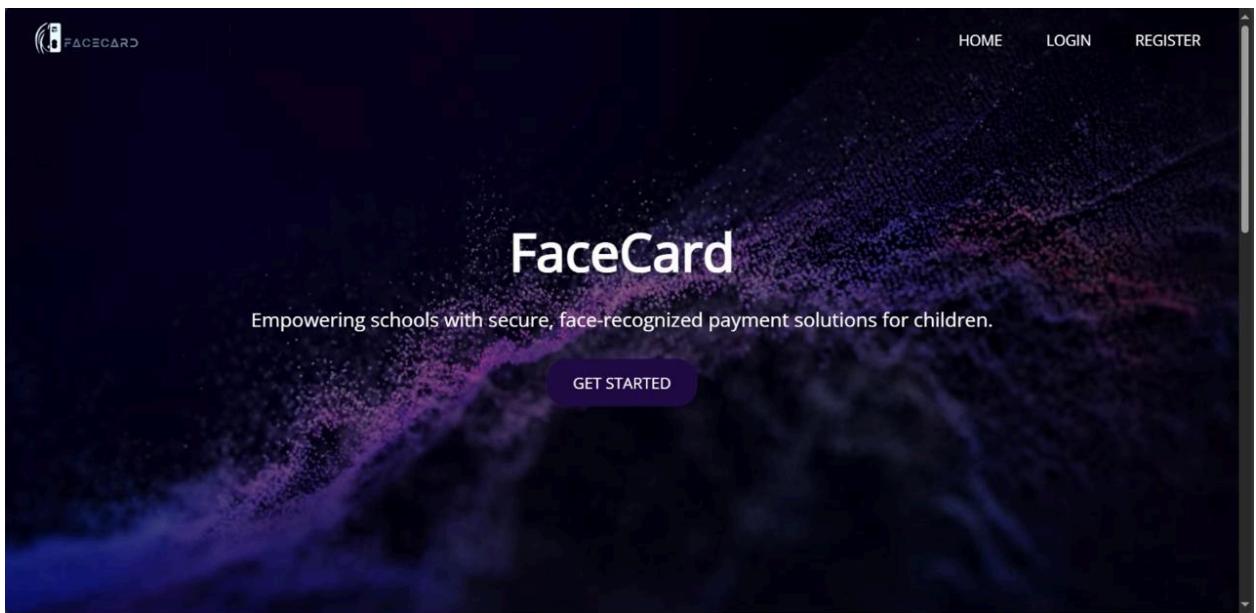
UI Flow Diagram:



The flow diagram above explains the processes that the system goes through in order to achieve the desired optimal result (success). Much like the activity diagram, the flow diagram shows the series of checks done in order to ensure

authorization, allergies and spending limits, while also checking for the payment gateway to make sure it is up and running.

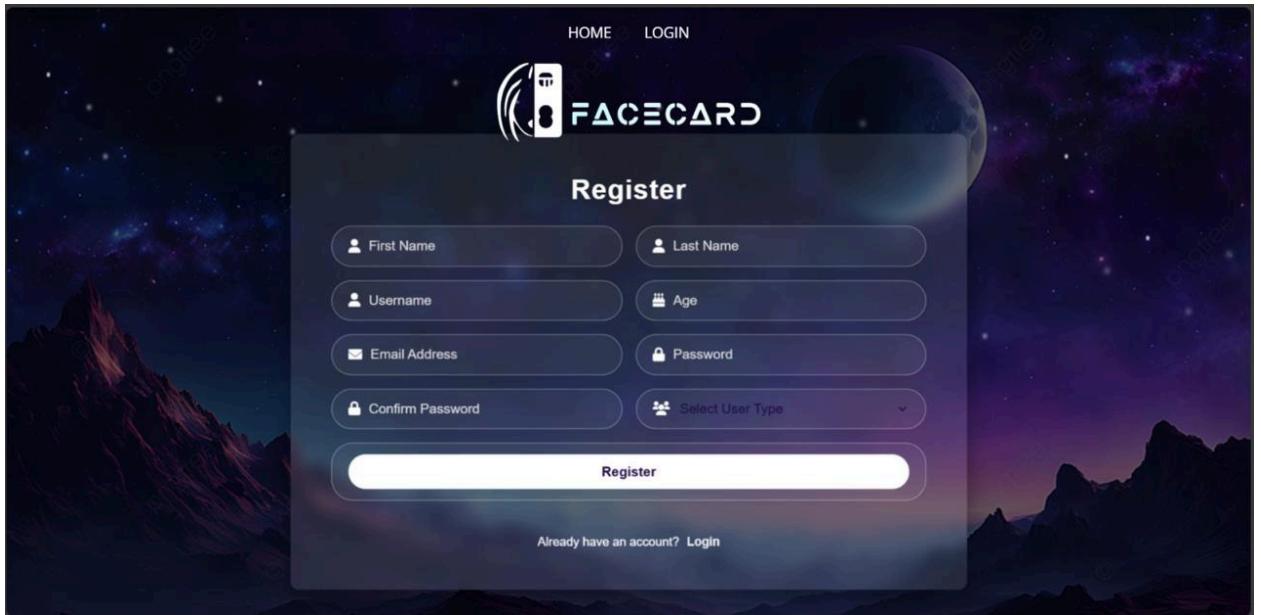
Home Page

The image shows the FaceCard features page. At the top left is the FaceCard logo. The main heading "Features" is centered above four feature boxes. Each box contains an icon and a title followed by a brief description. The features are: "Secure Transactions" (lock icon), "Face Recognition" (person icon with a checkmark), "School Integration" (school building icon), and "Allergy Management" (hand icon).

- Secure Transactions**
Advanced security measures keep every payment safe.
- Face Recognition**
Fast and reliable facial verification technology.
- School Integration**
Seamless integration in school environments for ease of use.
- Allergy Management**
Tailored features to ensure the safety of children with allergies.

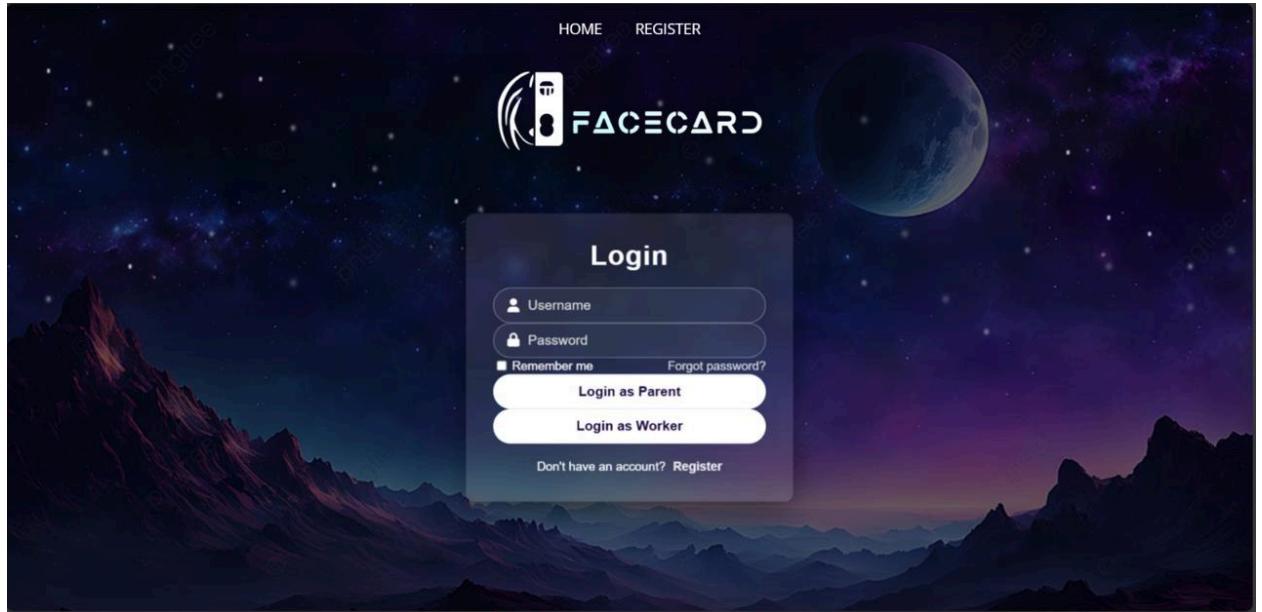
The initial page that users will see when accessing our web app, they have the options to navigate to a login or registration page, or to scroll down and learn more about us and our product.

Registration Page



Here users can register as a parent or worker by inputting their basic details and creating a password.

Login Page



Once an account is registered users simply input their username and password and will be redirected to their respected worker or parent pages depending on their account type.

Add Child Page

The image shows the 'Add Child' page of the FaceCard application. On the left, a sidebar displays a 'Parent Avatar' of a person named 'John Doe', along with navigation links for 'Home', 'Add Child', and 'Log Out'. The main content area is a form for adding a child. It includes fields for 'Name' (with placeholder 'Enter child's name'), 'Student ID' (with placeholder 'Enter Student ID'), and 'Age' (with placeholder 'Enter age'). Below these are sections for 'Select Child Allergies' and 'Any Other Allergies'. Under 'Select Child Allergies', there is a list of checkboxes for various allergens: Peanuts, Tree nuts, Milk, Eggs, Fish, Soy, Wheat, Honey, Strawberries, and Sesame. Under 'Any Other Allergies', there is a text input field with the placeholder 'Type here...'. At the bottom right of the form is a blue 'Add Child' button.

On this page parents can add their child as a subsidiary to their account, inputting their child's allergies as well as other relevant information.

Worker/Cashier Account Page

The screenshot shows a user interface for a worker/cashier account. On the left is a sidebar with a profile picture of a person with short brown hair, a dashboard link, a Go to POS link, and a Log Out link. The main content area has a welcome message "Welcome, bob marley". Below it are two sections: "NOTIFICATION" and "PROFILE & MESSAGES". The notification section contains a blue box titled "News letter" with text about facial scanning. The profile section shows basic information: Name: bob marley, Username: bob123, Email: bob@hotmail.com. Below these are two tables: "RECENT TRANSACTIONS" and "CHARGE HISTORY". The recent transactions table lists three purchases: 1. Chips at 3.00 AED on 2025/1/10, 2. Chicken Rice at 10.00 AED on 2025/1/9, and 3. Water at 2.00 AED on 2025/1/9. The charge history table shows two entries: 1. A successful transaction of 150.00 AED on 2025/1/5, and 2. A failed transaction of 150.00 AED on 2025/1/5.

#	Time	Product Name	Price (AED)
1	2025/1/10	Chips	3.00
2	2025/1/9	Chicken Rice	10.00
3	2025/1/9	Water	2.00

#	Time	Charge Balance	Process
1	2025/1/5	150.00	Success
2	2025/1/5	150.00	Fail

Here is the worker dashboard where the workers information and prior transactions are shown.

Parent Dashboard

Welcome, simba king

Wallet Balance: AED 91.00 [Recharge](#)

Your Children

Child Photo	Name	Age	Spending Limit (AED)
	tester	22	0
	shaqir	27	0
	husain	19	0

Orders & Payment History

Date	Child	Item	Amount (AED)
25 Mar 2025, 06:27	husain	Water (AED 2) apple juice (AED 2) 7DAYS Strawberry Bread (AED 4)	AED 8.00
24 Mar 2025, 15:25	shaqir	Water (AED 2) apple juice (AED 2)	AED 4.00

Notifications

Reminder: Please update your child's facial data by next month for seamless payments.

Quick Contacts

School Admin: +971 501234567

This is what a parents dashboard looks like, where their children are shown as well as their respective transaction histories and along with their wallet balance and the ability to top up said wallet balance.

Point of Sale Screen

Select Customer: [-- Choose Customer --](#) [Face Scan for Customer](#)

Customer Details

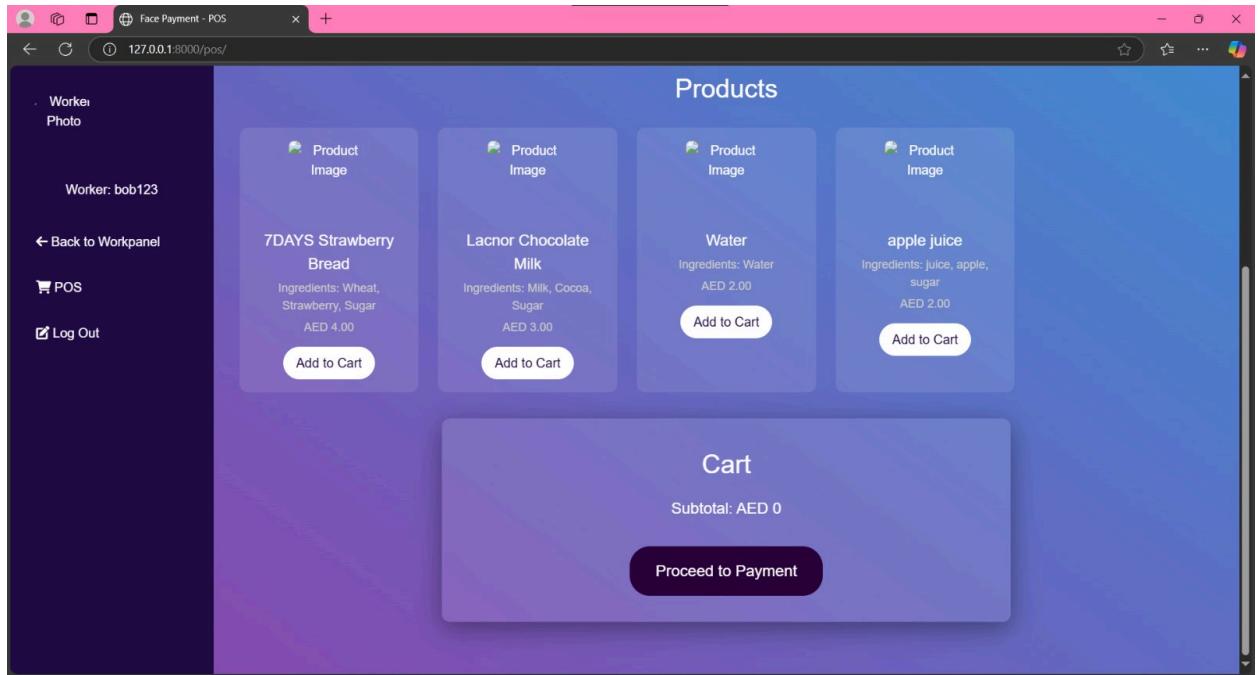
Student ID: N/A
Name: N/A
Wallet Balance: AED 0
Allergies: None

Products

Product Image	Product Name	Ingredients	Price (AED)	Action
	7DAYS Strawberry Bread	Ingredients: Wheat, Strawberry, Sugar	AED 4.00	Add to Cart
	Lacnor Chocolate Milk	Ingredients: Milk, Cocoa, Sugar	AED 3.00	Add to Cart
	Water	Ingredients: Water	AED 2.00	Add to Cart
	apple juice	Ingredients: juice, apple, sugar	AED 2.00	Add to Cart

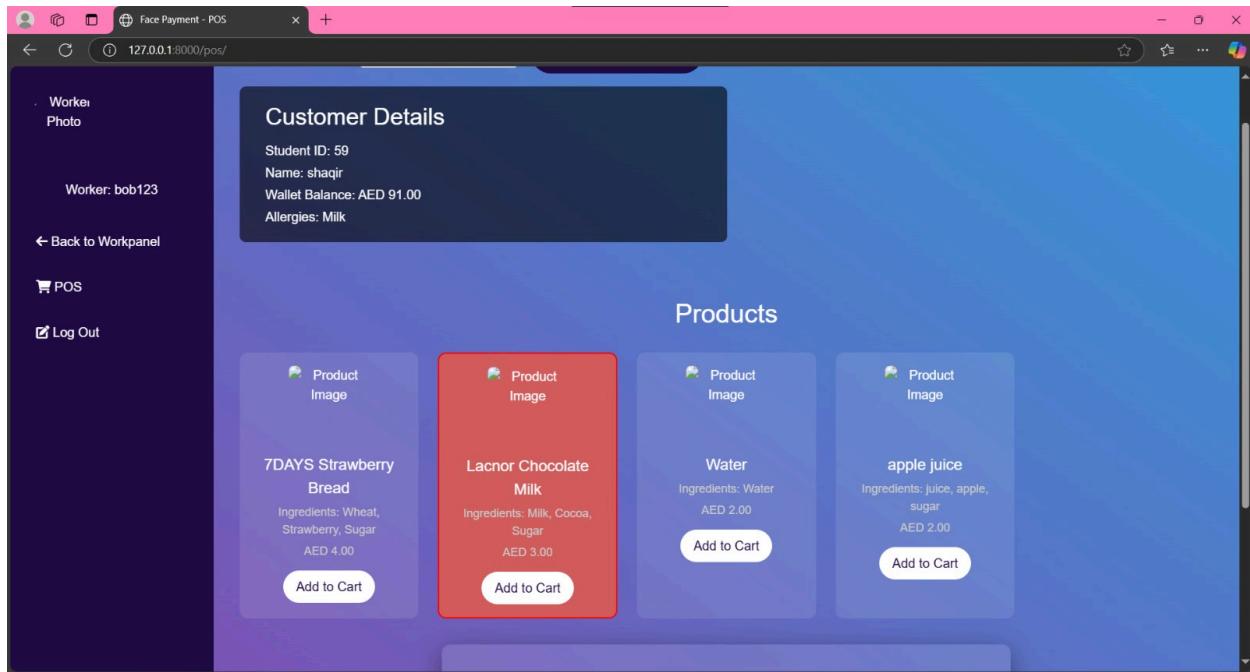
This is what the point of sale would display once a customer's face is scanned showing their relevant information and the options of products for purchase.

Point of Sale Cart



This is what the cart would look like on the point of sale.

If allergy is found:



This is an example of a customer attempting to purchase an item they are allergic to. The unavailable items will be shown in red, displaying that there is an issue regarding the ingredients.

Implementation

1. Database Implementation – PostgreSQL

Initial Stage: The project began with SQLite for simplicity during early development.

Final Database: Migrated to PostgreSQL for its robustness, scalability, support for JSON fields, and better relational data handling.

Key Models Implemented:

User model: with fields like role (student, parent, admin), email, name, and wallet balance.

Student: linked to users and parent accounts.

FaceEmbedding: stored the facial vectors per student.

Allergy and ItemAllergy: allowed allergy tagging for food items.

Item and Cart: tracked POS items and purchases.

Transaction: stored students transactions

Challenges:

Switching from SQLite to PostgreSQL required updating field compatibility.

Migrating data and adjusting Django settings for PostgreSQL connection.

2. Neural Network Integration

Built using TensorFlow and OpenCV.

Trained on custom student face images collected through the interface.

Model output: face embeddings (128D vectors) stored in PostgreSQL.

Optimized for child facial features using data augmentation (e.g., rotation, brightness variation).

3. Face Recognition Functionality

Used `face_recognition` Python library on top of `dlib`.

Captured student faces through a webcam or uploaded photo.

Compared live embeddings to those stored in the database.

Triggered automatic authentication and initiated the POS interface.

4. POS System

Interface built in Django (pos.html, workpanel.html).

Linked recognized student to their profile and wallet.

Allowed food item selection and auto-deducted from balance.

Integrated allergy checks by cross-verifying item ingredients with student allergies.

5. Add Kid Feature

Form for parents to register new students.

Collected: name, DOB, grade, allergies, and face image.

Used Django model forms + AJAX for smoother submission.

Facial image was immediately processed and embedded.

6. Parent Panel & Wallet

Parents could view transactions, set spending limits, and reload balance.

Each transaction recorded with timestamp and item details.

7. Allergy Detection

Allergy model linked to students and items.

Each POS transaction was auto-checked against flagged allergens.

If a match was found, a popup warning was displayed.

8. APIs & Libraries Used

face_recognition (dlib backend) – for facial comparison.

OpenCV – image processing and face capture.

TensorFlow – custom neural network training.

psycopg2 – PostgreSQL connection with Django.

Django REST framework – for future extensibility (parent app, admin control).

Bootstrap and jQuery – for UI enhancements.

TEST PLANS

Test Case ID	Requirement	Test Case Name	Explanation of test	Test Steps	Expected result	Actual	Pass/Fail	Error ID (if fail)
TC001	Face Capturing	Neural Network Camera Test	Ensure that the camera device connects to the neural network	1. Ensure a camera is connected 2. Run the code using a valid compiler	Camera turns on and a window with the camera screen appears	Camera turns on and a window with the camera screen appears	Pass	
TC002	Face Capturing	Neural Network Camera Quit test	The user should be able to quit the window if required	1. Camera window pops up 2. press 'Q'	Window is closed and a message appears	Window is closed and a message appears	Pass	
TC003	Face Capturing	Neural Network Camera Capture Test	Capturing an image using openCV	1. Camera window pops up 2. press 'S'	Window is closed and a prompt to type name appears	Window is closed and a prompt to type name appears	Pass	
TC004	Face Capturing	Name Test	Saving an embedding with a name	1. Take image 2. Enter first and last name	Name and embedding is saved as a record	Name and embedding is saved as a record	Pass	
TC005	Face Capturing	Blank name test	Not entering a name so embedding is not saved	1. Take image 2. Do not enter name, press enter immediately	Not saved in the database as a record	Not saved in the database as a record	Pass	

Test Case ID	Requirement	Test Case Name	Explanation of test	Test Steps	Expected result	Actual	Pass/Fail	Error ID (if fail)
TC006	Face Capturing	Invalid name	The expected input is a first and last name, so only one name will be entered	1.Capture image 2.Enter single name	Record will not be saved due to invalid data	Record is saved	Fail	NN-01
TC007	Face Capturing	Camera Timeout	Check if the camera times out after 3 minutes	1.Camera window pops up 2.Wait 3 minutes	Window will not timeout, but framerate is reduced	Window does not timeout, limited to 30 frames per second	Pass	
TC008	Face Capturing	Comparison with existing record	A person with their face already scanned will attempt to scan their face again	1.Camera window pops up 2.Image is taken 3.Name is entered 4. Successful match is shown	A match will be found and the euclidean distance will be shown for proof	A match is found and the euclidean distance is shown	Pass	
TC009	Face Capturing	Comparison using side view of a face	A person with an existing record will attempt to scan the side of their face	1.Camera window pops up 2.Image is taken 3.Name is entered	A match will not be found as face is obstructed	The best match is shown and value is accepted	Fail	NN-02

Test Case ID	Requirement	Test Case Name	Explanation of test	Test Steps	Expected result	Actual	Pass/Fail	Error ID (if fail)
TC010	Face Capturing	Comparison while wearing a mask	A user with an existing record will attempt to scan their face while wearing a mask	1.Camera window pops up 2.Image is taken 3.Name is entered	The model should give a correct match as it has been trained to do so	The model gives a correct match based on the given information	Pass	
TC011	Face Capturing	Comparison with no face in frame	An image will be taken with no face in frame	1.Camera window pops up 2.Image is taken 3.Name is entered	The model should return no match	The model returns a match with the given name	Fail	NN-03
TC012	Face Capturing	Comparison with 2 faces in frame	An image will be taken with 2 faces in frame	1.Camera window pops up 2.Image is taken 3.Name is entered	The model should take an image of the face which is more towards the center of the frame	The model takes a random image of either person in frame	Fail	NN-04
TC013	Face Capturing	API usage to save to the database	The saved record should be stored on the Django Database	1.Enter name and image 2.Record should be saved	API should work and the record should be saved in a children's table	API does not work	Fail	CS-01

Test Case ID	Requirement	Test Case Name	Explanation of test	Test Steps	Expected result	Actual	Pass/Fail	Error ID (if fail)
TC014	Face Capturing	Blurred face test	Check if the model rejects blurred images	1.Capture image with intentional blur 2.try to save it	If image contains valid features, it should save	Although the features are not sharp, the model saved the blurred image.	Pass	
TC015	Face Capturing	Low lighting test	Check if the model rejects images with bad lighting conditions	1.Capture image with intentional low lighting 2.try to save it	As long as face is still visible, embedding should still be saved	Although the features are not sharp and not too visible the model saved the image.	Pass	
TC016	Face Capturing	Smile test	Check if the model accepts a smiling facial expression	1.Smile during image capturing 2. Enter name	Should still be valid	Is valid	Pass	
TC017	Face Capturing	Open Mouth Test	Check if the model accepts an image with an open mouth	1.Intentionally open mouth during image capture 2.Enter name	Comparison should be valid	Comparison is invalid	Fail	NN-05

Test Case ID	Requirement	Test Case Name	Explanation of test	Test Steps	Expected result	Actual	Pass/Fail	Error ID (if fail)
TC018	Student Management	Add Student	Check to see if adding a new student is possible.	1.Add the student record to the table called students 2. Verify if the record exist	The student's record has been added and is accessible.	The record was successfully inserted and retrieved.	Pass	
TC019	Student Management	Retrieve Student	Check if we can see all the student info	select a student id from student table and then check if every field is returned correctly	All the student info should be retrieved	Student info retrieved	Pass	
TC020	Student Management	Update student info	Verify if student info can be updated	Select random student id and try to update his info	The info for the student should be updated	It has been updated successfully	Pass	
TC021	Student Management	Delete student	Make sure that we can delete any student	Select any id and try to delete it	The info for this student should not be accessible	Student record is deleted and we can't display this info anymore	Pass	

Test Case ID	Requirement	Test Case Name	Explanation of test	Test Steps	Expected result	Actual	Pass/Fail	Error ID (if fail)
TC022	Parent Management	Add Parent	Test to see if adding a new parent is possible.	Add the parent record to the table called Parents and Verify if the record exist	The parent record has been added and is accessible	The record was successfully inserted and retrieved.	Pass	
TC023	Parent Management	Display parent info	Test to see if we can display the parent info	Select any parent from parent table and see if the info is retrieved	It should show all the info that we have for the parent	All the info for the parent retrieved	Pass	
TC024	POS Transactions	Process payment	ensure that a transaction is recorded. when a student makes a purchase	1. Add the transaction record to the table called POS_Transactions. And make sure payment modified	The balance is updated in accordance with the recorded transaction	The record was successfully inserted and retrieved.	Pass	

Test Case ID	Requirement	Test Case Name	Explanation of test	Test Steps	Expected result	Actual	Pass/Fail	Error ID (if fail)
TC025	POS Transactions	Retrieve Transaction	Check if transaction records can be retrieved.	Insert a transaction and then we retrieve the transaction to confirm storage	The transactions should be correctly retrieved	Transaction is retrieved successfully	Pass	
TC026	POS Transactions	Ensure Price Calculation	Check to make sure the system calculates the total cost	Add multiple items and then we have to verify the calculation for the total price	The total price should equal to the sum of the items	The price calculation is made correctly	Pass	
TC027	Wallet Top-ups	Parent Top-up	Test that the parent can top-up their children account	insert top-up record into Wallet_Topups table. then we Check if student's balance is updated.	The student balance should increase	It got increased successfully	Pass	

Test Case ID	Requirement	Test Case Name	Explanation of test	Test Steps	Expected result	Actual	Pass/Fail	Error ID (if fail)
TC028	Wallet Top-ups	Retrieve Top-up	Test if we can see all the records for the top-ups	After inserting the top-up we see if we can retrieve it	The record for the top up should be retrieved	It shows the top-up data	Pass	
TC029	Allergy Alerts	Add allergy	Ensure that allergy information can be saved and accessed.	Add allergy details to the allergies table, retrieve and confirm the saved data	Allergy details are accurately stored and retrieved.	Allergy information obtained successfully.	Pass	
TC030	Allergy Alerts	Delete allergy	Ensure that allergy records are removed.	Remove an allergy entry and attempt to access the eliminated allergy.	Removed allergy must remain non-retrievable	Allergy record removed successfully.	Pass	
TC031	System Performance	Handle Large Transactions	Make sure many transactions can be handled by the database.	Add 1000 transaction records. Make sure that every transaction accurately stored.	Every transaction must be recorded accurately.	The system managed large transactions successfully.	Pass	

Test Case ID	Requirement	Test Case Name	Explanation of test	Test Steps	Expected result	Actual	Pass/Fail	Error ID (if fail)
TC032	System Logging	Ensure Transaction Log Integrity	Make sure that every transaction is accurately recorded.	Complete multiple transactions. Check the log entries	Every transaction must be recorded with accurate timestamps.	All transaction records confirmed successfully.	Pass	
TC033	Security & Access Control	Unauthorized Data Access	Make sure that student records are not accessible by unauthorized users.	Attempt to use a non-admin account to query student data. Verify whether access is restricted.	Records cannot be retrieved by unauthorized users.	Unauthorized entry was successfully prevented.	Pass	
TC034	Security & Access Control	Prevent Unauthorized Top-up	Make sure that parents can only add money to their child's account.	Attempt to add money to a different student's account. Check if its blocked by the system.	Only their own child are allowed to have top-up.	Unauthorized top-up was blocked.	Pass	
TC035	Backup & Recovery	Ensure Automatic Backup	Make sure that scheduled automatic backups are generate.	Review scheduled backup records and confirm the availability of recent backup.	The latest backup must be available in the system.	Backup schedule confirmed successfully.	Pass	

Test Case ID	Requirement	Test Case Name	Explanation of test	Test Steps	Expected result	Actual	Pass/Fail	Error ID (if fail)
TC036	Backup & Recovery	Test Data Restoration	Make sure that the database is restorable from its backup.	Recover the database from the backup and verify that all data is available.	Data needs to be restored accurately.	Backup successfully restored with intact data integrity.	Pass	

Test Case ID	Requirement	Test Case Name	Explanation of test	Test Steps	Expected result	Actual	Pass/Fail	Error ID (if fail)
TC037	Data Base	Comparison	Ensure Database connect successful	Match the correct username and password	Shows the most correct match given the threshold	Database connect fail	Fail	DB-01
TC038	Database	History Payment	Ensure Database connect successful	Button onclick scroll user to history table	Display the History table to the user.	When button onclick it do show the result	Pass	
TC039	Code	Contact Information	Ensure code working successful	Button onclick show user contact information	Display the contact information to the user.	When button onclick it do show the result	pass	
TC040	Code	Multiple information	Ensure code working successful	Button onclick show user more information	Show other slide information when button onclick	When button on click it do show the next slide information	Pass	

Test Case ID	Requirement	Test Case Name	Explanation of test	Test Steps	Expected result	Actual	Pass/Fail	Error ID (if fail)
TC041	Code	Social media icon	Ensure jump to the correct link	Click icon jump to the link page	Show it will link user to the correct page when button on click	It work successful when the button onclick bring user to the correct link	Pass	
TC042	Code	Back	Ensure take user back to the previous	Click on Back button	Bring user back to the main page if user want to read more before login	It work successful when the button onclick bring user to the Home page	Pass	
TC043	Code	News message	Ensure user can get news	System send message to the user allowed read news happen	User can read message such as if their wallet is empty	The news table area was added,but the message was not test.	Fail	DB-02
TC044	Code	Student profile table	User can see the student profile and check the mentions details	System display the child information to parents and workers	User can click the profile button and check the child information	Work successful when button onclick	Pass	

Test Case ID	Requirement	Test Case Name	Explanation of test	Test Steps	Expected result	Actual	Pass/Fail	Error ID (if fail)
TC045	Login Test	Valid credential login	Checking if signing in with valid credentials works	1.Go to login page 2.Enter valid credentials 3.Press login button	User logs in and is redirected to main page	User logs in and is redirected to main page	Pass	
TC046	Login Test	Valid credential login as admin/POS	Checking if admin login works and redirects to correct page	1.Go to login page 2.Enter valid credentials 3.Press login button	User logs in and is redirected to admin page	User logs in and is redirected to admin page	Pass	
TC047	Login Test	Invalid credential login test	Make sure that an error message comes up when invalid credentials are used	1.Go to login page 2.Enter invalid credentials 3.Press login button	User should be given an error message and not be logged in	User is given an error message and is not logged in	Pass	
TC048	Login Test	Check for form validation	Make sure that the values are entered and meet the form requirements	1.Go to login page 2. Enter information that does not meet form requirements	User should be given an error when typing in the form	User is given an error when typing in the form	Pass	

Test Case ID	Requirement	Test Case Name	Explanation of test	Test Steps	Expected result	Actual	Pass/Fail	Error ID (if fail)
TC049	Password Management	Prevent Weak Passwords	Test that the user can use weak pass and it will be rejected	Attempt to put a weak pass	The system should prevent this and show error message	Weak password are rejected (error enter another pass).	pass	
TC050	Password Security	Store Passwords as Hashed Values	Test and make sure that pass are saved as hashed value not as plain text	Create a new user account and assign a password. and then verify that the password is saved as a hashed value by checking the database.	Passwords should be stored as hashed values.	Pass is saved as haased not as text	Pass	
TC051	Password test	Forgot password testing	Verify that the forget password button works	1.Press on forget password 2. Enter valid User ID	User gets sent a confirmation email	User does not get sent an email	Fail	CS-02

Test Case ID	Requirement	Test Case Name	Explanation of test	Test Steps	Expected result	Actual	Pass/Fail	Error ID (if fail)
TC052	Logout	Testing logout functionality	Make sure the user can log out	1.Logged in user 2.Press on logout button	User gets logged out	User gets logged out	Pass	
TC053	Logout	Testing logout functionality	Make sure the user can log out	1.Logged in user 2.Press on logout button	User gets logged out	User gets logged out	Pass	
TC054	Password test	Forgot password testing	Verify that the forget password button works	1.Press on forget password 2. Enter valid User ID	User gets sent a confirmation email	User does not get sent an email	Fail	CS-02

Test Case ID	Requirement	Test Case Name	Explanation of test	Test Steps	Expected result	Actual	Pass/Fail	Error ID (if fail)
TC055	API connection	Allergy Detection	Ensure that POS detects student allergies at checkout	1.Face is scanned and match is found 2.Database is searched for allergies 3.Result shown up on screen	Detects allergy and rejects payment	Process accepted	Fail	API-01
TC056	Performance	Handle High API Load	Making sure that the API can manage 1000 simultaneous requests	Simulate 1000 API request, and evaluate response durations	Api should handel high load	API handled high load efficiently.	Pass	
TC057	Performance	Optimize Database Queries	Ensure database queries run within optimal time	We Run complex queries on large datasets. And we cheak the time to execute	Queries should execute under 500ms.	It took more time to execute	Fail	Api-02

TC058	Payment Gateway connection	Checking for active payment gateway	Verifying that the payment gateway is active and working	1.Face is scanned and match is found 2.checks are complete, valid results 3.Proceed to payment	Parent is charged from the virtual wallet for that student	Payment gateway does not have an active connection	Fail	API-02
TC059	Database connection	Verifying that top-ups can be completed	Ensure spending limit cannot be exceeded	1.Face is scanned and match is found 2.checks are complete, valid results 3.Proceed to payment	Detects student exceeding spending limit and rejects payment	Payment gateway does not have an active connection	Fail	API-02

Test Case ID	Non-functional Requirement	Explanation of test	Reasoning
TC060	Security	Verifying that login details are secure and hashed, by comparing the actual login with the stored one	In order to ensure security, values need to be hashed so that they cannot be retrieved or hacked by any outside users.
TC061	Performance	Ensuring that the web page can function under a workload	The web page needs to be able to handle a large number of visitors at the same time, meaning that the rate of delivery of data should not be too high (somewhere between 0.1 to 1 second). This is to make sure that the flow of actions is not interrupted. The metric does not include browser rendering time or any 3rd party API Gateways as those are not in the control of the developers.
TC062	Performance	Verifying that the data retrieval rate from the database is acceptable	The system heavily relies on the retrieval of data from the database to make checks such as allergies and spending limits, meaning that the rate of retrieval needs to be quick as to not slow the process down. The rate seems to be acceptable in the current moment.

Test Case ID	Non-functional Requirement	Explanation of test	Reasoning
TC063	Usability	Verifying that the system is simple to use and easy to understand	The overall simplicity and usability of the system is vital, as the user interface needs to be simple enough to be understood by the users. In this case it is parents who may not have much knowledge regarding web pages, so the pages need to be very clear
TC064	Portability	Verifying if the system can be portable and transported	The portability of the system is not as important in the case of FaceCard as it will be implemented in cafeterias, but it is quite a strong point. The system only requires a reliable internet connection and a functioning camera, meaning that it can be easily transported or used on a portable laptop.
TC065	Compatibility	Verifying if application displays accurately on various platforms	The reasoning behind the use of a web page is compatibility. Web pages can be opened on a variety of different devices, many of which are expected to be smartphones or personal computers/laptops.

Test Case ID	Non-functional Requirement	Explanation of test	Reasoning
TC066	Scalability	Verifying the system can be scaled	The system may be used on several cafeterias in the same school, or implemented in branches of the same school so it needs to be able to store the information and remain in sync. Django allows for easy scalability so that will not prove to be an issue

Work Distribution Sheet

Student Name: Ammara Nizardeen

Student ID: 8108328

Features worked on:

Back end

Integration

Implementation

Functionality

Basic security

Documentation

Student Name: Yahia Khalil

Student ID: 7309314

Features worked on:

Neural Network

Test Cases

Documentation

Student Name: Noor Askari

Student ID: 7395656

Features worked on:

Front end

Database

Test Cases

Documentation

Student Name: Joseph John

Student ID: 7422878

Features worked on:

Django structure

Documentation

Student Name: Jingbo Ma

Student ID: 7329672

Features worked on:

Django structure

Documentation

Helped with database integration