SOFTWARE REQUIREMENTS SPECIFICATION

Amazon Go

Estel Haliloğlu - 2247955
Doruk Gerçel - 2310027

Contents

List of Figures

List of Tables

Revision History

Name	Date	Version
SRS Phase I	04.03.2020	1.0
SRS Phase II	08.04.2020	1.1

1 Introduction

1.1 Purpose of the System

Introduced by Amazon itself, Amazon GO is the most efficient solution to make shopping in a supermarket faster than ever. It makes shopping faster as some of its systems are automated. To enter the market the user needs to download its application to their smartphone and as they continue shopping the items they pick are added to their personal cart and when they leave the shop, the total price is automatically taken from the wallet of the user. Therefore, the shopping time in the store is effectively reduced as there is no need to wait in line.

1.2 Project Scope

The scope of this project is to make shopping much faster for the users, by the use of automated systems in the supermarket and the application which is downloaded on the user's smartphone. There is no specified potential user group as everyone who uses a smartphone can use this service. System will have a user interface as an application, which will be downloaded to smartphones and the user will directly use this service from this app. On the app:

- The user creates a personal user profile.
- The user shopping profile is saved.
- The money transition between Amazon Go and the user is done directly via the application.

Also, besides this application there is an automated system in the supermarket. In the supermarket there is a main running program which uses:

- Sensor fusion
- Computer vision
- Deep learning technologies

By the use of these technologies the main Amazon Go program can detect user activities and then interacts with the user application to make changes on the user cart and to get payment when the user leaves the store.
Therefore, the users don't need to wait in line when they shop thanks to these combined systems.

1.3 System Overview

In this section, information about the factors interacting with one another in the system will be given.

1.3.1 System Perspective

The system consists of several components that it interacts with. It holds its main data (the user records and item stocks) in its own database system. The system interacts with the scanning gates at the entrance of the store, which control the entry of customers to the store and notifies the system. One of the most important component that it interacts with is the Detection System. Detection System consists of infrared cameras (used to identify the customer), the pressure triggered shelves (that identify whether an item is picked up or not) and the main computer system which operates the interactions of these subsystems of the Detection System and analyzes this information by using deep learning algorithms. Moreover, the Customer App is for the customers to interact with the system and the Employee App is used to help employees interact with the system. The Management and Control System is used by the system admin to check and control the system.

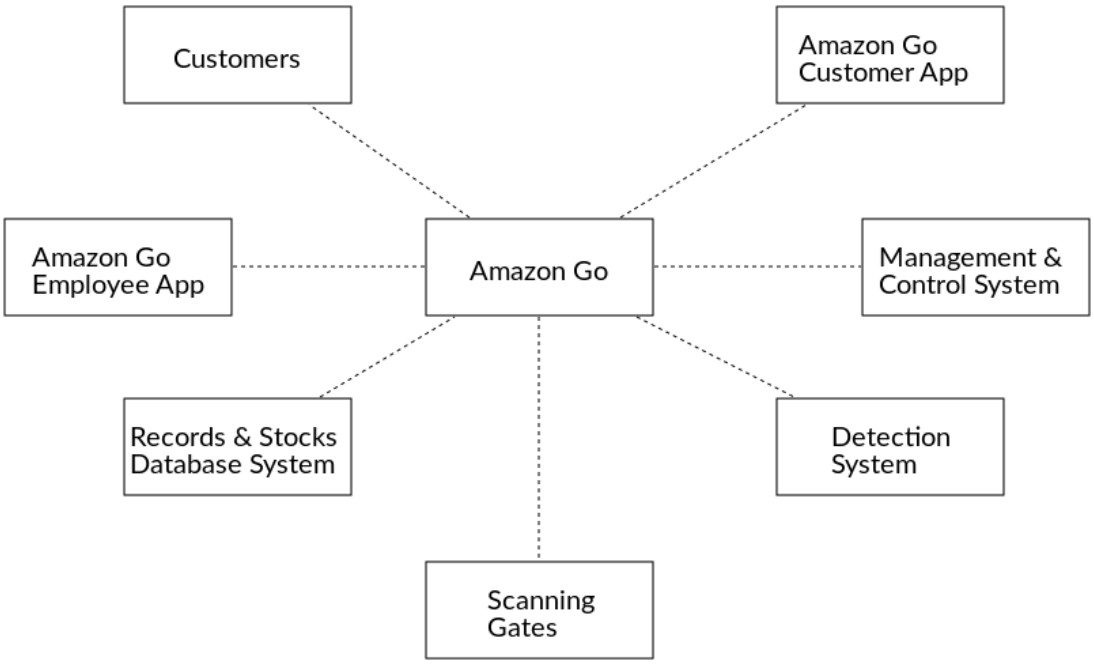

Figure 1.1: Context Diagram

1.3.1.1 System Interfaces

Amazon Go consists of multiple subsystems, therefore it requires several system interfaces to interact and communicate with each other in order to function properly. The main systems in Amazon Go that interact with one another are the main database server, detection systems, scanning gates and Amazon Go Customer App.

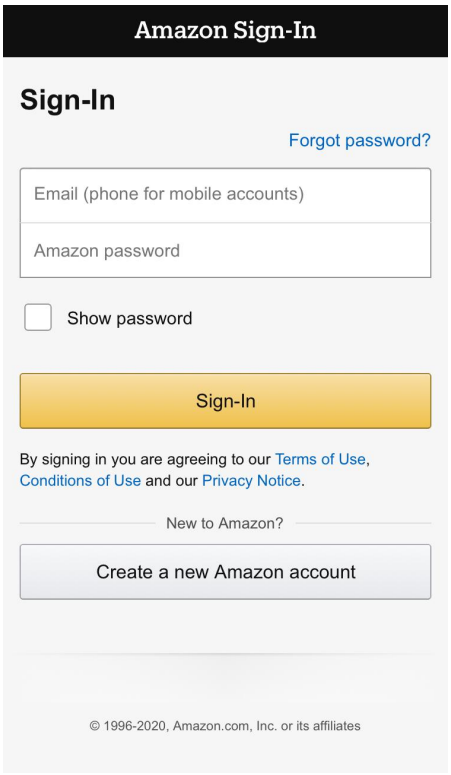

Figure 1.2: Customer app start page for login & registration

Customer Registration API: The main database server is required to store information about customers. The database has its own Customer Registration API. When a customer registers to the system, all the data is directly sent to the Customer Registration API and is stored in the database. This way the user app does not directly access the database system. When registration is approved by the database, the API sends a notification to the customer app to inform of the validation of the account. Operations related to adding or removing cards are connected to this API as well.

Customer Authentication API: The main database server is needed for user ID authentication in several cases (user login/ entering the store/ shop in the market). All the records about the customer are held in the database system, therefore the system must interact with the database in order to authenticate of the customer. The database system has its own Customer Authentication API, so when scanning gates, user detection system, or Customer App tries to get the records of the customer, they send their request to this API and reach the records of the customer.

Stock Condition API: In several cases the employee (via the Employee App) or the system admin (via the Management and Control System) may try to reach the stock condition. There is another API provided by the database system called Stock Condition API. When some system tries to reach the stock condition, it reaches the Stock Condition API and receives the requested data.

Stock Change API: When detection system detects a change in the shelf stock, it must directly decrease or increase the stock information in the system. Due to the design multiple systems might try accessing the stock info at any time. Thus, to prevent overloading, another API is provided by the database to make the changes in the stock

records. (Stock Change API is only used to change the stock records in the database, not to reach the stock info.) When a pressure change in the shelf is detected, the detection system directly sends this data to the database system.

Detection System API: When a customer enters the store, they must be sensed by the detection system's infrared cameras. When the scanning gates read a customer's QR code, they directly send a request to the detection system to match the customer (using their physical and facial map) to the QR code that was read by the scanning gates. So, throughout the shopping process, the detection system can match the customer with their user ID.

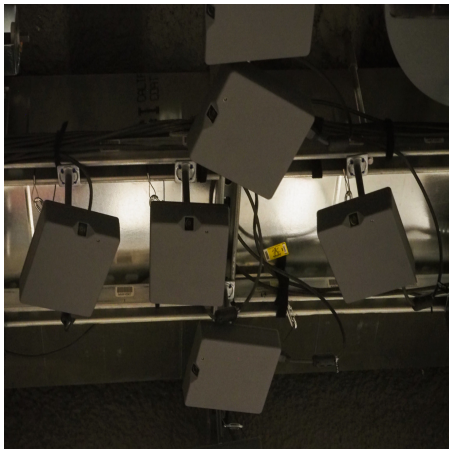

Figure 1.3: Infrared cameras in Amazon Go

Customer Application API: The detection system should always interact with the user app to add the merchandise that the user picks up to their virtual cart. When a user takes a product from the shelf (or puts it back on the shelf), the detection system sends this change info to the Customer App and adds the selected item to the virtual cart (or removes it from the virtual cart).

1.3.1.2 User Interfaces

The user interface of Amazon Go is divided among 3 different type of users: customer, employee and system admin. All of them need different types of interfaces, as their interactions with the system are different from one another. This section will give an outermost view for the interfaces. More detailed information can be found in External Interfaces, Section 3.1.

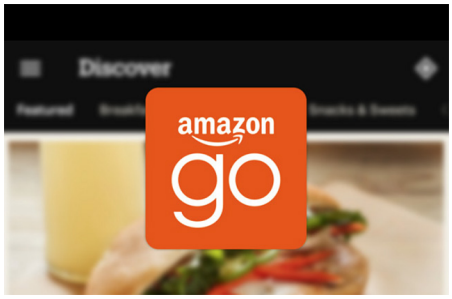

Figure 1.4: Amazon Go app logo

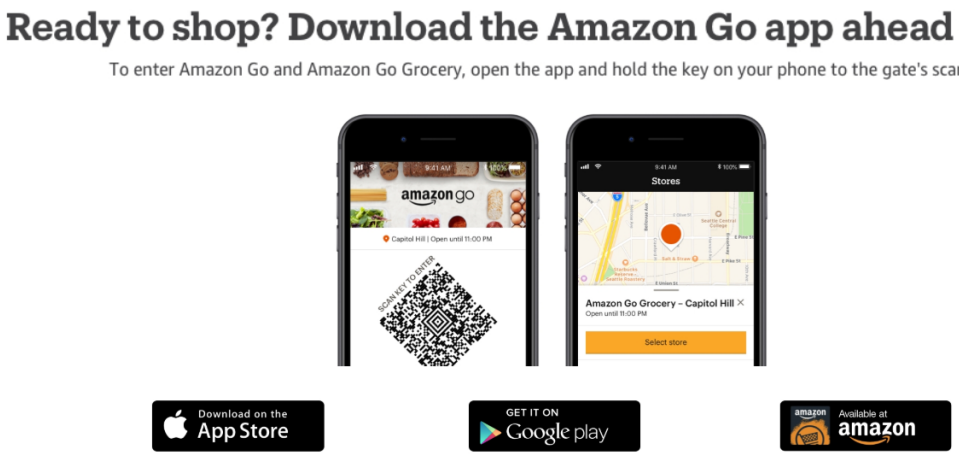

Figure 1.5: Download options for the app on the Amazon website

Amazon Go Customer App: This is the interface that allows users to interact with the system. The user can perform every operation through this interface. All user operations and functions like login, sign-up, manage payment etc. can be performed via this interface. The system will return validation of operation as an output for login and sign-up functions, and it returns the wallet status of the user as an output for the manage payment function.

Figure 1.6: Amazon Go customer app start page

Amazon Go Employee App: This is the interface that allows the employees to interact with the system. The employee can check the shelf stock by using this interface. The system returns the conditions of the stock throughout Amazon Go to this interface. Also, when the stock of an item is below a threshold, the system directly sends a warning signal to the interface to inform the employee. Moreover, the messages can be directly sent to the employee from the admin by this interface. However, an employee cannot send a message to the admin using this interface.

Management and Control System Interface: This is the interface that includes super user (admin) functions. The admin can blacklist cus-

tomers, send messages to employees, and send requests for stock information.
Functions like 'blacklist customers' are super user functions and they directly affect the system itself. The system only outputs an approval message when a customer is blacklisted. Functions like 'send messages to the employees' only return 'sent/not sent' status of the message as an output. Functions like 'send request for stock information' output the detailed status of the stocks.

1.3.1.3 Hardware Interfaces

Any smartphone with Internet connection capabilities and an appropriately updated operating system can be used to shop at Amazon Go.

1.3.1.4 Software Interfaces

Records and Stocks Database System: Amazon Go has its own DBMS. In this system, it holds both user records and stock information of the merchandise. In user records, the system holds the information (account id, name/surname, credit card number, current charge, facial and physical map) of the user. In the stock info, it contains the dot codes of the items, their place in the store (shelf number etc.) and their count.

Operating System: Amazon Go is spread among multiple platforms. Therefore, it uses several different operating systems for these different platforms. The Fire OS 7.3.1.1 (the latest version of Amazon's own operating system) is used in the smartphones that are distributed to the employees, so that they can run their Employee App on their devices. Both Windows and Linux can be used for Management and Control System as it can run on both of these operating systems.

1.3.1.5 Communication Interfaces

Amazon Go requires several connections between its systems that must be encrypted and formed in a secure manner. All Amazon Go Customer App, Amazon Go Employee App, and Management & Control System must be connected to Amazon Go with HTTPS, so that the connections between these systems are safe and secure. (Although it is not as fast as HTTP, for these communications security is the priority.)

1.3.1.6 Memory Constraints

Memory constraint is not a main issue. Amazon Go's database system is sufficient, but it is maintained by Amazon's own company database managers and regularly backed up into Amazon databases in case an unforeseen situation arises. So, there is no need to worry about the memory constraints of Amazon Go.

1.3.1.7 Operations

The operations of Amazon Go can be divided into certain categories:

Customer Operations:

- Sign-Up & Sign-In as User Authentication
- Manage Payment
- Enter the Store
- Exit the Store
- Shop

Employee Operations:

- Check Shelf Stock

Admin Operations:

- Manage Workers
- Blacklist Customers
- Get Stock and User Records

System Operations:

- Detect User
- Add User Shopping Info
- Update Stock Info

Explanations and detailed information about these operations will be provided in Section 1.3.2 and Section 3.2.

1.3.2 System Functions

Function	Summary
Sign-Up	Lets user register to Amazon Go through the Customer App.
Sign-In	Lets user log in to their account.
Manage Payment	Lets user set a default card for payment and add/remove a credit card to/from their account.
Enter Store	The user shows their QR ID to the scanning gates, is recognized by the system and can shop from the store.
Exit Store	User exits the store by passing through the scanning gates and the payment is done automatically.
Shop	The item taken from the shelf is automatically added to the cart of the user.
Check Shelf Stock	The employee checks the stock of the merchandise by using the Employee App.
Manage Workers	The admin can send messages to the employees or rearrange the employee salaries from the Management and Control System.
Blacklist Customers	The admin can blacklist some customers who violate the usage rules of the system from the Management and Control System.
Get Stock and User Records	The admin can reach any record in the database from the Management and Control System.
Detect User	The system must recognize the user as they enter the market and continue following their actions.
Add User Shopping Info	The system automatically saves the user shopping info to the database.
Update Stock Info	The system automatically decreases the stock number of items from the database if it is picked up by any user.

Table 1.1: System Functions

More detailed information about these system functions will be provided in Section 3.2.

1.3.3 User Characteristics

The main user that must have an advanced knowledge to control the system is the system admin. The system admin must be familiar with the system commands. Also, they must be familiar with basic database commands as well. The admin uses the Management and Control System, so this is the reason why he must be able to execute the system control commands.

1.3.4 Limitations

Regulatory policies: The system keeps important personal information (name, ID, credit card number etc.), so none of this data should be published.

Hardware limitations: As most of the outer systems such as the detection system, Management and Control Panel and the applications of both the users and the employees try to reach the database server, it should be fast enough to respond to their requests. Also, other devices like scanning gates and detection system components should send requests frequently and immediately react to changes. The applications do not need many attributes. Therefore, including necessary OS and internet connection should be sufficient for their operations.

Interfaces to other applications: Amazon Go must always be able to connect to the database server, detection system and the scanning gates in order to function.

Parallel operation: Amazon Go is used by multiple users at a time, so the system server must be able to co-operate with multiple processes.

Audit functions: All the employee salary and management data can be reached through Management and Control System by the admin.

Control functions: Database server cannot be reached directly. Instead, there are several APIs that are provided by the database, such that the systems that can reach the database can only interact with the database in a controlled manner.

Higher-order language requirements: The system mostly uses the operating system that are created by Amazon (Fire OS), so C, C++, Java are mostly used. As systems use deep learning algorithms, it uses the advanced libraries of Python. Also, as it uses database queries frequently, it uses SQL as a query language.

Signal handshake protocols: The user data must be conserved in a secure manner. Therefore, the system uses TCP to check the delivery of data. Also, all data transferred through the web uses HTTPs protocol.

Quality requirements: The system data must always be kept safe, as lots of crucial personal information is kept in the database. Also, the system must be backed up frequently in order to deal with system failures.

Criticality of the application: When the user is not in the store the system, failure is not a big issue as the user will not be able to perform their usual operations. However, when the user is in the store, the application always interacts with other systems like the detection system and the scanning gates. Therefore, it is essential for the application to

maintain its connection with other systems, and in case of a system failure, the system admin and the IT staff must try to recover the system immediately.

Safety and security considerations: As most of Amazon Go's systems are directly linked to Amazon systems, the security and safety considerations and measures are mostly taken by the Amazon staff. Nonetheless, there are several cases that concern the Amazon Go staff, like loss of connection between the systems or backup mechanism of the database. Therefore, if any kind of failure occurs, Amazon Go staff must directly try to recover the system. Also, the data that must be kept safe should not be reached directly. For example, the database system is not directly reachable, instead it uses several APIs, so when a worker or a component of a system tries to reach the database, it reaches the API so the system is kept secure.

Physical/mental considerations: There is no certain physical or mental considerations for this system. However, only people who are older than 18 years can use this system as they are not allowed to use credit cards and the system requests credit card and payment information during the registration step.

1.4 Definitions

Term	Definition
API	Application programming interface
HTTPs	Hypertext Transfer Protocol Secure
TCP	Transmission Control Protocol
ID	Unique number to identify a person or an item
Fire OS	Operating System that is developed by the Amazon
DBMS	Database management system

Table 1.2: Definitions

2 References

This document is written with respect to the below document:

IEEE. (2011, December 1). 29148-2011 - ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering.

Other sources:

amazon (December 5, 2016). "Introducing Amazon Go and the world's most advanced shopping technology" – via YouTube.

"Amazon's 1st high-tech grocery store opens to the public". CBC News. January 22, 2018. Retrieved March 25, 2020.

"Amazon opens store with no cashiers, lines or registers". Associated Press. January 22, 2018. Retrieved March 25, 2020.

Grewal, Dhruv; Roggeveen, Anne L.; Nordfält, Jens (March 2017). "The Future of Retailing". Journal of Retailing. The Future of Retailing. 93 (1): 1–6. doi:10.1016/j.jretai.2016.12.008

Johnston, Chris (January 22, 2018). "The supermarket with no checkouts". BBC News. Retrieved March 25, 2020.

3 Specific Requirements

3.1 External Interfaces

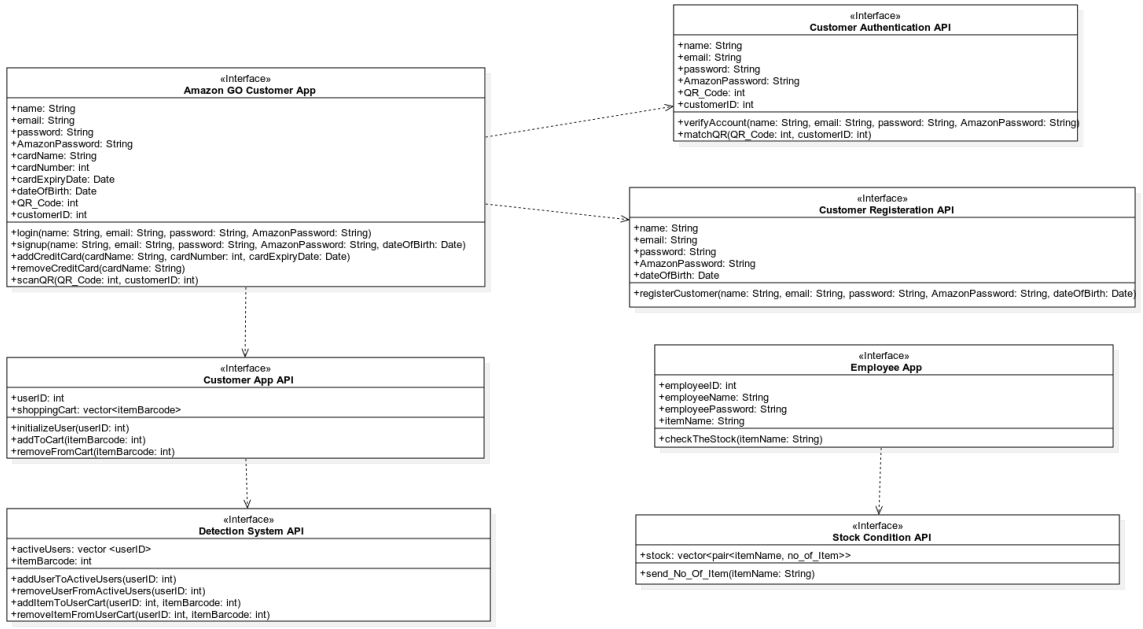

Figure 3.1: External Interfaces Class Diagram

Amazon Go Customer App Interface: This interface allows the customer to interact with the system. It allows the user to perform certain operations. The user can sign up to the system by providing the required information (that are given in the class diagram), and after registering to the system, they can sign in to their account by using their e-mail (or phone number) and user password. Sign-up operation directly interacts with the Customer Registration API and the registration is completed in this API. Login operation interacts with the Customer Authentication API, where the parameters for the login function are validated. When a user scans their QR Code, the Customer Authentication API is used again, both the QR Code and the user ID are sent to the API, and the validation signal is sent back to the Customer App. If the customer entry is validated, the app's API (Customer App API) is activated.

Customer App API: This API's purpose is to communicate with the Detection System. When it is activated by the Customer App, it sends a signal to the Detection System

API (via the initializeUser() function). Also, if it receives messages from the Detection System API, it puts the items that are selected by the customer to the virtual cart.

Detection System API: It is the interface of the detection system that interacts with the other components. It receives messages from the Customer App APIs and when it receives an initialization signal, it adds the user to the active users. This way, the system knows which users to follow. When an active user shops in the store, the items selected or removed are sent to the Customer App API of the active user.

Customer Authentication API: This API is provided by the database system. When a user account is to be verified, this API is used to check the parameters that are passed to the API. Also, when a user scans their QR code, it matches the QR Code with the user ID.

Customer Registration API: This API is provided by the database system. When a user account is to be registered, this API is used to store the information -that are passed by the Customer App- to the database system.

Employee App: This is the interface that allows the employees to interact with the system. The employee can check the stock condition of any item. They give the name of the item as a parameter (to the checkTheStock function) and can check the stock condition from the database (by interacting with Stock Condition API).

Stock Condition API: This API is provided by the database system. It holds the names of the items and number of these items. When it receives a request from the Employee App, it sends the number of selected item.

3.2 Functions

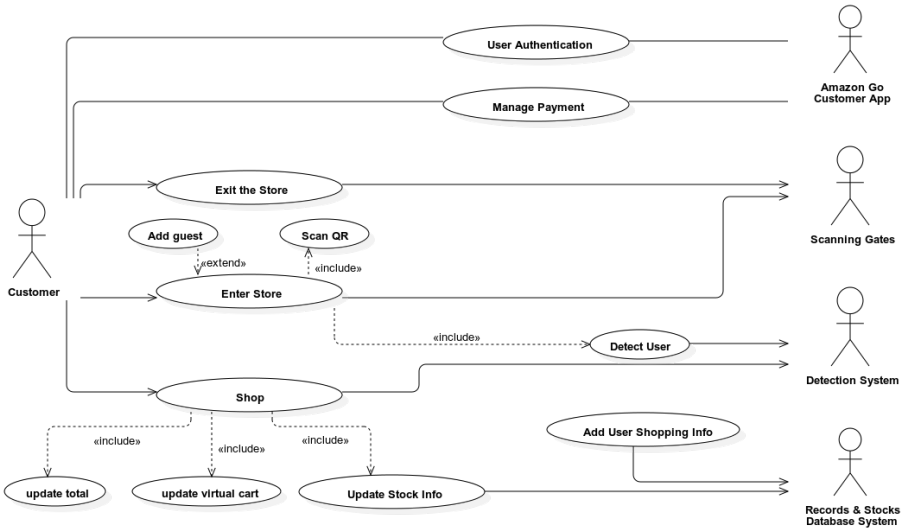

Figure 3.2: Use Case Diagram

Use case name	User Authentication
Actors	Customer, Amazon Go customer app
Description	Customer uses this method to create an account OR to log into an already existing account
Data	Name, e-mail address, password OR E-mail and associated Amazon password
Preconditions	Customer has the Amazon app installed on their mobile device
Stimulus	Clicking "Create your Amazon account" after entering name, e-mail, and password OR Clicking "Sign-In" after entering e-mail and password
Basic flow	I. Customer's name, e-mail, and password are received by the app II. Password is checked against preconditions and approved III. E-mail is authorized to be functional IV. An account is created in the Amazon Go database with name, e-mail and hashed password V. Customer is logged into the newly created account and redirected to the user start page on the app OR I. Customer's e-mail and password are received by the app II. E-mail and hashed password match is checked against user's information in the Amazon Go database III. User information is approved and customer is logged into the system
Alternative flow	User can substitute the e-mail address with their phone number.
Exception flow	If password fails to fulfill any of the following conditions: I. Password must be longer than 6 characters II. Password must contain at least 1 special character III. Password must contain at least 1 number IV. Password must contain at least 1 capital letter, display the Error Message "Please enter a password matching necessary conditions." If e-mail is unauthorized, display the Error Message "Please enter a valid e-mail." OR If the e-mail and password entered do not match, display the error message "E-mail or password incorrect."
Postconditions	Customer has successfully started an account. OR Customer has successfully logged in.

Table 3.1: User Authentication

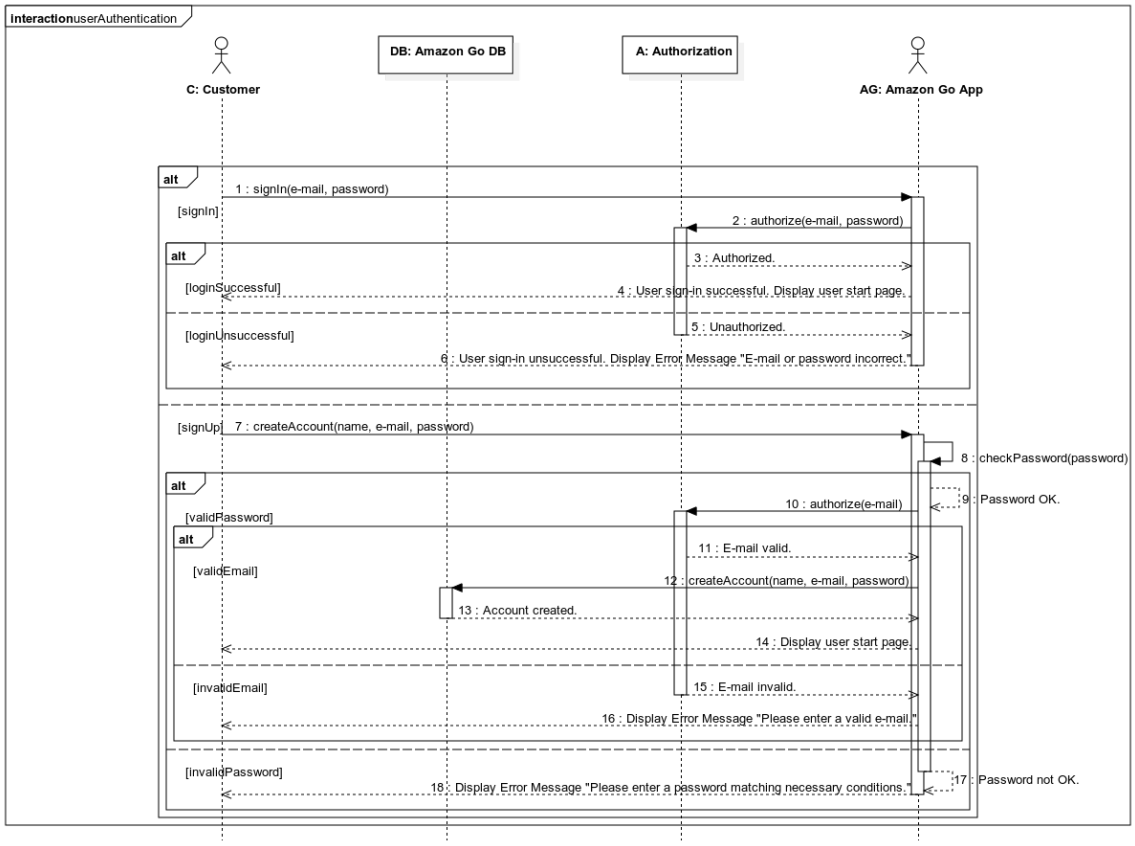

Figure 3.3: Sequence Diagram of the User Authentication Use Case

Use case name	Manage Payment
Actors	Customer, Amazon Go customer app
Description	This function's main purpose is to choose a default card to be used for checkout. It also supports methods to: remove an existing card in payment methods, edit an existing card's information, add additional payment methods
Data	Card to be set to default OR Card to be removed OR Card to be edited and the new name, expiry date, or address to update it with OR Name, card number, and expiry date on a new card to be added
Preconditions	Customer has an existing Amazon account
Stimulus	Clicking "Set as default card" after picking a card from Payment cards on the Amazon app OR Clicking "Remove" under the desired card for removal on the Manage payment cards page OR Clicking "Edit" under the desired card for update on the Manage payment cards page OR Clicking "Add a payment method" on the Manage payment cards page
Basic flow	The preferred card for payment is chosen and set as default on More >Settings >Payment cards after clicking the "Set as default card button" OR The card chosen for removal is removed from the customer's payment methods on More >Settings >Payment cards >Manage payment cards after clicking the "Remove" button underneath OR On More >Settings >Payment cards >Manage payment cards > Edit payment method, an existing card is updated with new name, expiration date or billing address once desired changes are supplied and "Save" button is clicked OR On More >Settings >Payment cards >Manage payment cards > Add a payment method, a new card is added to the customer's payment methods once correct and complete information is entered and "Add you card" button is clicked
Alternative flow	-

Exception flow	[No exception flow for default card setting] OR [No exception flow for card removal] OR If the name entered does not match the card number present, display Error Message "Name entered is incorrect." If the expiry date entered does not match the card number present, display Error Message "Expiration date entered is incorrect." [No exception flow present for billing address change] OR If any of the info fields have inconsistent data, display Error Message "Please check all information entered is correct."
Postconditions	Default card has been chosen OR A card has been removed from the app and disassociated with the user in the Amazon Go database OR An existing card has been updated with new info and the changes were reflected to the database OR A new card has been added and associated with the user in the database

Table 3.2: Manage Payment

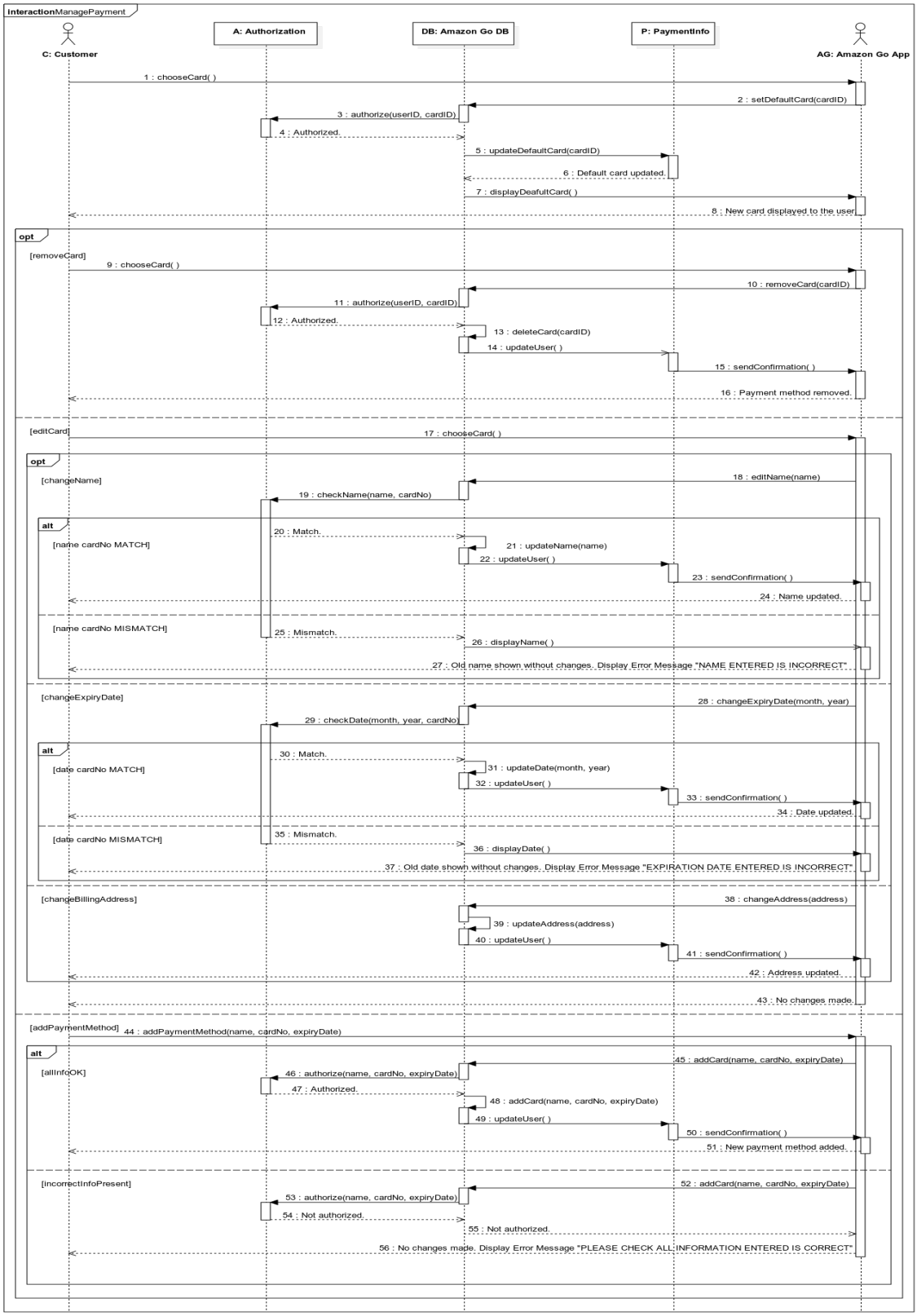

Figure 3.4: Sequence Diagram of the Manage Payment Use Case

Use case name	Enter store
Actors	Customer, scanning gates
Description	The customer enters the store through the scanning gates. This function includes two other use cases, Scan QR and Detect User. It also has an "extend" use case to it, Add Guest
Data	QR code, bodily habitus and facial data
Preconditions	Customer has the app on their phone and their account has previously been set up
Stimulus	Scanning your phone at the scanning gates
Basic flow	Customer goes up to the gates, scans the QR code on their app and the gates open for them allowing entry
Alternative flow	-
Exception flow	-
Postconditions	Customer has entered the store

Table 3.3: Enter Store

Use case name	Add guest
Actors	Customer, scanning gates
Description	The customer can bring in guests alongside by associating them with their Amazon account at the gates. This function is an extension of the Enter Store use case
Data	QR code
Preconditions	Customer has the app on their phone and their account has previously been set up
Stimulus	Additional scans of the QR code after clicking "Add guest" on the app
Basic flow	Customer chooses the "Add guest" functionality on the app and scans their QR code an additional time for each guest they are bringing in. Items picked by the guests are added to the Amazon customer's virtual cart
Alternative flow	-
Exception flow	-
Postconditions	Customer's QR code is recognized by the system and entry of guests is allowed

Table 3.4: Add Guest

Use case name	Scan QR
Actors	Customer, scanning gates
Description	The customer shows their QR code from their app to the scanners on the gates
Data	QR code
Preconditions	Customer has the app on their phone and their account has previously been set up
Stimulus	Showing your phone to the scanning gates
Basic flow	Customer lets the scanners on the gates read the QR code from their app and the gates open for them allowing entry
Alternative flow	-
Exception flow	-
Postconditions	Customer's QR code is recognized by the system and entry is allowed

Table 3.5: Scan QR

Use case name	Detect user
Actors	Customer, detection system
Description	Customer's ID from the scanners is matched with physical characteristics of them from the cameras in the database. This function is included in the Enter Store use case
Data	Bodily habitus and facial data, customer ID from their QR code
Preconditions	Customer has just scanned their QR code to enter the store
Stimulus	Customer ID from the QR scan being received
Basic flow	I. Customer ID from the QR code scan at the gates is received II. Bodily habitus and facial data is recognized III. Customer ID from (I) is combined with the data from (II) and a complete profile is constructed for the customer IV. Customer is tracked during their shopping with this profile
Alternative flow	-
Exception flow	-
Postconditions	Customer is now being tracked by the store's system

Table 3.6: Detect User

Use case name	Shop
Actors	Customer, detection system
Description	Customer's shopping and total is tracked using their profile in the database. This function includes several use cases, Update Total, Update Virtual Cart, and Update Stock Info
Data	Bodily habitus and facial data, customer ID from their QR code, item data
Preconditions	Customer has entered the store
Stimulus	Successfully entering the store
Basic flow	I. Customer ID from the QR code scan at the gates is received II. Bodily habitus and facial data is recognized III. Customer ID from (I) is combined with the data from (II) and a complete profile is constructed for the customer IV. Customer is tracked during their shopping with this profile
Alternative flow	-
Exception flow	-
Postconditions	Customer has completed their shopping

Table 3.7: Shop

Use case name	Update total
Actors	Customer, detection system
Description	Customer's cart total is continuously updated in the database throughout their shopping. This function is included in the Shop use case
Data	Bodily habitus and facial data, price of the latest item that was added to/removed from the virtual cart (VC)
Preconditions	Customer has successfully entered the store and started picking items
Stimulus	The cameras recognizing the customer and the item they are picking up/putting down
Basic flow	I. Adding items to VC: Customer and item is recognized. Price of the item is added to the customer's total in the database II. Removing items from VC: Customer and item is recognized. Price of the item is removed from the customer's total in the database
Alternative flow	-
Exception flow	-
Postconditions	Customer's cart total in the database now reflects the latest changes

Table 3.8: Update Total

Use case name	Update virtual cart
Actors	Customer, detection system
Description	Customer's virtual cart (VC) is continuously updated in the database throughout their shopping. This function is included in the Shop use case
Data	Bodily habitus and facial data, item specifics such as dot codes recognized by the cameras and weight recognized by the load cells on the shelves
Preconditions	Customer has successfully entered the store and started picking items
Stimulus	The cameras recognizing the customer and the item they are picking up/putting down
Basic flow	I. Adding items to VC: Customer and item is recognized. Item ID is added to the VC in the database II. Removing items from VC: Customer and item is recognized. Item ID is removed from the VC in the database
Alternative flow	-
Exception flow	-
Postconditions	Customer's virtual cart in the database now reflects the latest changes

Table 3.9: Update Virtual Cart

Use case name	Update stock info
Actors	Records and Stocks Database System
Description	The stocks of the store changes constantly. (When a customer takes something from the shelf or puts something back on the shelf, the stock changes.) Therefore the store's stock database must be updated continuously in accordance with the changes on the shelf. This function is included in the Shop use case
Data	The dot code of the item, the type of the item, the boolean indicating whether the items are put on the shelf or taken from the shelf, the number of items
Preconditions	-
Stimulus	Change on the shelf
Basic flow	I. The pressure change on the shelf notifies the detection system II. The detection system catches the item, type of the item and the number of this item. III. Amazon Go sends that data to the Records and Stocks Database System.
Alternative flow	-
Exception flow	If an undefined type of data is detected, it is written to the Error log file.
Postconditions	Records and Stocks Database System makes the changes according to the data received.

Table 3.10: Update Stock Info

Use case name	Exit store
Actors	Customer, scanning gates
Description	The customer leaves the store passing through the scanning gates once more
Data	Bodily habitus and facial data that was recorded by the Detect User function
Preconditions	Customer (and guests) has successfully entered the store and completed their shopping
Stimulus	The cameras recognizing the customer at the scanning gates using the aforementioned data
Basic flow	I. Customer's location at the scanning gates and their direction as exit is detected by the cameras II. Their cart total is deducted from their account upon exit III. A bill is generated IV. The e-receipt is sent to their app to be shown in the receipts history
Alternative flow	-
Exception flow	-
Postconditions	Customer (and guests) has left the store. Virtual cart total has been deducted from their account. Customer can now see their most recent receipt on their app

Table 3.11: Exit Store

Use case name	Add user shopping info
Actors	Records and Stocks Database System
Description	To create a user preferences profile (which customer buys which items), customers' shopping data is collected in the Records and Stocks Database System.
Data	The user ID, the dot codes of the items
Preconditions	-
Stimulus	Customer buys the items by exiting the store
Basic flow	I. Amazon Go confirms payment from the user. II. Amazon Go sends both the user ID and the dot codes of the items to the Records and Stocks Database System.
Alternative flow	-
Exception flow	If an undefined type of data is detected, it is written to the Error log file.
Postconditions	Records and Stocks Database System adds the items that are bought to the user profile.

Table 3.12: Add User Shopping Info

3.3 Usability Requirements

- Amazon Go app should be accessible whenever there is internet connection.
- Navigation in the app should be straightforward that anyone who uses smartphones in their daily life should be able to use the app with minimum difficulty.
- Upon app installation and launch, a text and video tutorial in using the system should be presented to the user and it should last less than 1 minute to get through.
- The user should be able pose their question via e-mail in case their problem is not addressed in the F.A.Q. An integrated e-mail utility to contact customer services is present in the app so that the user does not have to switch between apps.
- The user should be able to ask their question via phone calls as well. Once they enter their phone number, a representative should call them so that they will not have to hold the line waiting for someone to become available.
- Stores tab should have a search area so that the user can access store locations in their desired location easily.

3.4 Performance Requirements

- The mobile device should have a minimum of 24.1 MB of free space to install the app.
- The Amazon Go app can be installed on devices with iOS 10.0 and later or Android 4.0 and up.
- The app should launch and show the sign-up/sign-in page in less than 3 seconds.
- Once logged in, launching the app and receiving personal QR code should take less than 800 ms.
- The number of customers in the store should not exceed the number stated by the federal fire codes.
- So long as the above condition is met, each user can scan up to 9 guests with them. The number of people associated with an account at a given time in the store is at most 10 (including the Amazon Go account holder).
- Bill generation should at most take 5 seconds following the exit from the store.

3.5 Logical Database Requirements

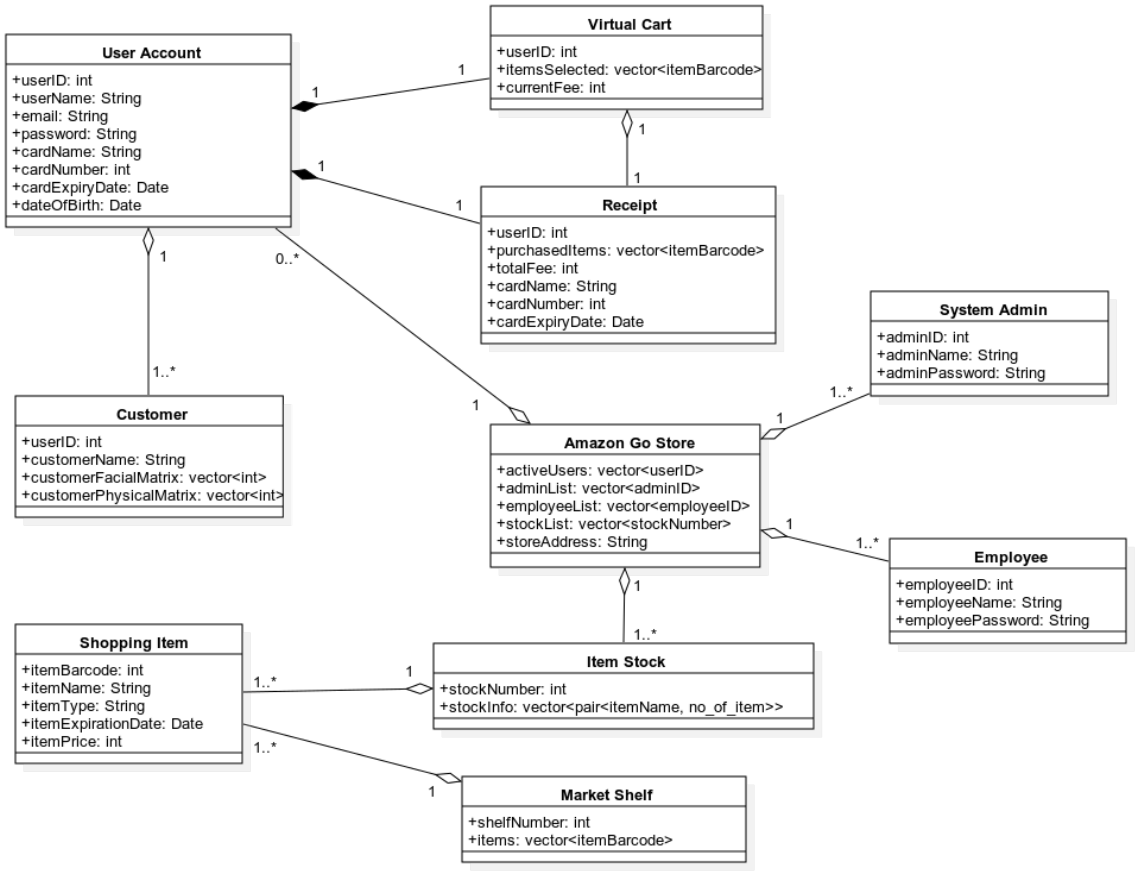

Figure 3.5: Logical Database Diagram

- Most of the operations are related to one Amazon Go Store. Therefore, most of the entities are connected to the store itself.
- Amazon Go Store table should keep the active user accounts (accounts of the users that are in the store), the list of the employees that work in the store, the list of the system admins that are responsible for the administration of the systems in the store, the list of stocks that the store provides items from and the address information of the store.
- System Admin table keeps the admin ID, name of the admin and the password of the admin.
- System admins have permission to access all the tables. They can add new employee entries to the Employee table or remove them. They can reach the Item

Stock table in order to get stock information. They can also reach the User Account table to blacklist customers.

- System Admin table has a many-to-one relationship with the Amazon Go Store table. Multiple system admins can be responsible for a single store, but every admin is responsible for only one store. Also, every store must have at least one system admin.
- Employee table keeps the employee ID, name of the employee and the password of the employee.
- Employees have permission to reach the Item Stock table. They can check the stock of the selected item.
- Employee table has a many-to-one relationship with the Amazon Go Store table. Multiple employees work in a single store and every employee can be assigned to a single store at a time. Also, every store must have at least one employee.
- Customer table is about the customer that shops in the store. Customer table holds the user ID of the customer, name of the customer, facial matrix information, and physical matrix information of the customer. The facial matrix and physical matrix information are kept for the detection system. (It allows detection system to monitor the customer in the store.)
- User Account table is about the Amazon Go user account that has been created. It keeps the required registration (name of user, password of user, etc.) and payment information.
- Virtual Cart table is about the items that are selected by the customer during their shopping. It holds the user ID of the user that it is connected to, list of items that are selected by the customer and the current total.
- Receipt table is about the items that are bought by the customer in the shopping. It holds the user ID of the user that it is connected to, the list of purchased items, total of the purchased items and the payment information.
- The Virtual Cart table and the Receipt table look similar to each other but they have some differences. In Virtual Cart table, the selected items are very dynamic as the customer can select and remove any item during the shopping. On the other hand, the Receipt table contains the items that are purchased, so as their shopping is done, the list does not change. Additionally, Receipt table contains payment information as well, as the payment management is related to this table.
- Customer table has a many-to-one relationship with the User Account table. Several customers can use the same user account, as users have a privilege to shop with the people that they come with. This way, multiple customers use the same user account while shopping. Also, every user account must belong to at least one customer (real person).

- Virtual Cart table has a one-to-one relationship with the User Account table. Every user has only one virtual cart during their shopping, and every virtual cart is assigned to only one user. Virtual Cart is a weak entity of the User Account, as it can not exist without a user.
- Receipt table has a one-to-one relationship with the User Account table. Every user has only one receipt after they finish their shopping, and every receipt belongs to only one user. Receipt is a weak entity of the User Account, as it can't exist without an user account.
- Receipt table has a one-to-one relationship with the Virtual Cart table. Every virtual cart has a receipt connected to it which is formed after the customer finishes shopping.
- User Account table has a many-to-one relationship with the Amazon Go Store table. In a store, there are multiple users at a time, so there are multiple connections of the user accounts to a single store. The store can be empty at a given time, so no user may be connected to the store as well.
- Shopping Item table should keep information about an item such as the item's dot code, name, type, expiration date and price.
- Market Shelf table is about the shelves in the store which store the items. The table should keep the shelf number and the list of items in the shelf.
- Item Stock table is about the stocks of the store that contain the shopping items. Therefore, the table should keep the stock number and the list of the names of the items and the number of these items that are in the stock.
- Shopping Item table has a many-to-one relationship with the Market Shelf table. This is because multiple items are available in a single shelf. Also, a shelf must contain at least one item in-order to be active for use.
- Shopping Item table has a many-to-one relationship with the Item Stock table. There can be multiple items that are available in a single stock. Also, a stock must contain at least one item in-order to be active for use.
- Item Stock table has a many-to-one relationship with the Amazon Go Store table. A store uses multiple stocks for its merchandise, but a stock serves to only one store at a time. Also, the store uses at least one item stock for its merchandise.

3.6 Design Constraints

- Data collected by the Amazon Go system about customers is stored on the Amazon Go database. The type of information collected by Amazon can be found here.

- Amazon does not sell customers' information to others. Personal information is only shared as stated in the Amazon Privacy Notice and with subsidiaries of Amazon.com, Inc., which are subject to this privacy notice or follow practices at least as protective as those from this one.
- In certain cases, Amazon may use customers' information to comply with legal obligations.
- During transmission, personal information is subject to encryption protocols and software.
- During the handling of credit card data, Payment Card Industry Data Security Standard (PCI DSS) is followed.
- The customer can choose not to share certain information with Amazon. They would then be ineligible for some of the Amazon Services.
- Customers under the age of 18 may use Amazon Services only with the involvement of a parent or guardian.

3.7 Software System Attributes

3.7.1 Reliability

The data in the Amazon Go Database should be backed up in the Amazon Database. Backups should take place twice a day.
The ROCOF (rate of occurrence of failures) should be 2 failures a day i.e. the MTTF (mean time to failure) should be 12 hours.
Staff should run checks on the system every Saturday to ensure no failures have arisen and if they have, they should take necessary measures to fix them.

3.7.2 Availability

Allowed slots for downtime of the Amazon Go system is from 8 p.m. to 7 a.m. on weekdays and all day long on the weekends. Downtime should not exceed 30 minutes per failure and 60 minutes total in a day. Thus, AVAIL (availability) should be 0.042.

3.7.3 Privacy and Security

Customers' names, e-mails, passwords, address and payment information, and any other data collected via their shopping are not sold to other parties by Amazon.
Such information is shared only with subsidiaries of Amazon.com, Inc. which operate under the same privacy notice as Amazon Go.
Sensitive information stored on the Amazon Go database is accessible by the customer themselves and by system admins strictly in legal cases.

3.7.4 Maintainability

As Amazon Go is clearly a long term project, documentation should be detailed and carefully done to ensure no problems arise as new people are introduced into the development process.

3.7.5 Portability

The user needs to have the Amazon Go app installed on their mobile device to shop at the stores.
The system is only available as an application for smartphones. Non-mobile options do not exist.

3.8 Supporting information

Amazon Go is a chain of cashierless convenience stores in the United States. The aim of the system is to avoid the bottleneck at the checkout due to long lines using the Just Walk Out Technology.
Amazon Go is a promising project to increase revenue as the currently one-of-a-kind detection system in the convenience stores is sure to reduce shrinkage and the free space due to the lack of checkout counters and lines makes room for additional merchandise displays.

4 Verification

5 Appendices

5.1 Assumptions and dependencies

5.2 Acronyms and abbreviations