SOFTWARE REQUIREMENTS SPECIFICATION

AMAZON GO

Zeynep Özalp 2237691
Hilmi Cihan Yıldırım 2237949

CHANGE HISTORY

VERSION	DATE
1.0	01.03.2020
1.1	11.03.2020

Table of Contents

List of Figures

List of Tables

1. Introduction

This document is a Systems Requirements Specification for Amazon Go system, powered by Amazon.

1.1 Purpose

The purpose of this project is to make the customers' experience of shopping efficient. With the world's most advanced shopping technology, Just Walk Out technology, customers will be able to shop with no lines and no checkout. In order to accomplish this, customers simply use the Amazon Go app, scan the QR Code unique to their account, shop and leave. The Amazon Go system is developed so that the experience of the customers are smooth and enjoyable.

1.2 Scope

- System will have a user interface where the users can login to their existing accounts or create a new account. After login, users will specify their personal information and credit cards. Using the user interface, users will be able to display their QR Code, discover meals, see their receipts, search for the Amazon Go stores near by and get help/contact/sign out.
- System will have an interface for the IT staff. IT staff will be able to view system logs. The IT staff will display the system reports, errors and will be notified in terms of emergency.
- System will keep the user information, unique QR Code, card information, receipt address and past receipts for each user via database interface. After scanning the QR Code, system let the user and its dependants pass the gate of the store.
- System will keep track of the user in the store and the products s/he takes via Sensor Fusion. When the user leaves the store, system will charge the account and send the receipt via email.
- The system will not provide online shopping.

1.3 System Overview

This section defines the system's interaction with other products.

1.3.1 System Perspective

The Amazon Go system uses Gate system to scan the user's key and open the turnstile by giving a permission. Sensor Fusion sytem determines whether a user or its dependants is taking a product inside the store and add the product to the user's virtual cart in the application. AI system is used to analyze users' data, optimize the profit by giving recommendations on the app. The Amazon Go application has different interfaces and is used mostly by the users. They can sign up, see their virtual cart, personal information and recieve their receipts after shopping. Team of associates is the working staff in the store with responsibilities of helping the customers and restocking the shelves. IT Staff helps the system function without any errors.

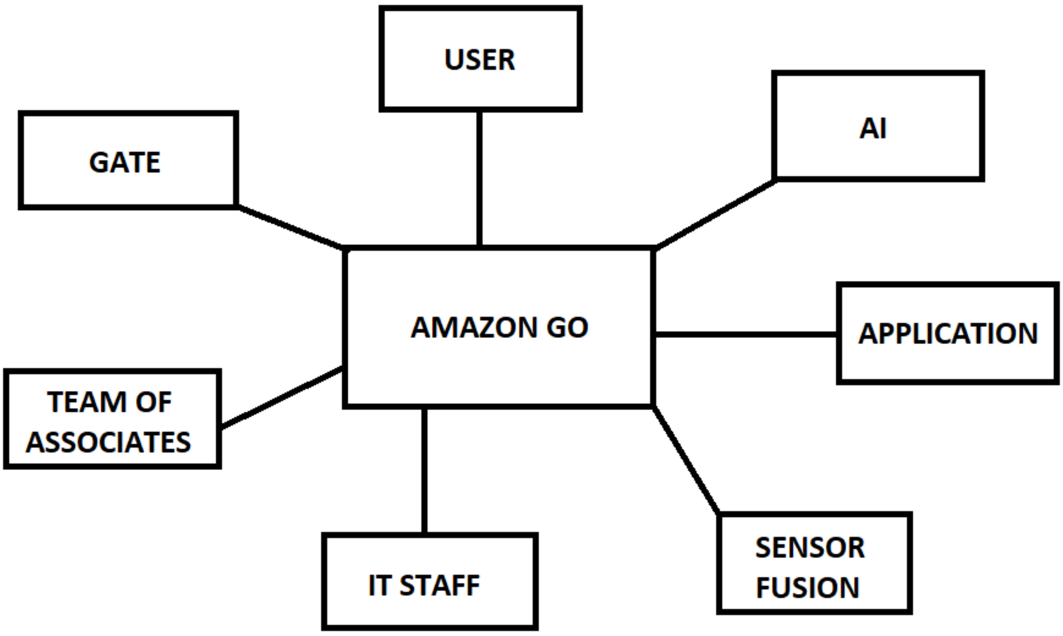


Figure 1: Context Diagram

1.3.1.1 System Interfaces

- **Amazon Web Services (AWS) API:**
- **Database Management System (DBMS) API:** Amazon Go system requires DBMS API to keep track of many information such as users' names, receipt addresses, payment card information etc. Products that users buy also stored to send to the AI system and to improve income.

1.3.1.2 User Interfaces

- **Registration Interface:** This interface is used for a non-user to register as a user. Non-users are required to write their full names, email addresses and new passwords to create an account. Also, they can read the Amazon's Conditions of Use and Privacy Notice. By clicking the "Crate your Amazon account" button with the required information, non-users are consider to accept these terms.
- **Payment Card Interface:** After the registration, users have to enter a payment card in order to continue to the application. Users have to enter their names on the card, card number and its expiretion date so that they chose that credit card as a payment card. Users can also manage their payment cards, delete them or add new ones with this interface at any time they want.
- **User Interface:** Users use this interface to interact with our system. After registration and selection of a payment card, users are directed to the user interface where they can display their keys at the gate, discover meals, see their receipts, find stores nearby and more. From the "More" section, users can contact the IT staff for related issues.
- **Team of Associates Interface:** Team of Asscoiates gets nofitified when a product lacks from the shelves or there is problem in the store. Also, they can see the current status of the stocks, variety of products, products that will arrive to the store etc. They are expected to use this interface to manage the store effectively.

- **IT Staff Interface:** IT Staff will be notified when there is an error with system via this interface. They also be able to examine the related log files to solve these errors and prevent them to happen again. Moreover, they will be able to filter and sort logs to answer the failures quickly.

1.3.1.3 Hardware Interfaces

Any hardware with an opearting sytem (OS) that can run the Amazon Go system is enough. For Amazon Go application, it can work on mobile devices and tablets that can have IOS or Android.

1.3.1.4 Software Interfaces

- **DBMS:** Amazon Go uses database to store all kind of data. Cameras, weight sensors, user information, product information, workers information etc.
- **OS:** An operating system is required for communication between the software and the hardware.

1.3.1.5 Communications Interfaces

Amazon Go system requires an internet connection to handle the operations. Application uses Wi-Fİ Direct for user device to communicate with the system.

1.3.1.6 Memory Constraints

User device need to have enough memory to install the application. 30 MB is enough. System need to have enough memory to sustain AI operations and database operations.

1.3.1.7 Operations

The operations that Amazon Go system offers are the followings:

User Operations:

- Take/drop item
- Get information
- Ask questions

Team of Associates Operations:

- Restock shelves

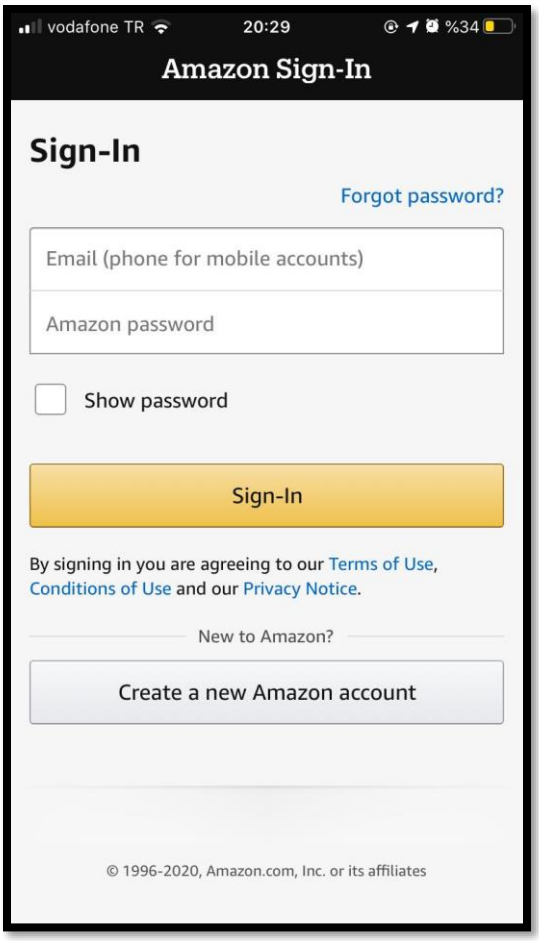
IT Staff Operations:

- View system logs

System Operations:

- Send receipt
- Scan QR Code
- Open turnstile
- SKU optimization
- Product location optimization

These operations are explained in 1.3.2 and 3.2.



Amazon Sign-In

Sign-In

Forgot password?

Email (phone for mobile accounts)

Amazon password

□ Show password

Sign-In

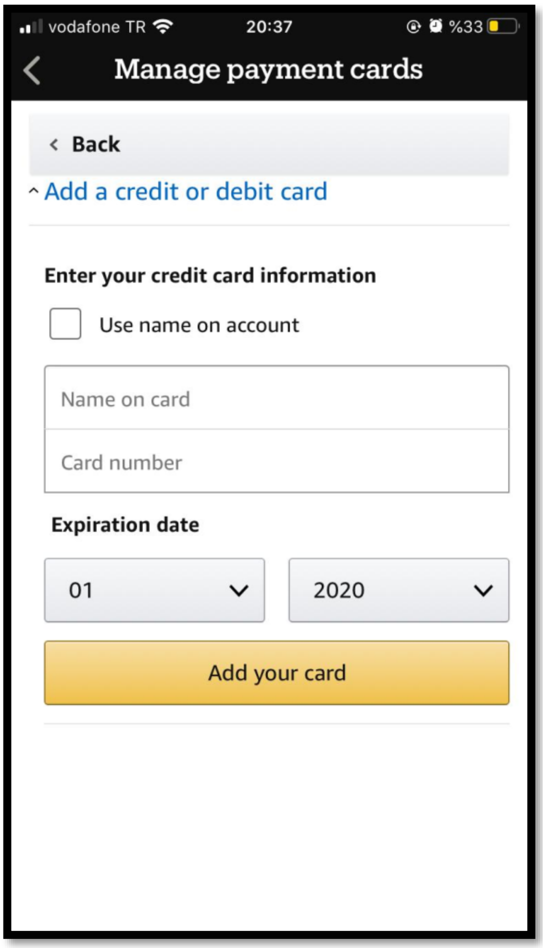
By signing in you are agreeing to our Terms of Use, Conditions of Use and our Privacy Notice.

New to Amazon?

Create a new Amazon account

© 1996-2020, Amazon.com, Inc. or its affiliates

Figure 1: Sign-in Interface



Manage payment cards

‹ Back

Add a credit or debit card

Enter your credit card information

□ Use name on account

Name on card

Card number

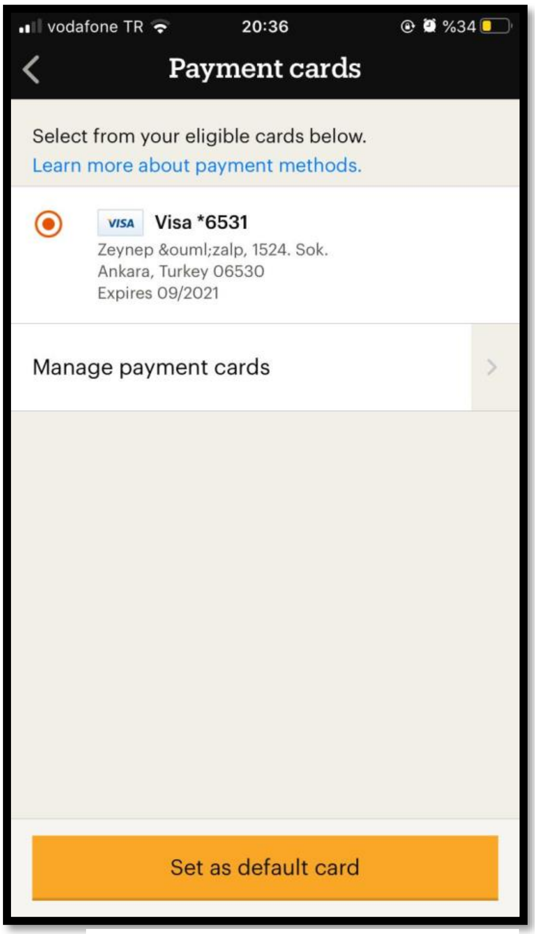
Expiration date

01

2020

Add your card

Figure 3: Addition of a Credit Card



Payment cards

Select from your eligible cards below.

Learn more about payment methods.

● VISA Visa *6531

Zeynep &ouml;zalp, 1524. Sok.

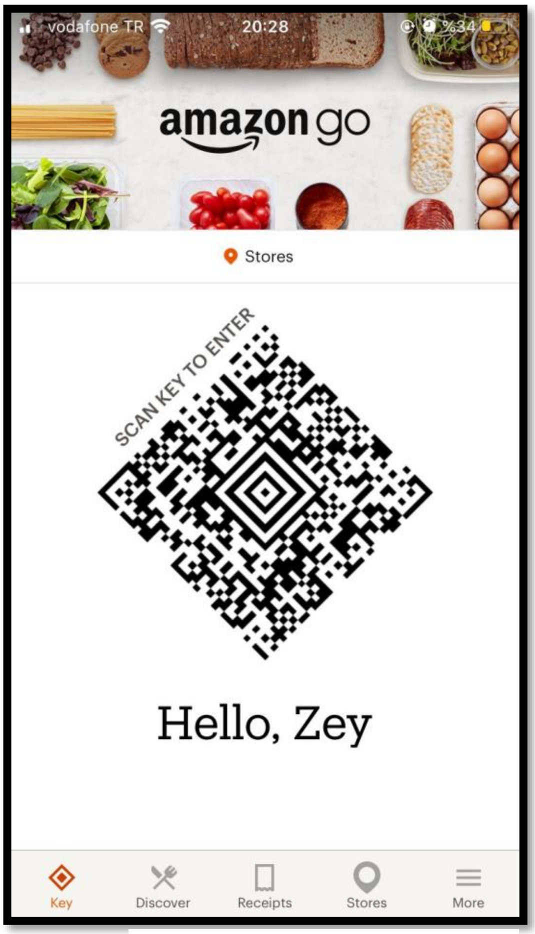
Ankara, Turkey 06530

Expires 09/2021

Manage payment cards

Set as default card

Figure 4: Payment Cards Interface



Hello, Zey

Figure 5: User Interface

1.3.2 System Functions

Functionalities of Amazon Go is displayed and summarized in the following table. Their detailed versions and description tables can be found in the 3.2 Functions section.

Function	**Summary**
Take/drop item	Virrtual cart of the user is updated if a product is taken or dropped.
Get information	Users get information about the Amazon Go system in the user interface.
Send receipt	After the user leaves the store, receipt of the shop is sent to the user and notifies her.
Scan QR code	To enter the store, QR code that is uniquely identify the each customer is scanned at the gate.
Open turnstile	After receiving the confirmation of the permission from the system, gate system open the turnstile in front of the user.
Restock shelves	Workers at the team of associates restock the empty shelves.
Ask Questions	Users ask questions about the system and application to the team of association.
View system logs	IT Staff views the system related error logs in the IT Staff interface.
SKU optimization	Variety of products arranged by the AI to meet the product requests.
Product Location Optimization	Location of products arranged by AI to improve them and their sales rates.

Table 1: System Functions

1.3.3 User Characteristics

The users of the Amazon Go system can be categorized into two parts as user and IT Staff. User should have basic knowledge about mobile devices and tablets, also about Amazon Go application and Gate system. IT Staff should have an expertise in their fields and handle the problems accordingly.

1.3.4 Limitations

- **Regulatory policies:** Amazon Go database keep the personal and credit card information. These information should not be published to community or others.

- **Hardware limitations:** Hardware should provide high bandwith to support AI data stream, payment service and others.
- **Interfaces to other applications:** Amazon Go should be compatible with the Amazon Go application in order to maintain the contunity of the system.
- **Parallel operations:** Amazon Go system should be able to serve many people in parallel without any failures.
- **Audit functions:** System should be able to handle transaction of payments.
- **Control functions:** Database control function should be only available to suthorized workers. Any others should not be able to control any operation that may cause errors in the system.
- **Higher-order language requirements:** System should be written using a object-oriented programming language. Since the IT staff interface is web-based, HTML, CSS and a scripting language (JavaScript) is reqiured. For the mobile application, Swift and Java should be used. Also, Python is used for AI.
- **Signal handshake protocols:** HTTPS is required for web based applications. Telnet or SSH is used when http is not available. TCP/IP is used for database connections.
- **Quality requirements:** It is really crucial to keep the users' and AI data backed up and systems should do back ups regularly.
- **Critically of the application:** Application should not crush in order to keep the system going from the user's perpective.
- **Safety and security considerations:** Safety and security of the system should be ensured by admins. Database should not be easily affected by any types of attacks and failures.
- **Physical/mental consderations:** There is no possible physical/mental consderations.

1.4 Definitions

Term	Definition
AI	Artificial Intelligence
User	People who have signed in to the system
IT Staff	Staff that are working in the Information Technology department
Sensor Fusion	System for combining different data streams from cameras and weight sensors to get more accurate data
DBMS	Database Manament System
SKU	Stock Keeping Unit
API	Application Programming Interface

OS	Operating System
IOS	Iphone/Ipad Opearting System

Table 2: Definitions

2. References

This document is written with respect to the specifications of the document below:

29148-2011 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes --Requirements engineering.

Other sources:

Dhruv Grewal, Anne L. Roggeveen, Jens Nordfält, The Future of Retailing, Journal of Retailing, Volume 93, Issue 1, 2017, Pages 1-6, ISSN 0022-4359, https://doi.org/10.1016/j.jretai.2016.12.008.

3. Specific Requirements

3.1 External Interfaces

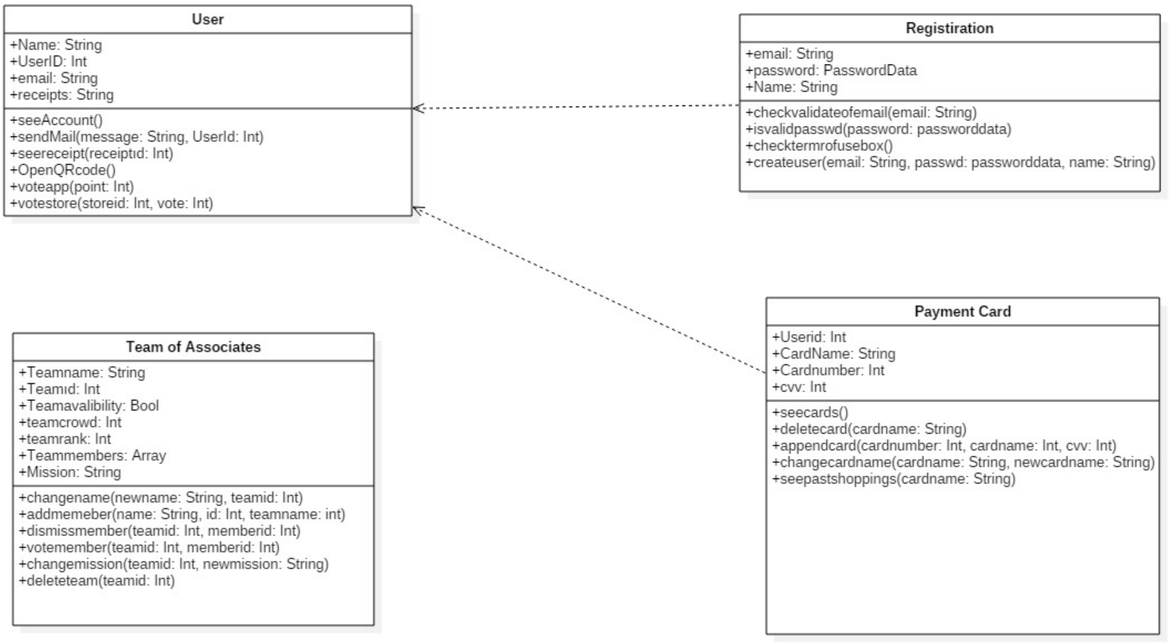

User
+Name: String
+UserID: Int
+email: String
+receipts: String
+seeAccount()
+sendMail(message: String, UserId: Int)
+seereceipt(receiptıd: Int)
+OpenQRcode()
+voteapp(point: Int)
+votestore(storeid: Int, vote: Int)
Registiration
+email: String
+password: PasswordData
+Name: String
+checkvalidateofemail(email: String)
+isvalidpasswd(password: passworddata)
+checktermrofusebox()
+createuser(email: String, passwd: passworddata, name: String)
Team of Associates
+Teamname: String
+Teamıd: Int
+Teamavalibility: Bool
+teamcrowd: Int
+teamrank: Int
+Teammembers: Array
+Mission: String
+changename(newname: String, teamid: Int)
+addmemeber(name: String, id: Int, teamname: int)
+dismissmember(teamid: Int, memberid: Int)
+votemember(teamid: Int, memberid: Int)
+changemission(teamid: Int, newmission: String)
+deleteteam(teamid: Int)
Payment Card
+Userid: Int
+CardName: String
+Cardnumber: Int
+cvv: Int
+seecards()
+deletecard(cardname: String)
+appendcard(cardnumber: Int, cardname: Int, cvv: Int)
+changecardname(cardname: String, newcardname: String)
+seepastshoppings(cardname: String)

Figure 6: External Interfaces Diagram

Registration Interface: This interface serves as a guide for non member users to become members. Non-member users enter name, surname and accept some contracts. After, this inputs go to database from application. All fields must be filled. Any invalid input gives error message like special characters or missing characters like “@” for email box. Also any empty area is not allowed otherwise sign in will fail. There is no time limit for this interface so timing is not crucial. However, when app close or refreshing attempt cause data to be lost. Very organized and simple interface. It consists of very few boxes to enter text inputs. All the input data are text inputs and click inputs. After the progress finished application direct the user to payment card interface.

Payment Card Interface: This interface is to assign a card to user to charge them for the products they buy. Users enter card number, card name etc. This data are saved and sent to database and checked for validity. If card number is invalid or if there is any inconsistency with card numberand name, application gives error mesages like inconsistent card number or inconsistent number name coupling. All data consist of some text inputs and some number inputs. There is no time limit for this interface so timing is not crucial. However, when app close or refreshing attempt cause data to be lost. Very organized and simple interface. It consists of very few boxes to enter text inputs. After the progress finished application direct the user to user interface.

User Interface: The purpose of this interface is to enable users to see some data like their key at the gate, receipts etc. and communicate with workers to get information. User inputs consist of clicks and some texts and this inputs goes application and application triggers related interface. They can take warning message if there is no data about clicked area. For example, if user did not do any shopping there is no data in receipts. There is no time limit for this interface so timing is not crucial.

Team of Associates Interface: The purpose of this interface is to maintain products availability and variety. This interface provide to manage the store effectively. The inputs of this interface comes from application and sensor fusion. Inputs consist of some orders with location of data and name, id of product. After order came, team of associates restock shelves witd appropriate location and appropriate products. After that, the output is mission completed or mission is not completed because not enought product or not enough worker to restocking and this outputs goes application. There is time limit after order comes, it is30 minutes. Inputs consists of orders and output consist of click boxes and text messages. End messages like mission is completed or not completed. Organized simple and easy to use interface.

3.2 Functions

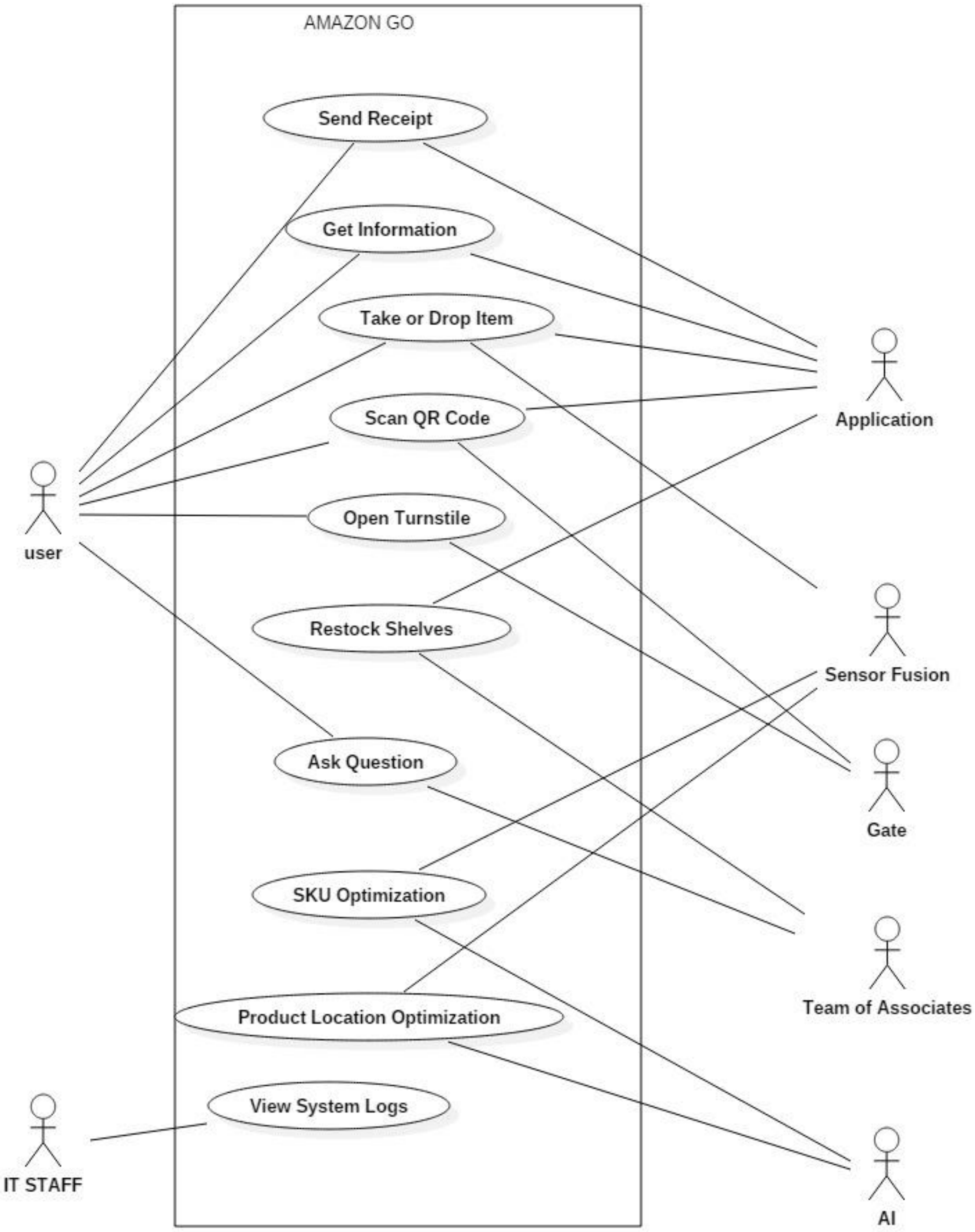


Figure 6: Use Case diagram

Use case name	Take/drop item
Actors	User, Application, Sensor fusion
Description	User takes/drops an product from the shelves, sensor fusion detects it and the product is added to/removed from the virtual cart of the user
Data	Weight sensor data, camera data, product id, user data
Preconditions	User should have an associated account
Stimulus	Taking/dropping the product from the shelves
Basic Flow	Step 1 - After taking/dropping the product, process the weight sensor and camera data in sensor fusion system Step 2 - Send the data from Sensor Fusion to Application Step 3 - Product is added to/removed from the user's virtual cart and displayed in the user interface
Alternative Flow	-
Exception Flow	The user hands over the product to someone but the product will stay on the user's cart or user puts the product to a different shelf
Post Conditions	Taken/dropped product is recorded in the DBMS and total cost is updated

Table 3: Take/Drop Item

Use case name	Send receipt
Actors	User, Application
Description	User receives the receipt via email after leaving the store
Data	User's email address, user's receipt address, products in the virtual cart, total cost
Preconditions	User should leave the store with product(s)
Stimulus	User walks out of the store
Basic Flow	Step 1 – Information that user left the shop comes to the application Step 2 – The data is collected from DBMS Step 3 – Receipt is displayed in the application and added to the database
Alternative Flow	-
Exception Flow	If the DBMS connection is lost, the IT staff gets a notification
Post Conditions	Application notifies the user and charges the card

Table 4: Send Receipt

Use case name	Scan QR code
Actors	User, Application, Gate
Description	User gets into the store by scanning the QR code which is displayed in the application at the gate
Data	QR Code, user data
Preconditions	User should have an appropriate QR code
Stimulus	User scans the QR Code with the scanner
Basic Flow	Step 1 – Scan QR Code Step 2 – Check if the QR code is in the database
Alternative Flow	-
Exception Flow	The QR Code may not be readable because of the obstacles in front of the screen (such as fingers) or the angles
Post Conditions	Send or do not send a permission to the system

Table 5: Scan QR Code

Use case name	Open turnstile
Actors	User, Gate
Description	Open turnstile in front of the store if there is a permission
Data	Permission
Preconditions	Permission need to be given
Stimulus	Confirmation of the permission by the system
Basic Flow	Step 1 – Open the gate
Alternative Flow	-
Exception Flow	Permission is not given or the gate is broken and do not open
Post Conditions	User enters the store

Table 6: Open Turnstile

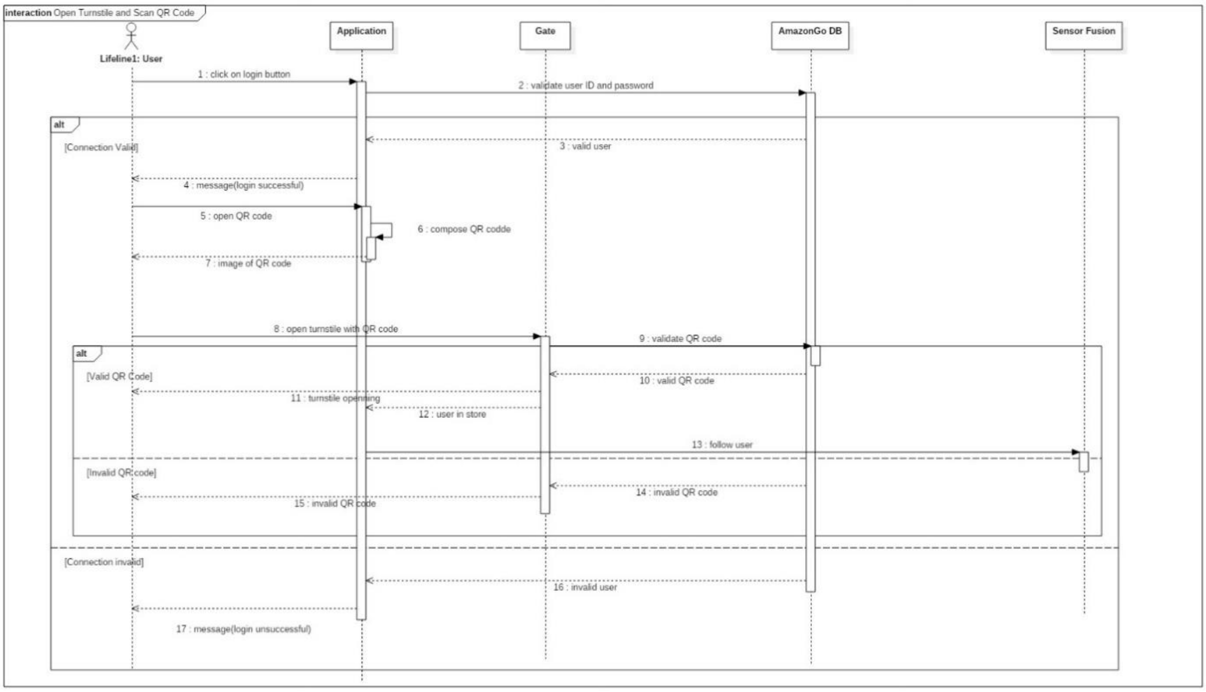


Figure 9: Open Turnstile and Scan QR Code Sequence Diagram

Actors	Get information
Description	User gets information about the system and application in user interface
Data	Scripts of questions and answers
Preconditions	User should have a signed up account and the application
Stimulus	User clicks the More > Help button
Basic Flow	Step 1 – Display the subtitles in the user interface Step 2 – User clicks the one of subtitles except Contact Us Step 3 – Display the scripts of questions and answers related to that subtitle
Alternative Flow	-
Exception Flow	Scripts data can not be provided
Post Conditions	User reads the scripts

Table 7: Get Information

Use case name	Restock Shelves
Actors	Team of associates, Sensor fusion, AI
Description	If any shelf is empty , it is detected by cameras and sensors then this shelf is refilled with new products

Data	Weight sensors data, cameras data, product id, shelf location
Preconditions	At least one empty shelf, available worker to refill shelves
Stimulus	At least a shelf became empty
Basic Flow	Step 1- Empty shelf is notified by fusion system Step 2- Fusion system send an a warning to system Step 3- System detect necessary product and location of shelve Step 4- System controls stocks for this product Step 5- System send an order to team of associates for refill shelve with necessary product Step 6- Team of associated refill shelve with necessary product Step 7- Sensor fusion detect shelve is refilled Step 8- System marks the warning as handled
Alternative Flow	Step 4-If product is not in stock, system gives an order for this product and wait until it is available in stock again
Exception Flow	If product no longer available anywhere, system notifies AI
Post Conditions	Shelf is refilled by products

Table 8: Restock Shelves

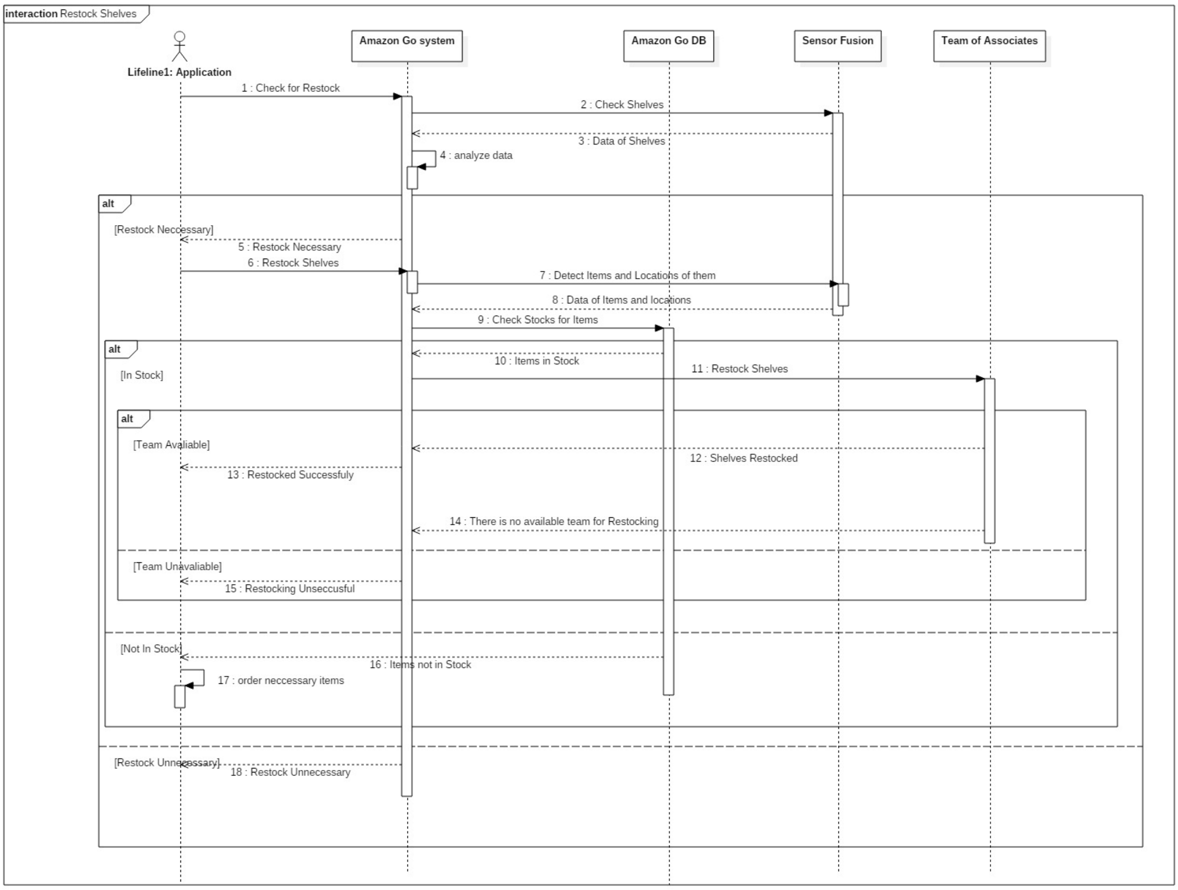


Figure 10: Restock Shelves Sequence Diagram

Use case name	SKU Optimization
Actors	Sensor fusion, AI, Team of associates
Description	AI keeps data that is sent from sensors and analyze it to improve product variety according to product demand
Data	Camera data, product id
Preconditions	System sends a receipt to improve efficiency of product variety
Stimulus	Product variety is not appropriate for customers
Basic Flow	Step 1- System sends a request to improve product variety Step 2-AI analyzes data which is sent from sensor fusuion system Step 3-AI detects necessary and unnecessary products Step 4-AI sends this data to system Step 5-System rearrangse product variety list
Alternative Flow	Step 5-If there is no change in product variety, system do not reaarange product variety

Exception Flow	-
Post Conditions	Products are arranged according to new list

Table 9: SKU Optimization

Use case name	Product Location Optimization
Actors	Sensor fusion, AI, Team of associates
Description	AI keeps data sent from sensors and analyze it to improve products' locations
Data	Camera data, product id
Preconditions	System send an receipt improve product location
Stimulus	Product location is not appropriate for customers
Basic Flow	Step 1- System sends a request to improve product locations Step 2-AI analyzes data which sent from fusuion systems Step 3-AI detects appropriate locations for each item Step 4-AI sends this data to system Step 5-System rearranges products locations according to this data
Alternative Flow	Step 5-If there is no change in product locations, system do not reaarange them
Exception Flow	-
Post Conditions	Products are arranged according to new data

Table 10: Product Location Optimization

Use case name	View System Logs
Actors	IT staff
Description	IT staff can list the system logs of Amazon Go and logs can be arranged by their date of formation and where the events take place. These logs include technical details of Amazon Go application.
Data	System logs
Preconditions	IT staff must be assigned and authorized to read system logs.
Stimulus	IT staff send a request to system for reaching system logs.

Basic Flow	Step 1-System logs request is sent to system. Step 2-System gives permission to IT staff. Step 3-Logs became visible in IT staff interface
Alternative Flow	-
Exception Flow	If connection or logs are lost, system notifies IT staff.
Post Conditions	System logs became visible on IT Staff's interface.

Table 11: View Sytem Logs

Use case name	Ask Question
Actors	User, Team of Associates
Description	Users can ask question to Team of Associates about the store, application, etc.
Data	-
Preconditions	At least one non-busy worker
Stimulus	-
Basic Flow	Step 1-User has a question Step 2-User finds an appropriate worker Step 3-User asks question to worker Step 4-Worker communicates with the user and answers the question
Alternative Flow	Step 3-If all workers are busy user waits someone to be avaliable
Exception Flow	If no worker is available in the store
Post Conditions	User's question is answered

Table 12: Ask Question

3.3 Usability Requirements

- Entering and exiting the system should be very fast and easy.
- It should be possible to get the necessary information by navigating within the system in 2-3 steps.
- Considering the order of priority, a system should be created and the first action should be done first and fastest.
- Users should be able to learn the system without any help and the system should be explained with simple instructions at the first entrance.
- Buttons should be of a size that is easy to see and simple readable descriptions should be placed on them.

- System admins should be able to easily enter the system and quickly see the errors they are looking for.
- There must be a search feature in the system to avoid a situation such as absence of a wanted thing.
- When the user enters anywhere, they should be able to easily go to the home screen.
- There should be a protection system in case the user runs out of charge or has any problems with his phone.
- The user should be able to get out of the system easily and quickly.
- Considering the disabled users, a system that they can use in them should be designed.

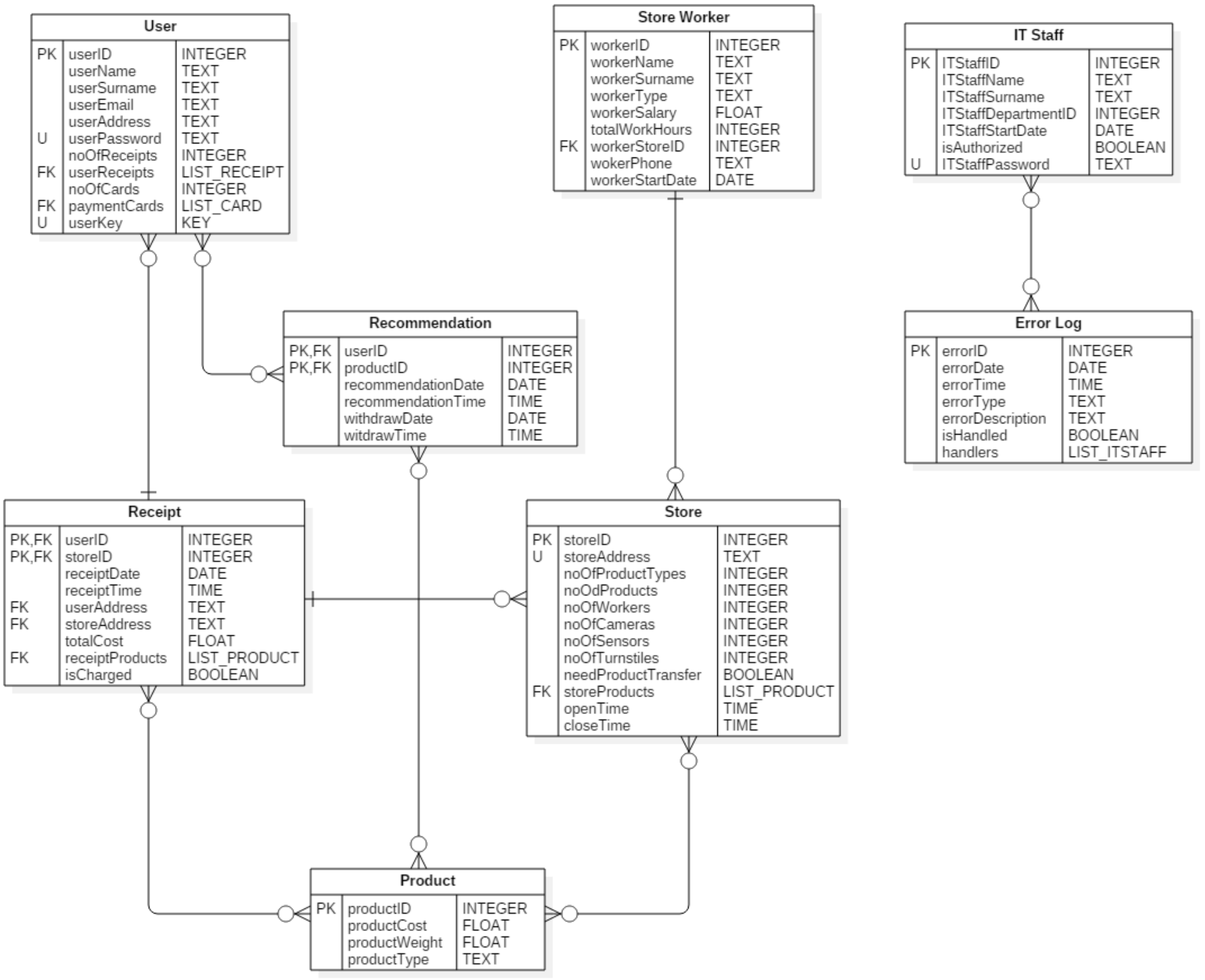
3.4 Performance Requirements

- The system must be able to protect user data and not allow others to gain access.
- More than 1000 users should be able to use the system at the same time.
- Even low internet speeds like 100 kbps should be enough to use the system.
- The data should be stored in at least 4 databases and protected against things like any natural disaster.
- The system should avoid unnecessary updates.
- The application should take up little space on the phone.
- A protection system should be developed considering the situations such as theft of the phone.
- The application should be customized so that personal information is not stolen, taking into account such situations as viruses on the phone.

3.5 Logical Database Requirements

- Only those workers whose worker type is “authorized” is able to view, filter and serach all the tables.
- New workers are added only by an authorized worker.
- Entries are added to the User table when an user is registered using regisration interface.
- User’s email, password and key have to be unique.
- User’s email, password and key shall be encrypted.
- User can have 0 or more receipts/payment cards and those are kept in the userReceipts/paymentCards lists.
- A payment card have to be validated before adding to the database.
- A receipts is added when user enters a store, take some products and leaves.
- Receipt entity shall be in one-to-many relationship with user entity. That is, a receipt belongs to only one user but an user can have multiple receipts.
- A product is added by store workers when they restock the shelves using the team of associates interface.
- Receipt entity is in many-to-many relationship with the product entity. That is, receipts may contain multiple products and a product can belong to multiple receipts. Be aware that a tuple in product table represents thousands of physical

products. For example, there is one record for “Classic Oreos” but there products. For example, there is one record for “Classic Oreos” but there are many in the store.



User

PK	userID	INTEGER
	userName	TEXT
	userSurname	TEXT
	userEmail	TEXT
	userAddress	TEXT
U	userPassword	TEXT
	noOfReceipts	INTEGER
FK	userReceipts	LIST_RECEIPT
	noOfCards	INTEGER
FK	paymentCards	LIST_CARD
U	userKey	KEY

Store Worker

PK	workerID	INTEGER
	workerName	TEXT
	workerSurname	TEXT
	workerType	TEXT
	workerSalary	FLOAT
	totalWorkHours	INTEGER
FK	workerStoreID	INTEGER
	wokerPhone	TEXT
	workerStartDate	DATE

IT Staff

PK	ITStaffID	INTEGER
	ITStaffName	TEXT
	ITStaffSurname	TEXT
	ITStaffDepartmentID	INTEGER
	ITStaffStartDate	DATE
	isAuthorized	BOOLEAN
U	ITStaffPassword	TEXT

Recommendation

PK,FK	userID	INTEGER
PK,FK	productID	INTEGER
	recommendationDate	DATE
	recommendationTime	TIME
	withdrawDate	DATE
	witdrawTime	TIME

Error Log

PK	errorID	INTEGER
	errorDate	DATE
	errorTime	TIME
	errorType	TEXT
	errorDescription	TEXT
	isHandled	BOOLEAN
	handlers	LIST_ITSTAFF

Receipt

PK,FK	userID	INTEGER
PK,FK	storeID	INTEGER
	receiptDate	DATE
	receiptTime	TIME
FK	userAddress	TEXT
FK	storeAddress	TEXT
	totalCost	FLOAT
FK	receiptProducts	LIST_PRODUCT
	isCharged	BOOLEAN

Store

PK	storeID	INTEGER
U	storeAddress	TEXT
	noOfProductTypes	INTEGER
	noOdProducts	INTEGER
	noOfWorkers	INTEGER
	noOfCameras	INTEGER
	noOfSensors	INTEGER
	noOfTurnstiles	INTEGER
	needProductTransfer	BOOLEAN
FK	storeProducts	LIST_PRODUCT
	openTime	TIME
	closeTime	TIME

Product

PK	productID	INTEGER
	productCost	FLOAT
	productWeight	FLOAT
	productType	TEXT

Figure 11: Logical Database Diagram

- Recommendation entity is in many-to-many relationship with both user and products. Recommendations show multiple products to multiple users using user interface and added by only AI.
- Store Worker entity is in one-to many relationship with Store entity and a record to the Store Worker is only added by an authorized worker.
- Only IT Staff is able to access the Error Log table.
- Record are added to the Error Log table by the system or IT Staff.
- Error Log entity is in many-to-many relationship with IT Staff entity.
- Store entity is in many-to-many relationship with Product entity.
- Receipt and Store worker is in one-to-many relationship with Store entity. That is, a receipt/worker is related to one store but a store is related to many receipts/workers.
- Database backup shall be done twice in a day.

3.6 Design Constraints

All information shall be stored for legal purposes. Keys and credit card information shall be kept as classified. The systems should be designed accoridng to laws.

3.7 Software System Attributes

a) **Reliability:** System shall be tested for reliability before a new upgrade is released. PRobability of data corruption shall not be exceed 0.01. Mean time of failure shall be one in 21 days.

b) **Availability:** System should be available all the time except for special cases. All the components should be restarted less than 3 minutes after a failure and IT Staff will be responsible for that. Sytem should be able to handel minor errors and be online.

c) **Privacy & Security:** Users' names, email adrresses, credit card numbers, keys and receipt adrresses should be safe and protected. Only authorized workers need to have an access to them.

d) **Maintainability:** Documentation of ecah components and bug fixing steps should be done. DevOps is responsible fort he maintenance.

e) **Portability:** Amazon Go application should be available for different mobile devices and tablets with IOS/Android. Application should be installed by the user. In desktop, IT staff and team of associates shall be able to see necessary information.

3.8 Supporting information

4. Verification

5. Appendices

5.1 Assumption and Dependencies

5.2 Acronyms an d Abbreviations