# Software Design Descriptions
for
AmazonGO

Burak Bolat (2237097)
İlter Taha Aktolga (2236891)

METU, 25. June 2020

Contents

Change History

Revision	**Date**	**Author(s)**	**Description**
1.0	25.06.2020	Aktolga I.,Bolat B.	SDD Complete

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Purpose of the system

In today's digitally connected world, people are not willing to waste their time by waiting in long queues. Scientists and engineers have been seeking alternative ways to reduce these wasted times. One of the daily problems of this kind is shopping store lines. By using advanced perception system that includes sensor fusion, computer vision and deep learning algorithm, Amazon GO which aims to create a checkout-free shopping system, is introduced as a candidate to solve this problem. The main purpose of the system is to provide "Just Walk Out" technology. In other words, taken and returned items for each customer are tracked in a virtual shopping cart. By doing so, the customer can just walk out whenever he/she is done with shopping. The receipt will be sent on the app and charging will be done via the Amazon GO account.

1.2 Scope

- The system will have a mobile app for communication with the customer. The customer gets in the store, checks his/her virtual shopping cart, shows accounts those are linked to his/her account via this app.
- The system will have an interface for sensor maintenance system to maintain perception system since perception system depends on sensor fusion. Via the sensor maintenance system, if a non-functional sensor (temperature, camera, etc.) is detected, it will be reported to the related employee.
- The system will communicate with Perception System which will give information on which item is taken or returned and by whom this action is taken. Apart from taken information, System will address action on Management System to keep the virtual shopping cart updated.
- The system will have an interface with Management System which keeps different virtual shopping carts for each customer, actual stock information for items and an interface for the store manager. Via Management System, System can request action for a customer and his/her virtual shopping cart.
- The system will authenticate customers by validating the QR code generated on their mobile phone. Then turnstiles will allow the customer to walk into the store.
- System should track misplaced products on the shelves and warn the employees to avoid confusion.
- The system will interact with Payment System to charge purchases and sends the bill to the customer's mobile phone when the customer leaves the store. Perception System is responsible for detecting leave action.

1.3 Stakeholders and their concerns

In Amazon GO, the commercial concern and research concern co-exist. On the commercial side, there exists store manager, store staff and customers. On the other hand, since Amazon GO includes numerous sensors, novel technologies and state of art estimation algorithms, researches are involved.

Customers: Customers are who enters the store, purchases items by just picking them and walks out. Their concern is not involving sophisticated purchase steps. Keeping customer interface simple to enter the store is their first concern. The second concern is tracking their shopping cart via an user-friendly interface. Lastly, fast and usable payment interface is their last concern.

Store Staff: Store staff has various work description such as sensor maintaining staff and SO staff. Their concerns are being assigned to well defined tasks, following assigned tasks easily and sending reports related with their task without exerting unnecessary effort.

Store Managers: Store managers are responsible for managing the store. Their first concern is examining updated stock. Secondly, their concern is manipulating item related tables in the database. Additionally, having control on employee related tables in the database. Obviously, since they are highly associated with database operations, they should have usable database interface.

System Developers: System Developers are the engineers specialized in various technical fields such as vision, security and IOT. They are responsible for developing the system. Their concern is being given well-defined requirements.

Researchers: Researchers are the people whose concerns are to collect data from the AmazonGO system and analyze using the state-of-art ML and DL methods to test their models on real-life scenarios such as activity prediction and stock price analysis.

Chapter 2

References

This document is written with respect to IEEE 1016-1998 standard:

IEEE standard for information technology–systems design–software design descriptions. (2009). New York, NY: Institute of Electrical and Electronics Engineers.

Other sources:

Day, M. (2018, January 22). Amazon Go cashierless convenience store opens to the public in Seattle. Retrieved from https://www.seattletimes.com/business/amazon/amazon-go-cashierless-convenience -store-opening-to-the-public/

Bosa, D. (2018, January 22). Amazon's automated grocery store will launch Monday after a year of false starts. Retrieved from https://www.cnbc.com/2018/01/21/amazon-go-automated-grocery-store-is-poised- to-launch.html

Chapter 3

Glossary

Term	**Definition**
App	Abbreviation for Application
CRUD	Abbreviation for Create/Read/Update/Delete operations which are used in database management.
DBMS	Abbreviation for Database Management System
DL	Deep Learning.
HTTPS	Abbreviation for Hypertext Transfer Protocol Secure
ID	Identity
ML	Machine Learning.
MYSQL	Open source relational database management system that runs as a server providing multi-user access to a number of databases
Perception	General term used for estimation models, sensor fusion systems and pattern recognition models.
SATA	Serial Advanced Technology Attachment which is used to transmit data from server to storage unit
SO	Abbreviation for Shelf Organization
SO staff	Employee who is responsible for refilling, maintaining store design

Table 3.1: Glossary

Chapter 4

Architectural Views

4.1 Context View

In this viewpoint, context of the system with all actors are defined in general and detailed viewpoints. In the context diagram, actors and their interaction with the Amazon Go will be explained in general terms. Use case diagrams and the detailed explanations of some possible use case of the system will be specified below the context diagram.

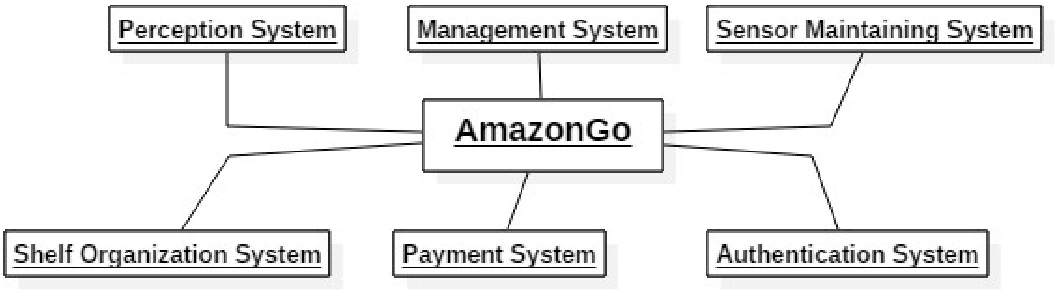

Figure 4.1: Context Diagram

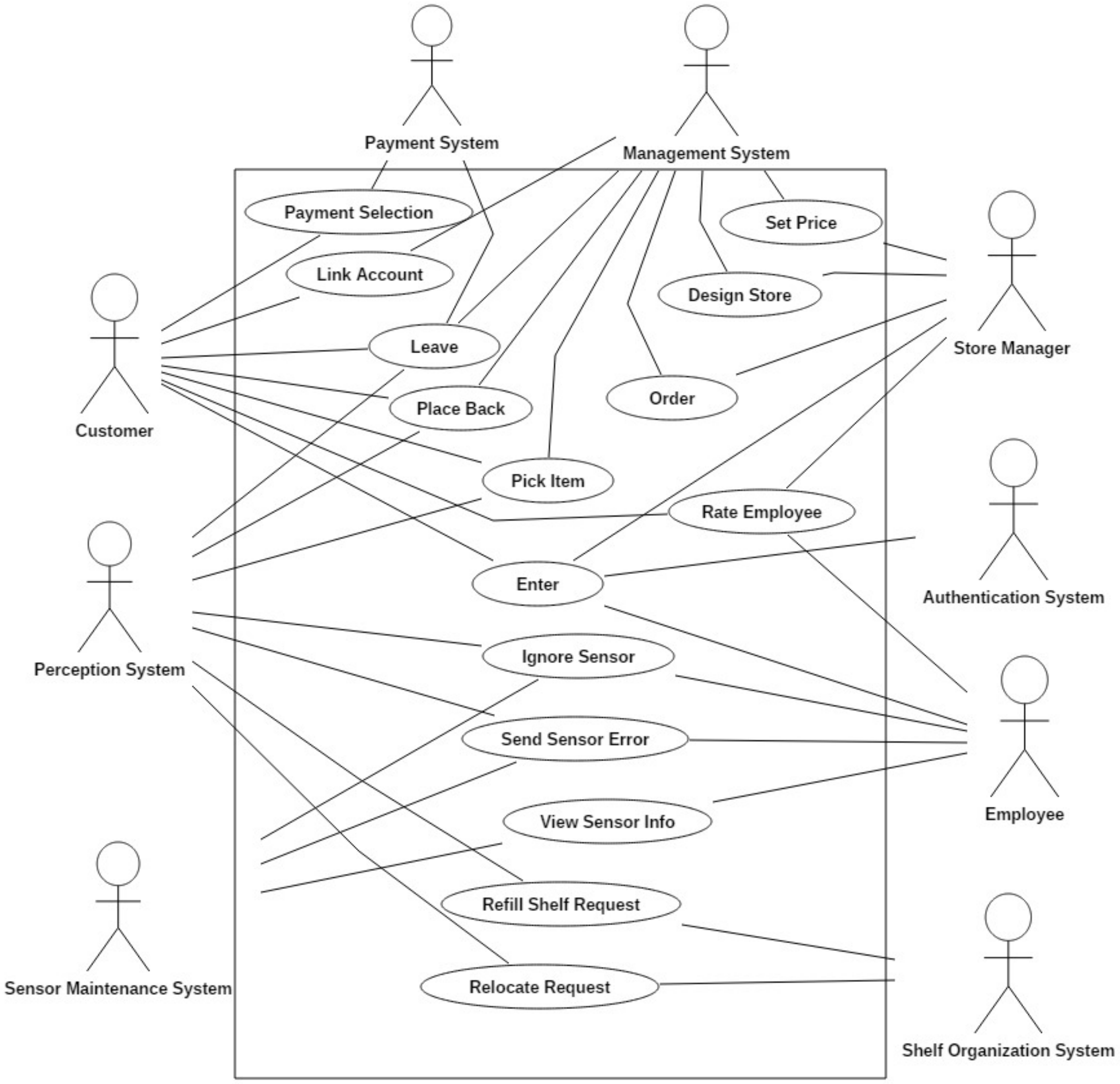

Figure 4.2: Use Case Diagram

Use Case Name	Link Account
Actors	Customer, Management System
Description	There may be multiple users who want to pay through one account. In that case, a user can link her/his account with other users.
Data	The user account of a master user, the User account that desired to be linked with
Preconditions	User has logged in the app.
Stimulus	Linking Request by clicking create link button
Basic Flow	Step 1: App asks for login again to satisfy security requirements. Step 2: The system waits for the target customer id from the customer. The target customer id is accepted via the app. Step 3: Linking request is passed to Management System with master and target customer ids. Step 4: Validation check for target customer is made. Step 5: An acceptance message is sent to the target customer. Step 6: The shopping cart of target customer attached to the master's cart.
Alternative Flow	Step 5: If a validation check for target customer fails, the error message is shown via app to the customer. Step 6: The system turns back to Step 2 in Basic Flow.
Exception Flow #1	If a user can not log in again as stated in the first step, it is written to the Log File. Login trial number is increased by one. In case this number exceeds the predetermined number, the customer is logged out from the App.
Exception Flow #2	If the target customer denies linking request, the whole process is cancelled. Master customer is informed about the rejection of his/her request.
Post conditions	Group list with all linked users is shown to master customer via the app.

Table 4.1: Link Account

Use Case Name	Enter
Actors	Customer, Authentication System, Employee, Store Manager
Description	When a person decides to enter the Store, Amazon Go sends a QR code to the app on the person's phone.
Data	Login time and date, User credentials
Precondition	Customer must be a verified customer.
Stimulus	Logging in the app
Basic Flow	Step 1: App sends an entrance request Step 2: Virtual shopping cart is created on Management System Step 3: The authentication system is awakened with customer id and date of entrance. Then, it creates a QR code with a timestamp. Step 4: Created QR code is sent to the customer via app and turnstiles with a timestamp. Step 5: Turnstile matches QR code which is on the screen of customer's mobile device with its QR database. If they match, it checks whether QR code is out of date by using a timestamp.
Alternative Flow	Step 3: If a customer is in a group and not the master of the group, the master is asked for his/her acceptance for merged payment. Step 4: Customer is notified with the response of the previous step. Afterwards, execution continues from Step 3 of Basic Flow.
Exception Flow	If the timestamp of QR code is out of date, the system asks for sending a new QR code to customer via the app.
Post conditions	When the entrance is successfully done, the app switches the shopping cart view.

Table 4.2: Enter

Use Case Name	Leave
Actors	Customer, Perception System, Management System, Payment System
Description	When a customer decides to leave the Store, Payment System is triggered to invoice.
Precondition	-
Data	Shopping Cart of the customer, User card info
Stimulus	Leave info from the Perception System
Basic Flow	Step 1: Perception System informs the AmazonGo about leave action of the customer. Step 2: Amazon Go requests invoice from Management System. Step 3: The amount taken from the invoice is passed to Payment System. Step 4: Payment System asks for payment type which can be cash or online via the app.
Alternative Flow #1	Step 5: Authentication System creates a QR code for cash payment. Created QR code is sent to the customer via the app and cash payment kiosk Step 6: Same as entrance, the customer has his/her QR code scanned to the kiosk. Step 7: Shopping cart is shown on the screen of the kiosk to the customer and waits for verification from the customer. Step 8: After the customer verified his/her shopping cart, kiosk waits for payment.
Alternative Flow #2	Step 5: Payment System directs payment request to the intermediary payment system which will handle payment request and bank interaction. Payment System can not handle this action without intermediary system since it is prohibited legally. Step 6: Stocks are decreased according to purchased items. Furthermore, the endorsement of the store is updated.
Exception Flow	Customer may not have enough money in his/her bank account or enough cash with him/her. Another case is that customer can select cash payment and just walk through. These cases are called no payment case. In no payment case, the customer is informed via the app and sent a message. Also, the Store Manager is informed with customer credentials and his/her invoice.
Post conditions	A copy of an invoice is sent to the customer via the app. In the case at least one of the purchased item has low stock, The Store Manager is notified with the item and its remaining amount.

Table 4.3: Leave

Use Case Name	Pick Item
Actors	Customer, Perception System, Management System
Description	Customer picks an item from any shelf that he/she wants. Perception System perceives taken item. Then, it sends item info to the Management System. Finally, the Management System adds the item to the customer's shopping cart.
Precondition	-
Data	Item info, Customer info
Stimulus	Picking action detected by sensors while taking an item from a shelf
Basic Flow	Step 1: Perception system awakes the Amazon Go with the information of customer and item that has been picked. Step 2: Amazon Go passes that data to the Management System. Step 3: Management System adds item given by Amazon Go to the related customer. Moreover, it returns whether the action is successful to the Amazon Go and other additional information.
Alternative Flow #1	Step 4: If the return value of the Management System indicates that the given item is not found, Amazon Go sends a notification to the customer via the app. After notification being shown, the app asks for manual adding for the undetected item.
Alternative Flow #2	Step 4: The app is updated with a shopping cart and the total amount.
Exception Flow	The picked item shouldn't be given to another customer. In that case, the picked item remains in the shopping cart of the first customer who has picked the item.
Post Condition	If any low amount of information is taken from the Management System, Store Manager is informed.

Table 4.4: Pick Item

Use Case Name	Place Back Item
Actors	Customer, Perception System, Management System
Description	A customer left free to put an item back where he/she wishes for.
Precondition	-
Data	Item info, Place back location
Stimulus	Human action estimator output, Sensor info that gives an item put into the shelf
Basic Flow	Step 1: Perception System awakes Amazon Go with the information of customer and item that being placed back by the customer. Step 2: Amazon Go informs Management System. Step 3: Management System removes the item from related customer's shopping cart and returns whether the action is successful. Step 4: Amazon Go updates shopping cart on the app. Step 5: The Management System is updated with an increment of item.
Post Condition	In case the customer leaves item to the wrong place, Perception System takes place and makes "relocate request" which is another use case.

Table 4.5: Place Back Item

Use Case Name	Relocate Request
Actors	Perception System, SO System
Description	It is possible to have such a situation that a customer places back an item to an undesired position in the store. Perception System detects these items. Furthermore, it sends a warning to the SO System which will handle the Relocating Operation.
Precondition	Beforehand, Place Back Item function is called. Afterwards, Perception System checks for the item's current place, and, if it is placed wrong place, this function is called.
Data	Current location of wrong placed item, the correct shelf of the wrong placed item
Stimulus	The estimation that reveals an item is in the wrong position
Basic Flow	Step 1: Perception System notifies Amazon Go with credentials of the item, current location of the item and correct location of the item. Step 2: Amazon Go searches in available employees for the title "stand organizer". Step 3: List of stand organizer list is sorted by the workload of employees in increasing order. The workload for an employee is the sum of estimated times of his/her tasks. Step 4: Relocate Request is assigned to first three (or determined another number) employee from the list. Assigned three employee is notified with request. Step 5: First Employee who accepted task is assigned to the task. Assignment is recorded with its starting time, due time, description and credentials of assigned employee. Also, task list of the employee is updated with relocate task. Step 6: Remaining two employees are notified with that the task is taken.
Alternative Flow	Step 5: In case, a predetermined time for accepting task is passed and any of the picked three (or determined another number) employee has not accepted the task, the System picks the following three (or determined another number) of employees from the list. This step reoccurs until an employee accepts the task. Step 6: All employees that are asked for accepting the task and not accepted it is notified with that the task is no longer available.
Exception Flow	If the available list of stand organizer is ended and there is no employee assigned for the task, the Store Manager is warned with that situation.
Post Condition	Employees interface is updated with his/her task, remaining time of tasks and priority of tasks.

Table 4.6: Relocate Request

Use Case Name	Send Sensor Error
Actors	Perception System, Sensor Maintenance System, Employee
Description	In case, a sensor becomes non-operational, Perception System informs Amazon Go . Then, Amazon Go creates a report for the Sensor Maintenance System which will fix the error. Although Perception System catches errors well enough, sending error messages by employees is allowed because there are still some errors. For instance, the temperature of a refrigerator sends incorrect values to the Perception System. One way to catch that is by observing the melting. This is an insuperable problem for the Perception System. At that point, catching the error is the responsibility of the employee.
Precondition	The sensor is not reported as non-functional. If so, this request is not created.
Data	Non-operational sensor identity, Error description
Stimulus	Connection loss or data loss that belongs to a sensor, "Create Sensor Error Report" button in the employee app
Basic Flow	Step 1: Perception System directs sensor id, sensor location and sensor type of the non-functional sensor to the Amazon Go. Step 2: Amazon Go decides the priority of error by using the predetermined impact of the sensor in perception, a number of items that are non-functional in near area and type of error. Step 3: Amazon Go sends a report to Sensor Maintenance System with sensor credentials, error time, error type, error time and priority of sensor. Step 4: The error instance is kept in the system until it is fixed and whenever the due date is close it notifies Sensor Maintenance System and Store Manager.
Alternative Flow	Step 1: An employee sends sensor error report via the employee interface. The report contains sensor id, sensor location and sensor type of the non-functional sensor. Execution continues from Basic Flow Step #2

Table 4.7: Send Sensor Error

Use Case Name	Refill Shelf Request
Actors	Perception System, SO System
Description	To maintain service at the store, shelves must be refilled when they are empty or not far away from being empty. Sensor data taken from shelves are used to sense this case. As a result, the Perception System sends "Refill Request" to the Management System.
Precondition	Sensor group that is creating this request must be functional. In other words, if a few sensors in the group is reported to Sensor Maintenance System and Amazon Go still has an open issue for these sensors, the request is not processed.
Data	Shelf identity, Fulfillment of the shelf
Stimulus	Low item signal comes from a shelf
Response	Report of request that contains which item will be refilled with the amount of it and shelf identity, Validity of the request
Basic Flow	Step 1: Perception System sends Refill Request with the information that consists of shelf location and type, item credentials and amount of item. Step 2: Amazon Go checks the item whether it is in the stock via Management System. This is called the validity of the request. If so, the Amazon Go notifies SO System to refill. Step 3: Amazon Go store searches in available employees for the title "stand organizer". Step 4: List of stand organizer list is sorted by the workload of employees in increasing order. The workload for an employee is the sum of estimated times of his/her tasks. Step 5: Relocate Request is assigned to first three (or determined another number) employee from the list. Assigned three employees is notified with a request. Step 6: First Employee who accepted task is assigned to the task. An assignment is recorded with its starting time, due time, description and credentials of the assigned employee. Also, the task list of the employee is updated with relocate task. Step 7: Remaining two employees are notified with that the task is taken.
Alternative Flow	Step 6: In case, a predetermined time for accepting task is passed and any of the picked three (or determined another number) employee has not accepted the task, the System picks the following three (or determined another number) of employees from the list. This step reoccurs until an employee accepts the task. Step 7: All employees that are asked for accepting the task and not accepted it is notified with that the task is no longer available.
Exception Flow #1	If the available list of stand organizer is ended and there is no employee assigned for the task, the Store Manager is warned with that situation.
Exception Flow #2	If the validity of the request returns false, i.e. item is out of stock, Management System is responsible for ordering the item. Amazon Go store notifies Store Manager to take action.
Post Condition	After refill request is done, the employee states that. The amount of the item in the shelf is updated.

Table 4.8: Refill Shelf Request

Use Case Name	Set Price
Actors	Store Manager, Management System
Description	Store manager set new prices for items whenever he/she wants.
Precondition	Store Manager is logged in the Management System.
Data	Item description, New price
Stimulus	Selecting a new price in the management application
Basic Flow	Step 1: Store Manager asks Amazon Go to set the price of an item. He/she selects an item from a list or gives item id. Step 2: Amazon Go calculates the percentage of the change in the price. Step 3: If change is small in the percentage, Amazon Go sends a request to the Management System. Step 4: After Management System is updated, Amazon Go asks Management System for customers who are currently in the store and has already picked that item. Also, Amazon Go request shopping cart update for the customers in the response. Step 5: Amazon Go updates shopping carts on the app of customer who is in the return of Step #4.
Alternative Flow	Step 3: If change is big in the percentage, Amazon Go needs to log in again by the Store Manager. Step 4: If the action is discount execution follows Basic Flow Step #4. Otherwise, if the action is marking up the price, the customers in the return of Basic Flow Step #4 purchase that item with the price when they have picked the item. Furthermore, the old price is in use until the end of this Alternative Flow. Step 5: An employee is notified to change the price tag on the shelf, which has high priority.
Exception Flow	If the Store Manager logged in and tries to make a big change in price and then fails to log in again, the Store Manager has 2 more chances. Setting price is banned for a while in case of no remaining trial.
Post Condition	If there is a discount, a message that informs customer in the store is sent.

Table 4.9: Set Price

Use Case Name	Order
Actors	Store Manager, Management System
Description	Store Manager orders items.
Precondition	The Store Manager is logged in the Amazon Go
Data	Item description, Order amount
Stimulus	Clicking the "Order" button in the management application
Basic Flow	Step 1: Store Manager selects an item to order and decides the amount of it. Step 2: Order request passed to Management System and is checked if the item exists. After the item is found, the amount is also checked. The amount is limited by predetermined capacity. For instance, the Store Manager can not order more than warehouse capacity. Step 3: Management System returns that it is ready for order. Step 4: Amazon Go asks for Store Manager to log in again to meet with security requirements. Step 5: Amazon Go confirms the order and sends confirmation info to Management System. Step 6: Management System handles the order and returns a copy of the invoice to Amazon Go. Step 7: Store Manager is notified that order is done.
Alternative Flow	Step 3: If the item is not found or amount is higher than the threshold, Amazon Go Store warns the Store Manager. Execution restarts with selecting the item to order and amount.
Exception Flow	If security requirement does not meet, System locks ordering for a while and asks for restarting password.
Post Condition	Warehouse employees are notified with the order so that design warehouse appropriately.

Table 4.10: Order

Use Case Name	Design Store
Actors	Management System, Store Manager
Description	Store Manager can arrange items and which shelf they stands on via Management System.
Precondition	-
Data	Item ID, Shelf ID
Stimulus	"Edit Item Place" button on the Management System
Basic Flow	Step 1: Store Manager chooses an item. Step 2: Item info and Shelf info is returned from Management System. Step 3: Store Manager picks a new shelf from the list or creates a new shelf. Step 4: Management System edits database tables according to new information.
Exception Flow	Step 4: If chosen shelf is full, action rejected and store manager is asked to pick another shelf or create a new shelf.
Post Condition	Store Manager is informed and virtual store map is updated

Table 4.11: Design Store

Use Case Name	Payment Selection
Actors	Customer, Payment Selection
Description	Customer can select payment type whether "in shop" or "online" payment is used.
Precondition	Customer is logged in to app or website.
Data	Selection of payment type
Stimulus	"Approve Selection" button in app, "Approve Selection" button in website
Basic Flow	Step 1: Customer selects "Payment Options" tab in the app. Step 2: Customer fills the necessary fields for selected payment type and sends request to the system. Step 3: Request investigated and database is updated with payment type and information.
Alternative Flow	Step 1: Customer selects "Payment Options" tab in the website. Other steps are same with Basic Flow.
Post Condition	-

Table 4.12: Payment Selection

Use Case Name	Rate Employee
Actors	Customer, Store Manager, Employee
Description	Store Manager and Customer can rate an employee.
Precondition	Customer is logged in and verified.
Data	Rate, Customer ID, Comment
Stimulus	Rate button in interfaces of customer and store manager
Basic Flow	Step 1: Customer or Manager selects "Performance" tab in their interfaces. Then selects "Rate Staff" tab. Step 2: Due to security, authentication is needed again. Step 3: Store Manager picks an employee from the list of employees Step 4: Store Manager gives a rate and if desires, adds a comment.
Alternative Flow	Step 3: Employee list is not shown to the customer. Instead, description or name of the employee that will be rated is asked. Step 4: Customer gives a rate and if desires, adds a comment. Step 5: Employee Description or name is evaluated. If he/she is found, rate action is handled.
Exception Flow	Step 3: If authentication fails, report is created if authentication fails for second time. Moreover, action sender is notified that authentication fail and a second chance is given to authenticate.
Post Condition	After fair amount of time passes, employee informed with that a rate update is occured.

Table 4.13: Rate Employee

Use Case Name	Ignore Sensor
Actors	Perception System, Employee, Sensor Maintenance System
Description	Whenever a sensor or a group of sensors is nonfunctional, Perception System is informed not to involve the sensor when managing perceptions. Ignore action can be sent from an employee or Sensor Maintenance System.
Precondition	-
Data	Sensor ID
Stimulus	Invoking Perception System via its API
Basic Flow	Step 1: Employee or Sensor Maintenance System picks a sensor or sensor group to ignore. Step 2: Request via API is decoded and ignore action handled. Afterwards, result is returned.
Exception Flow	Step 3: If return is "ignore action failed", sender of the request is warned.

Table 4.14: Ignore Sensor

Use Case Name	View Sensor Info
Actors	Employee, Sensor Maintenance System
Description	Employee has the ability of monitoring sensors. View Sensor Info gives the sensor data through a given time interval for a sensor or a sensor group.
Precondition	Employee logged in has a work type that resides in technical category.
Data	Sensor data, Sensor groups
Stimulus	Request for sensor info
Basic Flow	Step 1: Employee selects from sensor list or types manually the id of sensor or group and selects if given id is a sensor or a group. Step 2: Employee sends a request to have sensor info. Step 3: If time interval is not given Sensor Maintenance System returns line chart of data from the starting datum.
Exception Flow	Step 2: If sensor or group are not found, Sensor Maintenance System sends error and turns back to Step 1.
Post Condition	Employee informed after updating Employee Interface by sending notification via interface.

Table 4.15: View Sensor Info

4.2 Composition View

In this view, components and interfaces between them are presented in a high level view. Detailed explanations are given in the related sections.

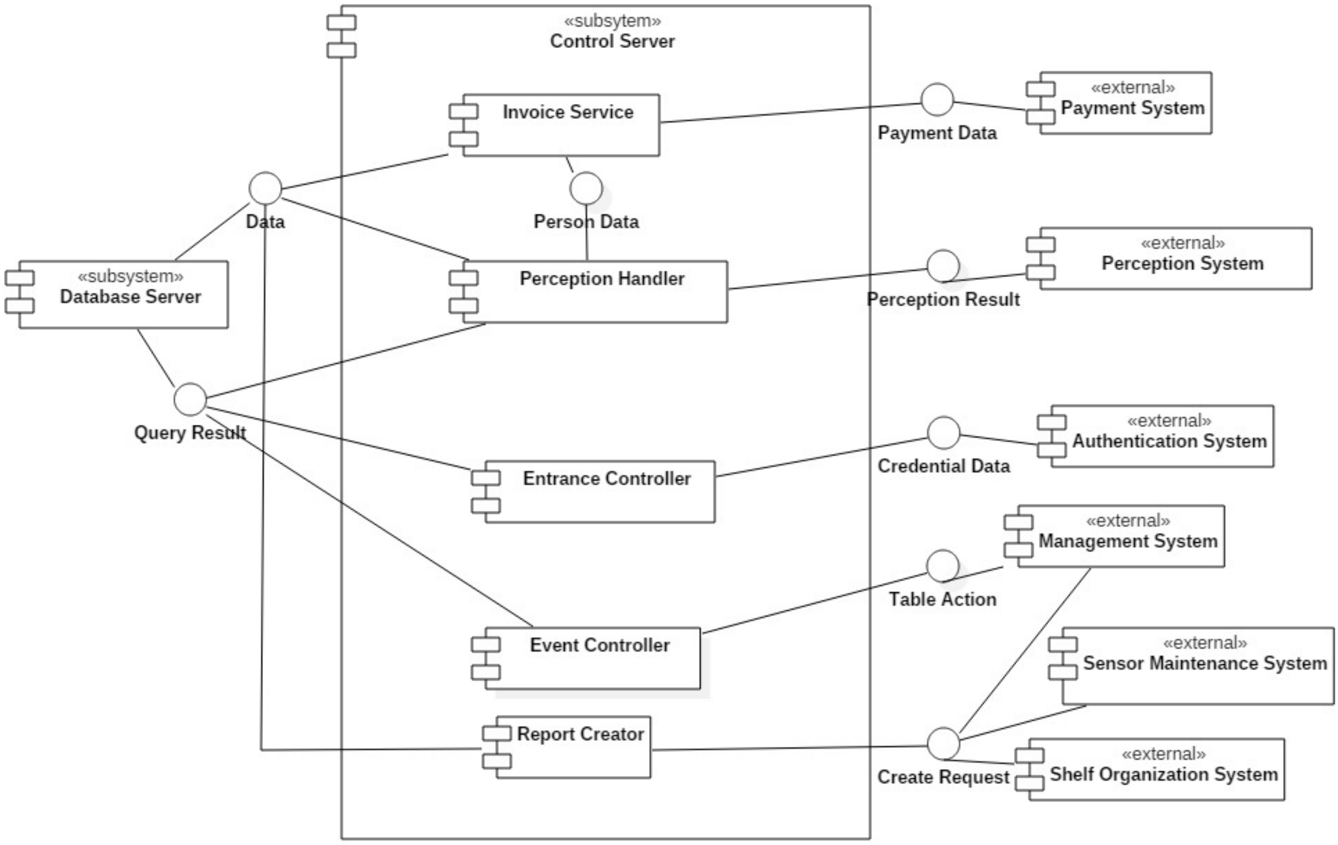

Figure 4.3: Component Diagram

Design Rationale:

- Control Server controls components which have external interfaces. It invokes components with respect to data and interrupts that it gathered. For security consideration, components have few interfaces between themselves, thus communication between components are mostly passed through Control Server.
- Control Server stores current store data such as members in store, action is taken, detected data. For instance, by using store data, it links items with customer, i.e. adds item to customer's cart.
- Control Server keeps temporary data. For instance, when Invoice Server fails to store transaction data in to database, Control Server stores transaction data for one day. In other case, events are stored in Control Server before updating database until event duration starts.
- Database Server keeps larger data that ruins the performance of Control Server such as human face, shopping cart, logs, etc.
- Invoice Service manages leaving action of customer. It is awaken by Control Server and gets data from Perception Handler and Database. Depending on gathered data it prepares an invoice for selected payment option. Then, it forwards the invoice to Payment System. After having payment result, it stores a summary of it in the Database and Control Server.
- Perception handler is invoked by the Perception System or Control Server. It can request from Perception System or Perception System requests to handle an estimation. Having perception, Perception Handler informs Control Server to decide which action will be taken up to perception.

- Entrance Controller takes control after Authentication System. It checks database in order to handle any error. Moreover, it controls the groups by using customer data of payment preferences and membership of a group, also Control System data if group master is in the store.
- Event Controller is responsible for scheduled events. It checks for time of an event and required database updates of an event. Even more, if event fails, it decides the action.
- Report Controller converts reports created by various external systems to valid report in the database. It notifies to Control Server, whether an action is needed for created report.
- Payment System is an external independent system. It takes invoice and customer preferences on payment to handle payment. It deals with kiosk for inStore payment and bank interfaces for online payment. Afterwards, it returns transaction data back to Invoice Service.
- Perception System is a collection of estimation and sensor fusion systems. Perceived human, action, item, etc. is transmitted to Perception Handler. It is open for Researchers to try their model.
- Authentication System is responsible for validation check of customer. It transmits result of validation to the Entrance Controller.
- Management System gives Store Managers ability to take decisions on store. It provides interfaces to manage the store. Via the interface, manager can charge for an item, adjust salaries, design store by editing shelf contents or position. Furthermore, Store Manager can schedule an event which is sent to Event Controller.
- Sensor Maintenance System provides infrastructure between sensors and technical staff. It logs the broken down sensors and creates reports for them which is assigned to a technical staff.
- SO System is awaken by data gathering from Store Manager, Employee or Perception System. It's task is providing an infrastructure between store organization employee and components which has ability on SO.

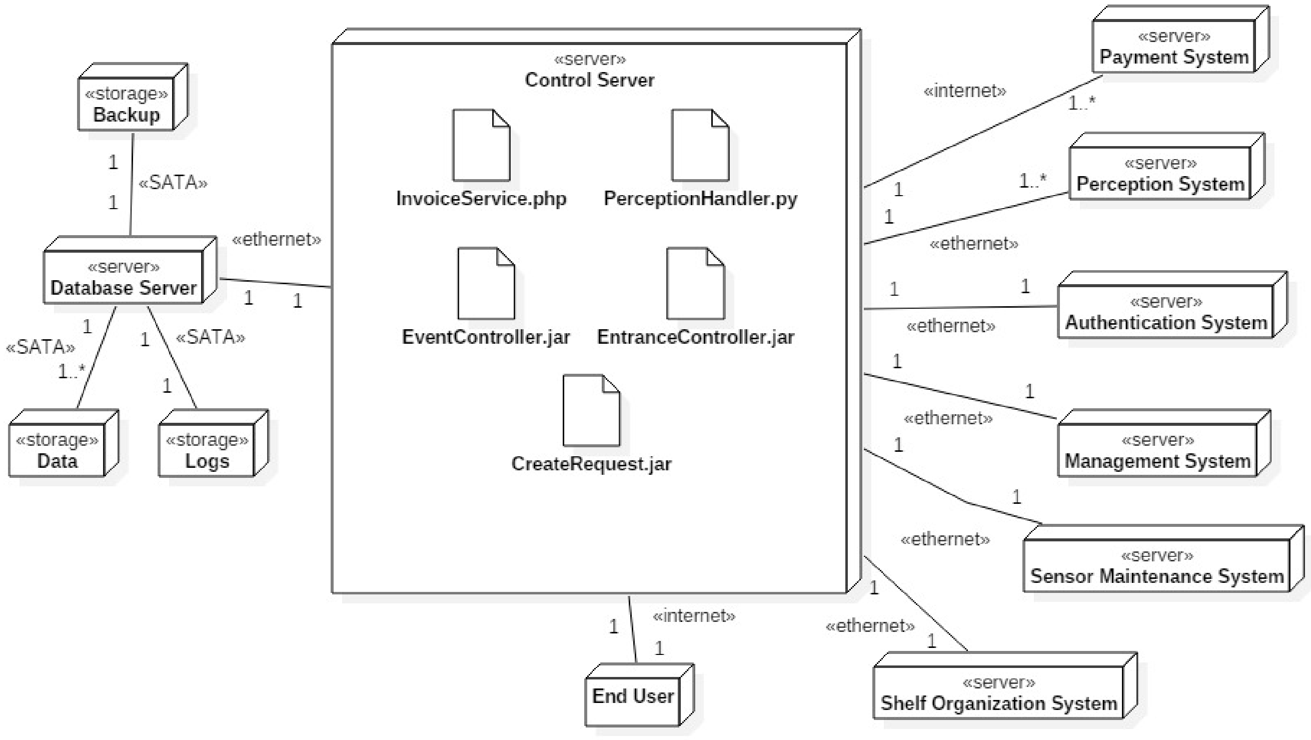

Figure 4.4: Deployment Diagram

Design Rationale:

- Java is used for communication components.
- Perception Handler is implemented by using Python. Since Perception System mostly uses Python, Perception Handler is implemented with same language to have extensively usage of Perception System.
- Payment System is dependent to bank organizations or kiosk, thus internet is used to communicate with it.
- Although Perception System, Management System, Authentication System, Sensor Maintenance System and SO System is external systems, they execute on computers which are in store. Consequently, Ethernet is used to have faster communication.
- End user is connected via internet by using app or website.
- Data as storage component stores human related data such as id, password, face image. Also, it stores transactions, shopping cart and item data.
- Logs stores entrance logs, reports of sensors and communication failure logs.
- Backup is for duplicating data to storage. System has backup time when store is closed at night.
- SATA is used to have a communication between server and storage fast enough.
- Connections which is internet based has https as protocol.
- MySQL is used to create and execute queries.

4.3 Information View

In this view, components and their structures which will be stored is expressed in terms of data manipulation, data transmission and data collection. Database operations are examined in this section.

4.3.1 Interfaces

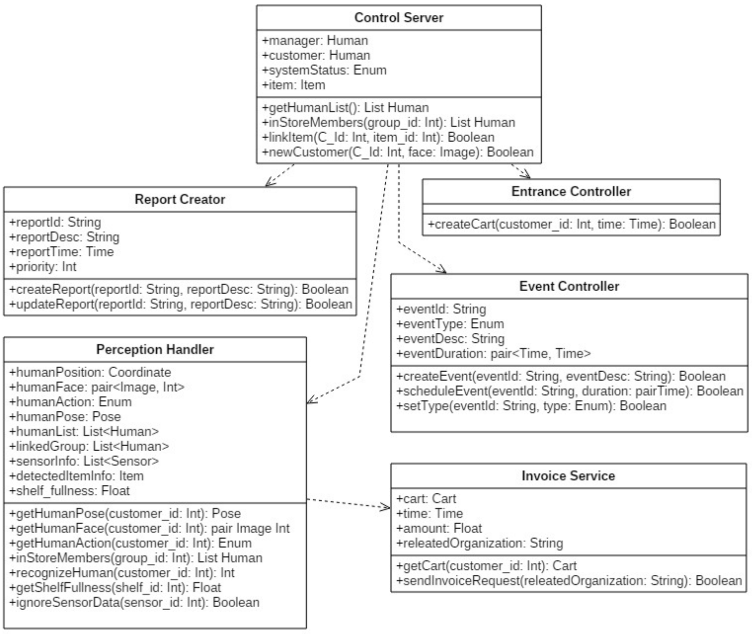
Control Server
+manager: Human
+customer: Human
+systemStatus: Enum
+item: Item
+getHumanList(): List Human
+inStoreMembers(group_id: Int): List Human
+linkItem(C_Id: Int, item_id: Int): Boolean
+newCustomer(C_Id: Int, face: Image): Boolean
Report Creator
+reportId: String
+reportDesc: String
+reportTime: Time
+priority: Int
+createReport(reportId: String, reportDesc: String): Boolean
+updateReport(reportId: String, reportDesc: String): Boolean
Entrance Controller
+createCart(customer_id: Int, time: Time): Boolean
Event Controller
+eventId: String
+eventType: Enum
+eventDesc: String
+eventDuration: pair<Time, Time>
+createEvent(eventId: String, eventDesc: String): Boolean
+scheduleEvent(eventId: String, duration: pairTime): Boolean
+setType(eventId: String, type: Enum): Boolean
Perception Handler
+humanPosition: Coordinate
+humanFace: pair<Image, Int>
+humanAction: Enum
+humanPose: Pose
+humanList: List<Human>
+linkedGroup: List<Human>
+sensorInfo: List<Sensor>
+detectedItemInfo: Item
+shelf_fullness: Float
+getHumanPose(customer_id: Int): Pose
+getHumanFace(customer_id: Int): pair Image Int
+getHumanAction(customer_id: Int): Enum
+inStoreMembers(group_id: Int): List Human
+recognizeHuman(customer_id: Int): Int
+getShelfFullness(shelf_id: Int): Float
+ignoreSensorData(sensor_id: Int): Boolean
Invoice Service
+cart: Cart
+time: Time
+amount: Float
+releatedOrganization: String
+getCart(customer_id: Int): Cart
+sendInvoiceRequest(releatedOrganization: String): Boolean

Figure 4.5: Interface Class Diagram

Operation	**Description**
getHumanList	Returns the human_list variable
inStoreMembers	Given the group_id finds the members that are currently in the store.
linkItem	This operation creates a connection between item and customer who takes it.
newCustomer	Creates and allocates new customer class with given parameters
createReport	Given report id and description, it generates a report.
updateReport	Given report id and new description, it updates the corresponding report
getHumanPose	Retrieves the human pose of the given customer id.
getHumanFace	This operation allows to reach the human face model of the customer
getHumanAction	Classified action of the requested customer can be accessed
recognizeHuman	Searches the human with given id in store cameras
getShelfFullness	Each shelf stores the fullness info. This operation returns the current fullness of the shelf with id.
ignoreSensorData	If the sensor reliability is under the tolerable threshold, this operation disables the sensor until it becomes reliable for further calculations.
createCart	At the given time, creates a shopping cart for given customer.
createEvent	With event ID and description, allocates a space for new event such as discount, campaign
scheduleEvent	For the given duration, finds the event with requested event ID, and enables it.
setType	sets or changes the type of the event
getCart	To complete invoice operation, fetches the shopping cart of the customer
sendInvoiceRequest	With given card info, sends the request to make purchase.

Table 4.16: Operation Descriptions

Operation	**Inputs**	**Outputs**	**Exceptions**
getHumanList	-	List of customers which contains perception info for each customer	No customer is in, check if store is open
linkItem	Customer id, item id	True if item is successfully linked, False otherwise	Item is found as "out of stock" No customer with given Customer ID
newCustomer	Customer id, Face	Returns a new customer object by initializing customer_id and Face data	Face data is not received Detection error occurs. No customer with given Customer ID
createReport	Report ID, report description	True if report is successfully created, False otherwise	Description is too long, size error
updateReport	Report ID, report description	True if the report is updated, False otherwise	Report is not found by given Report ID
getHumanPose	Customer ID	Pose of the customer with given ID	No pose data is found No customer with given Customer ID
getHumanFace	Customer ID	Face Data of the customer with matching ID	No customer with given Customer ID
getHumanAction	Customer ID	Classified action of customer	No customer with given Customer ID
inStoreMembers	Group ID	List of customers, each stored in a human object	Group is created but not verified yet
recognizeHuman	Customer ID	Returns 1 for successful detection, 0 for failed detection	No customer with given Customer ID
getShelfFullness	Shelf ID	Percentage of the fullness belongs to shelf with given shelf ID	Shelf can not be detected
ignoreSensorData	Sensor ID	1 for ignored sensors,0 for ignore failure	Sensor is already ignored
createCart	Customer ID, time	True for successful creation, False otherwise	A cart is created already, user must leave first
createEvent	Event ID, event description	1 for created event, 0 for ignored creation	Event Description is too long, size error
scheduleEvent	Event ID, Duration	1 for scheduled event, 0 for rejected operation	Duration is invalid
setType	Event ID, Event Type	1 after successful operation, 0 for failure	No permission to set given Event Type
getCart	Customer ID	Returns Cart class	No customer with given Customer ID Cart is expired
sendInvoiceRequest	Releated Organisation(e.g. VISA), Invoice in string format	Returns True for successful request, False otherwise	Organisation Service is not available

Table 4.17: Operation Design

Design Rationale:

- Multiple ignoreSensorData for same sensor or sensor group is executed only once.
- newCustomer operation calls getHumanList operation in order to check if new customer is already exists.
- Whenever getHumanAction returns "pick" action, linkItem is called.
- Control Server invokes Invoice Server whenever "leave" action is percieved. Then, Invoice Service use getCart, inStoreMembers and getHumanList.

4.3.2 Database Operations

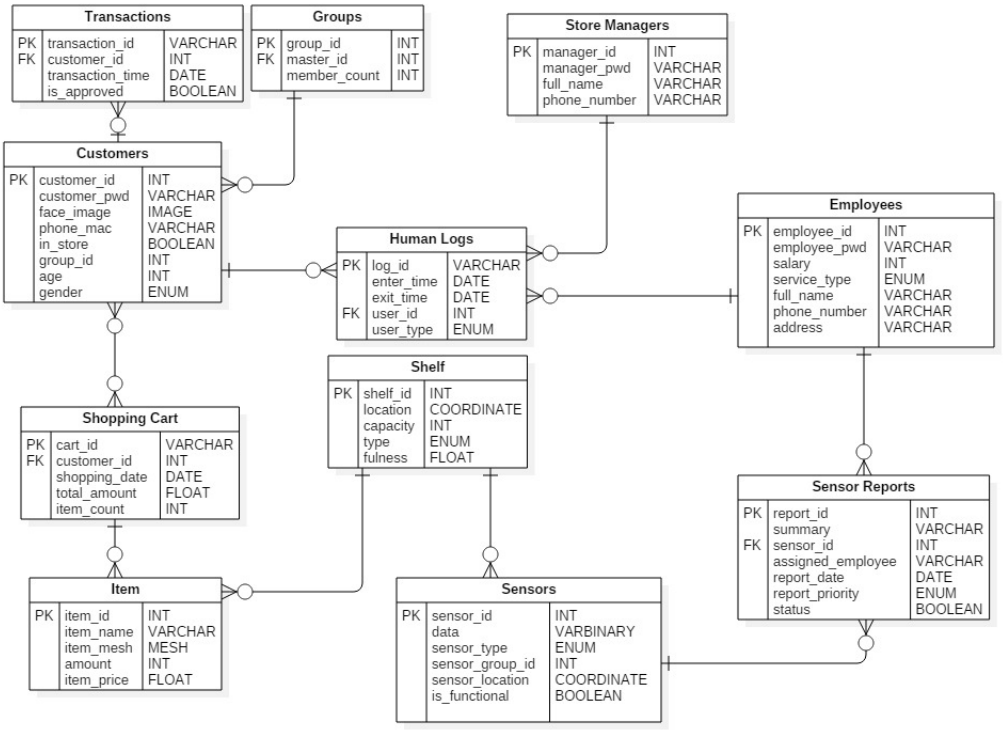

Figure 4.6: Database Class Diagram

First column	Second column
getHumanList	Create: - Read: Customers Update: - Delete: -
inStoreMembers	Create: - Read: Customers, Groups Update: - Delete: -
linkItem	Create: - Read: Customers, Item Update: Shopping Cart Delete: -

Operation	**CRUD Operations**
newCustomer	Create: Customers Read: - Update: Groups Delete: -
createReport	Create: Sensor Reports Read: - Update: Delete: -
updateReport	Create: - Read: - Update: Sensor Reports Delete:-
getHumanPose	Create: - Read: Customers, Human Logs Update: - Delete: -
getHumanFace	Create: - Read: Customers, Human Logs Update: - Delete: -
getHumanAction	Create: - Read: Customers, Human Logs Update: - Delete: -
recognizeHuman	Create: - Read: Customers Update: - Delete: -
getShelfFullness	Create: - Read: Shelf Update: - Delete: -
ignoreSensorData	Create: Sensor Reports Read: Sensors Update: - Delete: Sensors
createCart	Create: Shopping Cart Read: Customers Update: - Delete:-
getChart	Create: - Read: Shopping Cart, Customers, Item Update: - Delete: -
sendInvoiceRequest	Create: Transactions Read: Customers, Shopping Cart, Item, Groups Update: - Delete: -
createEvent	Create: - Read: - Update: Item, Shelf, Employees Delete: -
scheduleEvent	Create: - Read: -

Operation	CRUD Operations
	Update: Item, Shelf, Employees Delete: -
setType	Create: - Read: - Update: Item, Shelf, Employees Delete: -

Table 4.18: CRUD Operations

Design Rationale:

- MySQL is used to create and execute queries.
- linkItem reads amount of item and decrements. If item amount is 0, Shopping Cart table is updated, however, errors is sent to Store Manager. In case item amount is close to 0, Store Manager is informed.
- If newCustomer is not included in any group. Update on Groups table is ignored.
- Percepiton System uses Sensors table, thus whenever ignoreSensorData is used, sensor is deleted from table. When sensor is fixed, it is added table again.

4.4 Interface View

In this view, the internal interfaces between the components of the system and the external interfaces between the Amazon Go and the other systems will be specified in detail with design rationale.

4.4.1 Internal Interfaces

Interface between Event Controller and Database Server:
Event Controller waits the time for scheduled event. Events range from item price setting, salary setting for an employee to store design. Whenever it is time to execute it awakes the control server to create query and gets the query. It sends created query to database. Database returns if query is successfully executed or not. If query executed properly, event controller informs the store manager who creates the event and selected staff by creator. In case query fails, Event Controller decides according to type of event. For instance, if type is "crucial", it instantly warns the store manager. On the hand, types that have lower priority may choose to execute database query at later day.

Design Rationale:

- Event Controller has priority in database since it may have high priority queries.
- Query results are stored with cause of failure in database if it fails.

Interface between Perception Handler and Control Server:
Control Server will use Perception Handler in order to update cart of the customer with appropriate item. When a customer is detected and picking action is estimated, Control Server merges item data and customer data. The next step is updating shopping cart of the customer in database. Yet other action is placing back an item. In this case, perception handler passes customer and item information to the control server in order to delete item from customer's shopping cart and check other data whether estimation is valid. When a customer

is detected that she/he left the store, Control Server invokes Invoice Service.

Design Rationale:

- Perception Handler can not access control server data such as customer credentials since it has a external interface which can cause security vulnerability.
- Control Server can set Perception Handler parameters such as detection image resolution. When store has more customer than expected, these adjustments are available for control server.

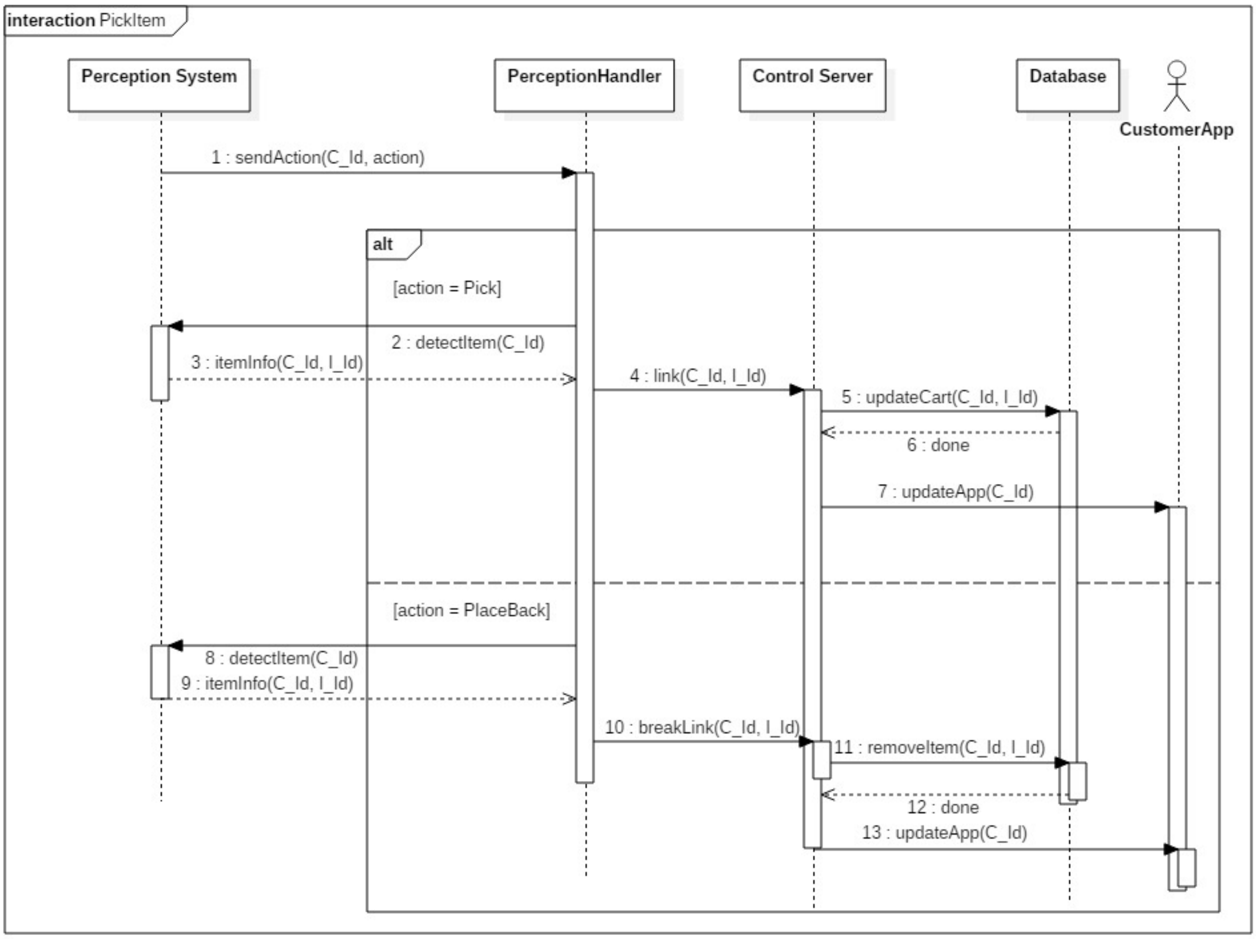

Figure 4.7: Sequence diagram showing the interface between Perception Handler and Control Server

Interface between Entrance Controller and Database:
Entrance Controller sends authentication credential to database in order to check whether customer is already in store. If customer is already in store, authentication system warned. In case authentication valid, Entrance Controller checks payment preferences for customer whether "shop with group" is selected or not. Moreover, if customer shops with a group, Entrance Controller links the shopping cart of the customer with master customer of group.

Design Rationale:

- Whenever customer is detected "already in", log of this failure is stored in database.

- Shopping carts link information is stored instead of merging tables since payment preferences may be changed during the shopping.

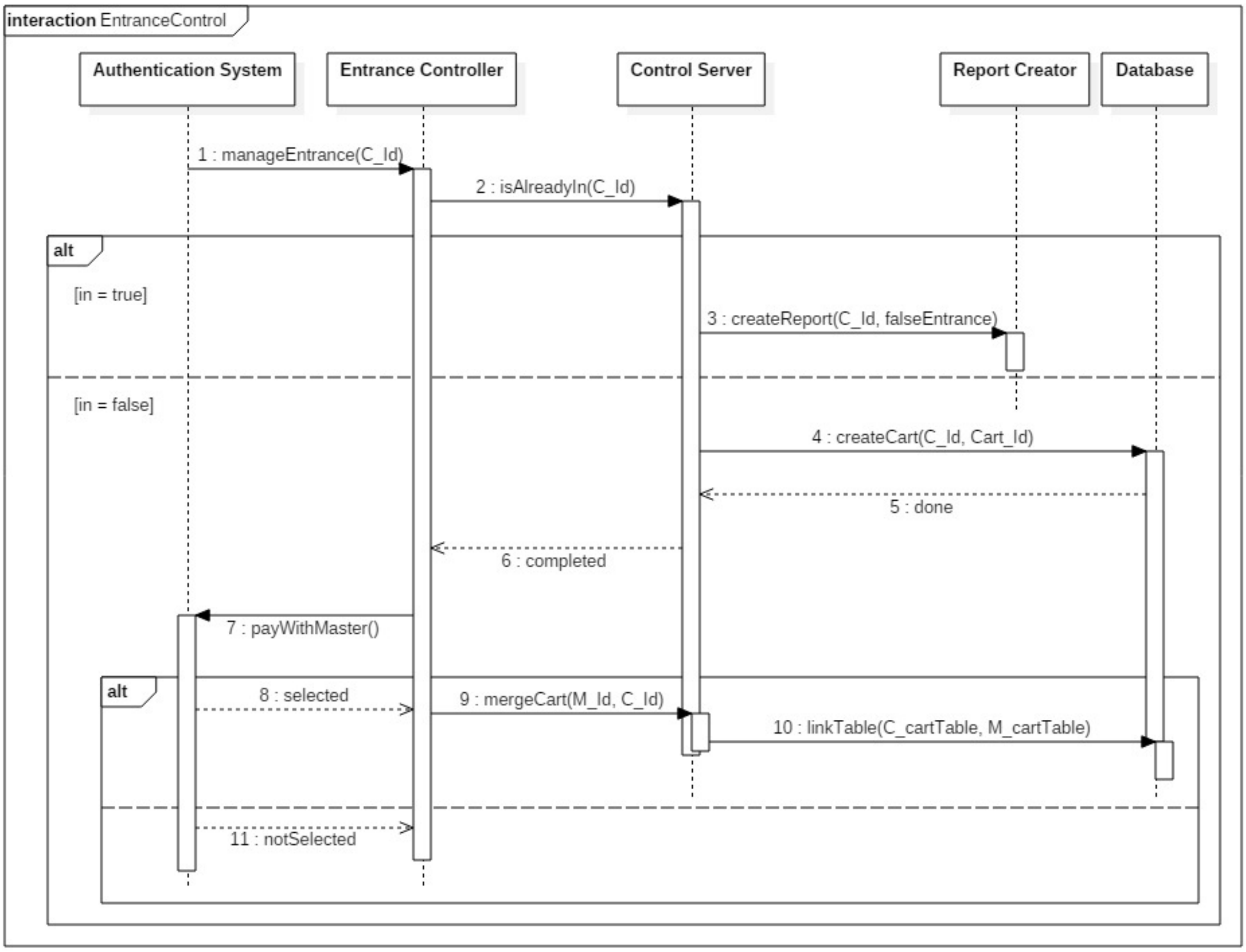

Figure 4.8: Sequence diagram showing the interface between Entrance Controller and Control Server

Interface between Invoice Server and Database:
Invoice Server is awaken with Perception System interrupt and has customer id from Perception System. By using customer id, Invoice Server retrieves data from database. Database returns shopping cart with respect to given customer and last entrance time. After Invoice Server handled payment, transaction information is stored in database.

Design Rationale:

- Invoice Service waits for an interrupt from database while it serves for other invoices. Whenever database prepares cart data, it sends interrupt to Invoice Service.
- In case any network error, Invoice Service stores transaction information for one day.

4.4.2 External Interfaces

User Interfaces

Amazon Go has interfaces for customers, store staffs, store managers and researchers. Their concerns and permissions varies depending on their position in the system. Detailed explanations are given for each user interface.

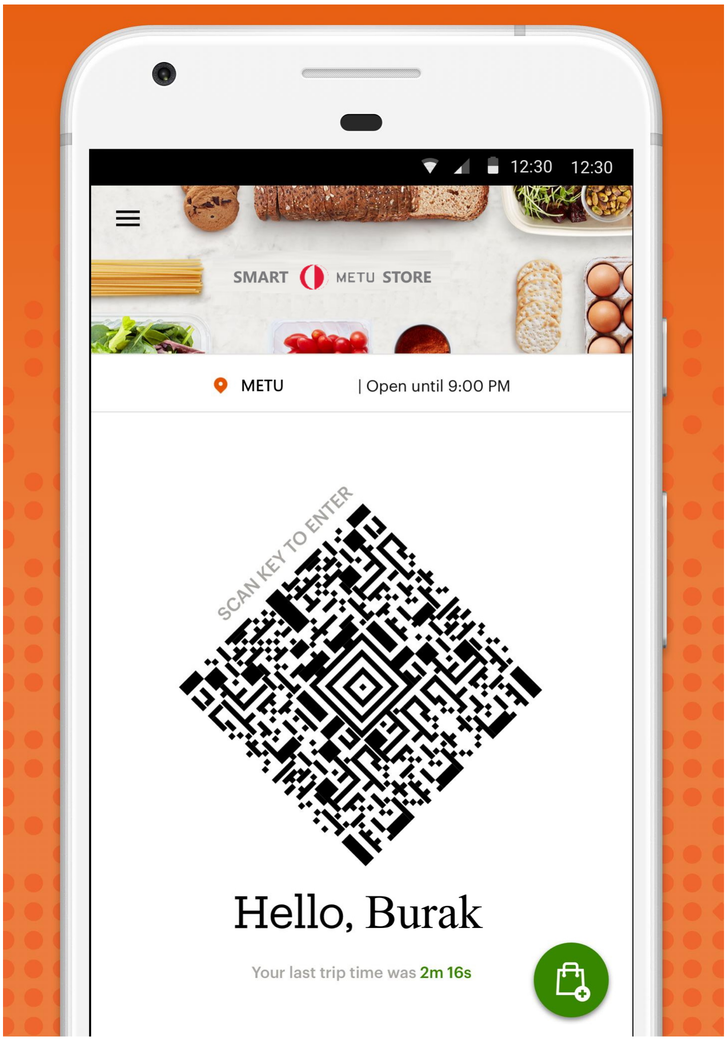

Figure 4.9: Amazon Go Mobile App

Customer
Customer interface creates a unique Qr code, which allows each customer to pass the turnstiles and successfully login to the system. With this interface customers can see their shopping cart during shopping. Also, they can see their shopping history. Besides, user interface allows customer to create or remove groups which will be used to merge shopping carts during invoice. At this point, only the group master can add or delete group members. Customers set the payment method by filling the credentials. They also can change the methods later. In addition, in this interface, each customer can rate employees after the shopping by giving them ratings and special comments.

Design Rationale:

- Each selected operation can be handled not exceeding 3 button operation.

- For security issues, mobile app should use 2-step verification for first login.
- Before changing payment options mobile app should request password.

Researcher
In the researcher interface, researchers can access Amazon Go system's data while the system is running concurrently. This interface contains three tabs; "Logs" tab, "Performance" tab and "Alternative Visualizations" tab.
In the logs tab, researcher can select to view estimation logs and sensor logs. Estimation logs contains action detection logs, pose estimation results with timestamps. Researchers can search for specific user, specific duration or for specific action. They can select data's here to use in the following performance and visualization tabs.
In the performance tab, there will be different models to train or pre-trained models to test data's. Researchers select the models to use with their selected data from the previous tab. After the selection, operation will be queued for training or evaluation. After the operation, results will be shown under this tab. Alternative visualization tab plots the selected data's in different views to gain insight information. Different graphs and methods are available.

Design Rationale:

- Interface should be compatible with many common browsers since it will use javascript for animations and representations.
- Before start training, researcher should be allowed to select the notification type when the training finished.
- Alternative visuals can be downloaded in different formats such as ".jpg, .png,"

Store Staff
Store staff interface allows the employees to interact with the Amazon Go system. Since there are many different work types for employees, there are multiple tabs in the interface. These tabs namely "Tasks", "Reports" and "Performance" can be accessed by the all staff.
In the tasks tab, employees can see the tasks assigned to them by store manager/managers, they can ask questions about tasks by clicking "Ask question" button after ticking the task.
In the reports tab, store staff can create and read the reports. In this tab, there are both automatically created sensor reports and manually created reports. While creating reports, report priority can be selected. In the performance tab, store staff can see their ratings which are collected from customers.

Design Rationale:

- High priority reports cannot be deleted unless it is marked as "Solved".
- Rating for each employee should be anonymous because of privacy policy.
- Each user must see only his/her rating and reports.

Manager
Manager interface allows many different operation on the database. It also contains the functions provided to staff members. It has 4 different pages; "Info", "Performance", "Trends", "Reports".
In the info page, manager/managers can see stock info and employee info. They can use this to promote some employees.(e.g. change salary option). Besides, stock operations (buying new item for store) can be handled from the "Order" section under this part.
In the performance page, there are weekly/monthly/yearly performances of employees measured using ratings

from the customers. Manager can send a direct message to the employee about his/her performance using the "Send performance message" button.
In the trends page, customer's buying habits are tracked and shown in weekly,monthly,yearly scale. In the reports page, manager can check the status of reports, reassign tasks according to reports.

Design Rationale:

- Manager interface shouldn't provide "Save password" option.
- Researcher's selected data for training shouldn't be affected after the changes done by manager.

System Interfaces

Amazon Go has external systems which are done critical tasks for Controller Server. In this subsection, detailed explanation of interfaces for external systems which is designed for communication is explained.

Interface between Invoice Service and Payment System
Payment System arranges payment options. For instance, if customer prefers online payment, Payment System communicates with system of preferred bank. Payment System gather payment preference and if necessary bank information from Invoice Server. Invoice Service also transmits customer information, amount and time of shopping cart.

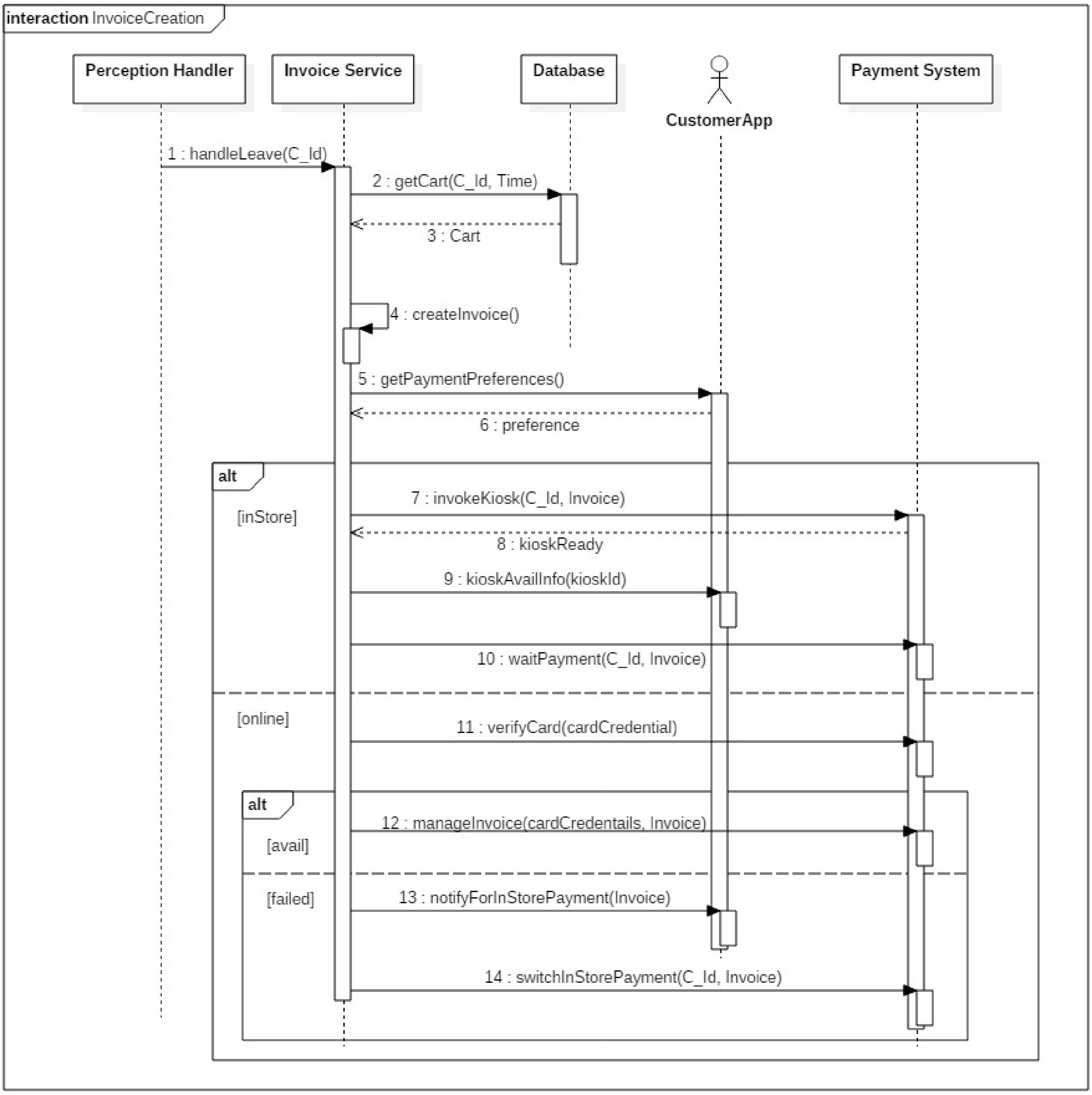
interaction InvoiceCreation
Perception Handler
Invoice Service
Database
CustomerApp
Payment System
1 : handleLeave(C_Id)
2 : getCart(C_Id, Time)
3 : Cart
4 : createInvoice()
5 : getPaymentPreferences()
6 : preference
alt
[inStore]
7 : invokeKiosk(C_Id, Invoice)
8 : kioskReady
9 : kioskAvailInfo(kioskId)
10 : waitPayment(C_Id, Invoice)
[online]
11 : verifyCard(cardCredential)
alt
[avail]
12 : manageInvoice(cardCredentails, Invoice)
[failed]
13 : notifyForInStorePayment(Invoice)
14 : switchInStorePayment(C_Id, Invoice)

Figure 4.10: Sequence diagram showing the interface between Invoice Service and Payment System

Design Rationale:

- Invoice Service does not store more than credit cart number since it is legally banned. Thus, cart info can not passed to Payment Store. It is gathered by customer.
- Customer information includes customer id and credit cart number.

Interface between Perception Handler and Perception System
Control Server gathers perception data from Perception Handler. Perception Handler waits for a perception from Perception System. When Perception System sends a perception to Perception Handler, it collects data and decodes. After decoding, it awakes appropriate Control Server action.

Control Server also sends perception request via Perception Handler. Perception Handler creates a request for Perception System and waits for a return.
Perception Handler is responsible for decoding various tasks. Detection of a human or item requires handling, i.e gathering necessary information from database and afterwards informing Control Server. Action estimations are related with human detection. Also, Perception Handler checks if estimated item coordinate is correct by gathering item data from database. If item is on wrong shelf, it warns Control System to create a report for SO System.

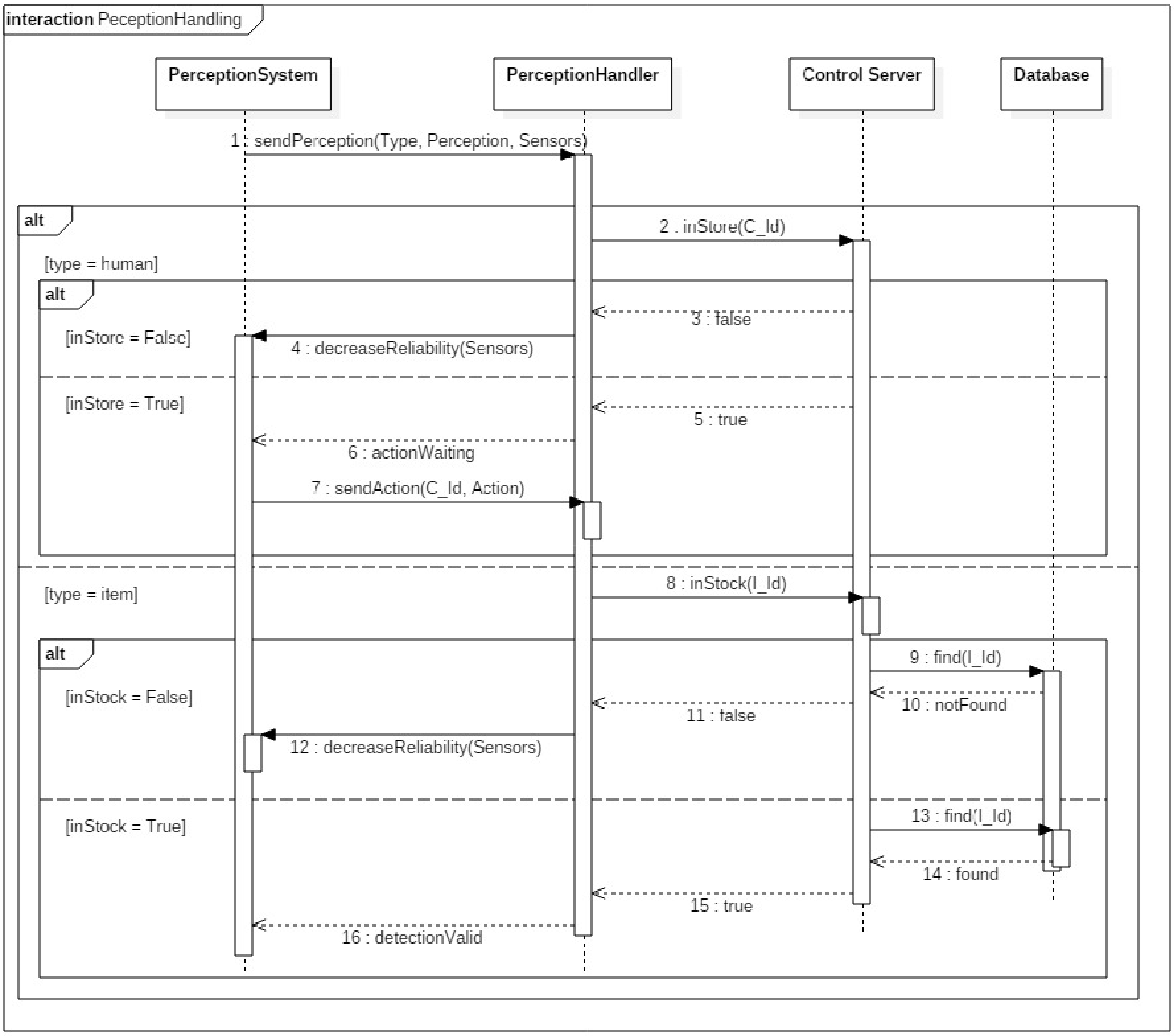

Figure 4.11: Sequence diagram showing the interface between Perception Handler and Perception System

Design Rationale:

- Perception Handler requires special structures same with Perception System in order to acquire data from Perception System. Image with estimation, Image with estimated action and Human are some of data structures.
- This interface has high band-with ratios since transferring data is bigger than other external systems.

- Perception Handler and Perception System connected with bus type connection. As Perception System consists of various systems ranging from estimation models to sensor fusion systems.

Interface between Event Controller and Management System
Store Managers can schedule events instead of making them operative immediately. Management System creates events with event duration in terms of start and end date of event. Event description can be sent to store in database, which helps to other Store Managers to understand event. Management System can assign type for an event in order to describe it's priority. Yet another effect of type assignment is determining action when scheduled event fails when it's time is come, it fails during execution or end time is passed but event is not ended. Event Controller also informs Store Manager for collided events.

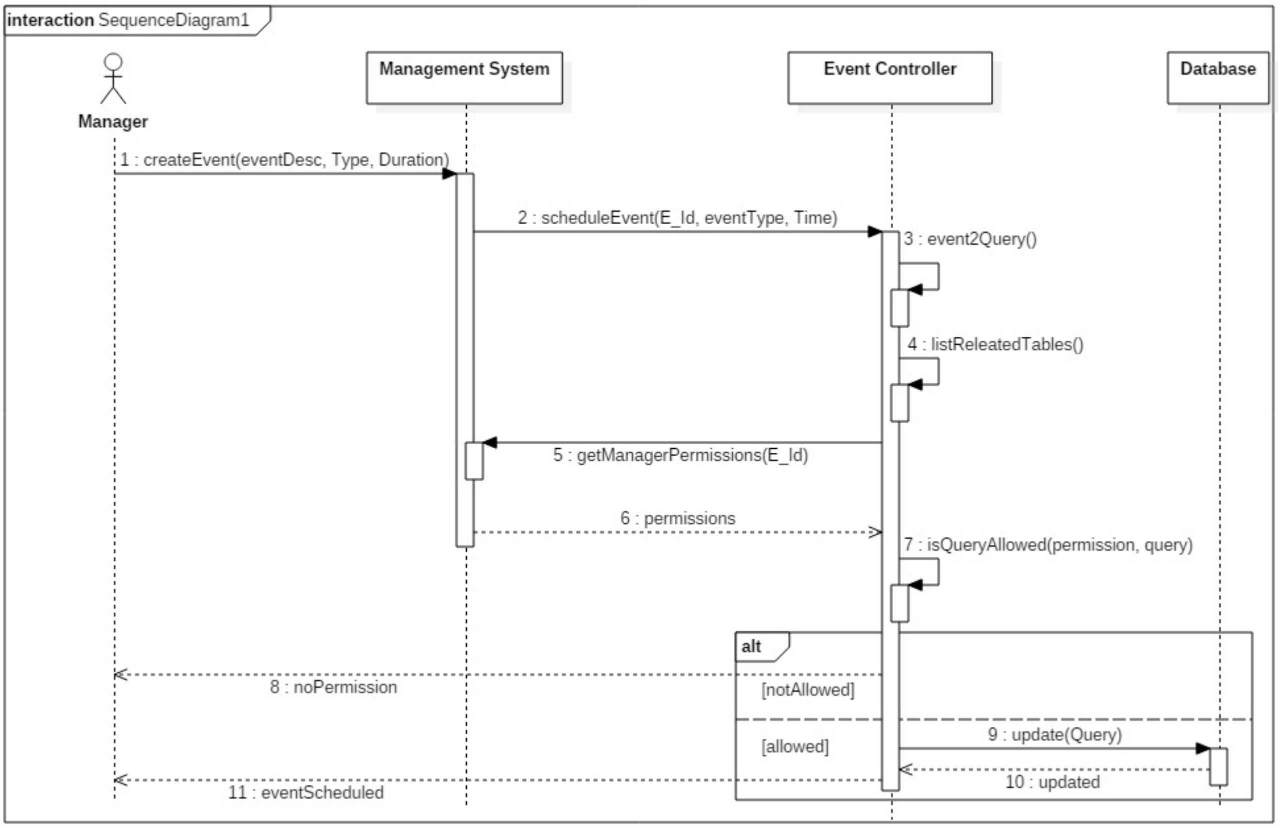

Figure 4.12: Sequence diagram showing the interface between Event Controller and Management System

Design Rationale:

- Event Controller provides list of event types with description in order to guide Store Managers.
- Dates are checked before event created in order to detect any invalid time interval is decided.
- Event Controller sends warning that includes collided events with selected time duration before event is created.

Interface between Report Creator and Sensor Maintenance System
Sensor Maintenance System provides interface for technical staff to monitor assigned task for them and overall non-functional sensor in the store. Report Creator stores these information as reports such that employee can monitor them via their interface with Control Server and at the same time they are stored in database.

Design Rationale:

- Report can be created for a sensor or a group of sensor. Two different report template should be presented for Sensor Maintenance System.
- Report can include a technical staff that is assigned by Sensor Maintenance System.

Interface between Report Creator and SO System
SO System is related with SO staff. SO staff views reports via their interface. Therefore, SO System creates reports to store them in database. It can use interface by proactively creating report or Control Server may ask for shelf check and report is created as a result of this request.

Design Rationale:

- Since various creator can trigger SO System, creator of report must be included in report.
- Report can include a SO staff that is assigned by SO System.