Almost Games

Functional Specification and Management Plan

Date:04/02/2016

Editors:
Tal Melamed
Brandon Mabey
Hadong Tao
Tania Akter

Table of Contents

Revision History

Feb 16, 2016. Version 1.0. Document creation, functional, user interaction, and management content.
March 3, 2016. Version 1.1. Incorporating customer comments, adding UML diagrams, and expanding system description.

Executive Summary

The purpose of this document is to describe the functional specifications of Servr, a system that handles the transport of full and empty carts to and from shoppers in various brick and mortar stores. The systems aims to make the shopping system more efficient by allowing users to send the chosen goods to the checkout stations while shopping for other goods thereby increasing store profits and customer satisfaction.

Servr delivers empty carts to users which make a request, taking their full carts to the checkout station. To do this, it must pinpoint the customer's location relative to the cart and keep track of obstacles in the store for which the cart must avoid. A mobile app will be the customer's main point of interaction for the system, allowing them to call and dismiss carts, provided that the pre-conditions set out in this document hold. The application will communicate with a central control station located within the premises which manages both the carts and users endpoints.

In addition, the minimum system to be implemented is specified, including possible implementations. Possible additions to the system if development progresses at a faster than expected pace are also defined in this document.

Functional Specification

Project Summary

The “Servr”shopping assistance system is a robot management and deployment system coordinating assistance to department store shoppers. Shoppers with a cart filled to maximum capacity can issue a request to the system which will dispatch a robot to bring the shopper an empty cart. The shopper, receiving the empty cart, can send the old cart to the checkout station and continue shopping for more items.

Needs and Objectives

The following needs and objectives are considered for the purposes of this document:

Needs

- Deliver an empty cart to the user’s location promptly after a request.
- Deliver the user’s cart to the checkout station.
- Avoid collision with obstacles and other objects in the store.
- Scan location identifiers within the store

Objectives

- Allow customers to purchase more goods than a single cart can carry.
- Improve checkout speed and reduce checkout lines.
- Increase store profits.
- Increase customer satisfaction and loyalty.

Significant Features

Customer request and dispatch interface

The shopping centre is populated with position information nodes at regular intervals. Users can request an empty shopping cart at these nodes by triggering an interaction between the nodes and the user’s mobile smartphone running the system’s mobile app. Triggering the interaction causes the central control system to dispatch an empty cart to the node’s location.

Each shopping cart is equipped with a module that the user can interact with to link the cart to their ownership and send the cart to checkout. Upon receiving an activation from the user, the module is assigned a unique identifier from the user. This identifier is used by the user to claim the goods later at checkout.

Robot shopping cart pathfinding system

The shopping cart robots drive along predefined paths as instructed by the pathfinding algorithm in the central control system. The central control system has a predefined library of paths to and from any position. Upon receiving a dispatch request, after processing all current and pending traffic, the system selects a viable path preventing collisions with other robots and assigns it to the requesting robot. The robot will follow the path it is assigned. Collision with shoppers and other objects is prevented by detecting obstructing objects with a rangefinder that every shopping cart is equipped with.

Robot shopping cart maintenance system

A storage area is outfitted with robot charging stations. Robots will reside in these stations when not in use to ensure full charge ready status

Central Control

The central control system is responsible for all decision making and robot path generation. The control system coordinates shopping cart request and dispatch orders. It stores the data of all shopping cart locations and customer information such as checked total and owned carts. The system runs as an application on an in-store computer connected to the internet.

Hardware and Performance

Servr is expected to utilize smart mobile phones as the main interface for customer interaction. The mobile application will wholly contain the user interface for all customer interactions. In addition, the response time of the system progress is under 0.1s to give the impression that it is reacting instantaneously. For actions that cannot be completed within 0.1s, a loading bar or spin circle will be shown within that time range to ensure that the user understands that system is working as expected.

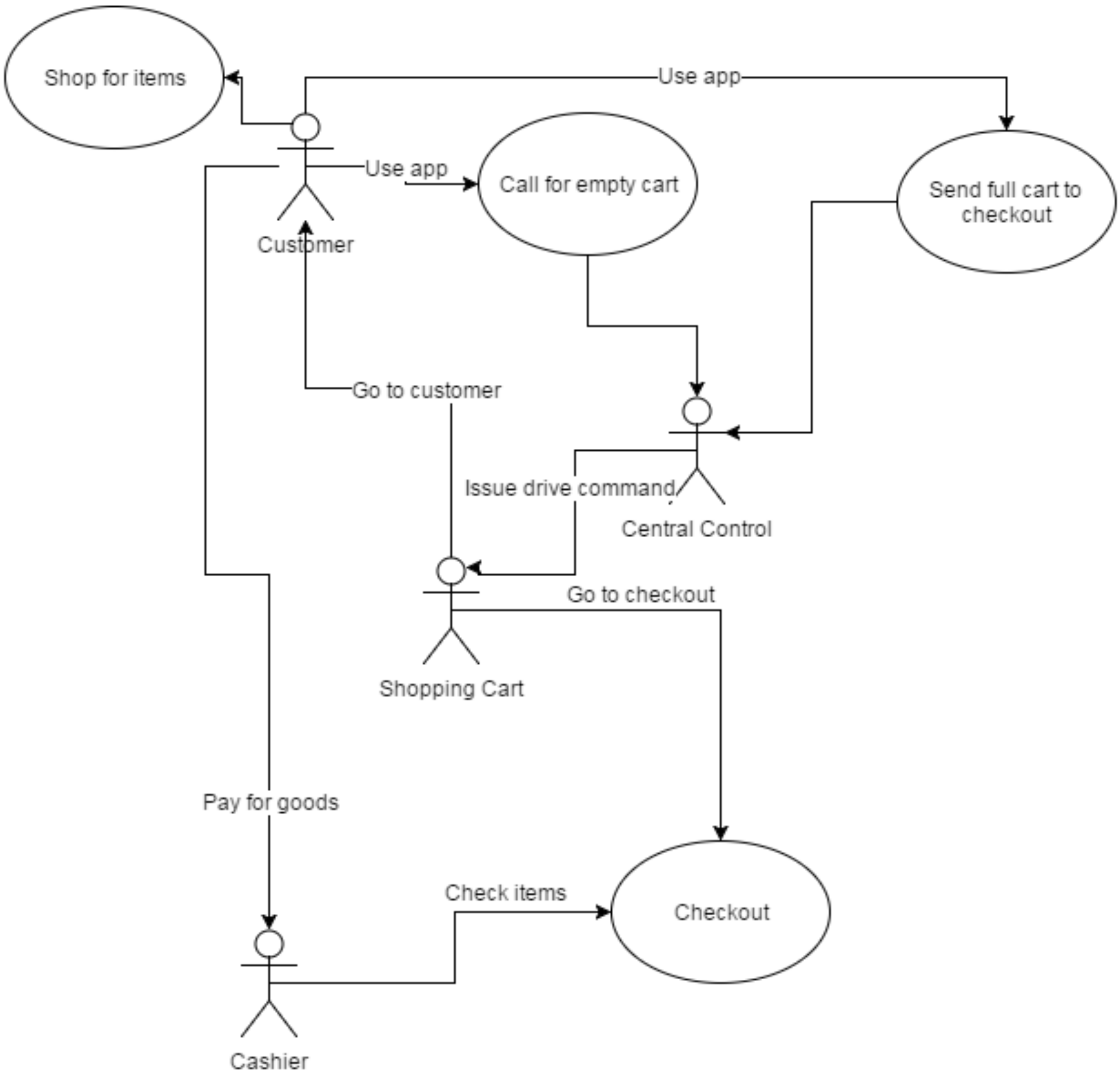
Stores will be required to have an army of robot shopping carts ready for usage. The robot carts will have driving motors and mechanisms capable of driving a predefined path through the store. Each robot must be equipped with rangefinder sensors and collision prevention software. Robots must be electric powered with a battery lasting for 30 minutes of usage. Stores require charging stations for the robots.

Beacons are positioned at regular intervals throughout the store for position information to the system, with at least 1 beacon per aisle.

The cart can not collide into other objects. The cart must arrive to the requesting customer within 1 minute of the issued request.

User Interaction

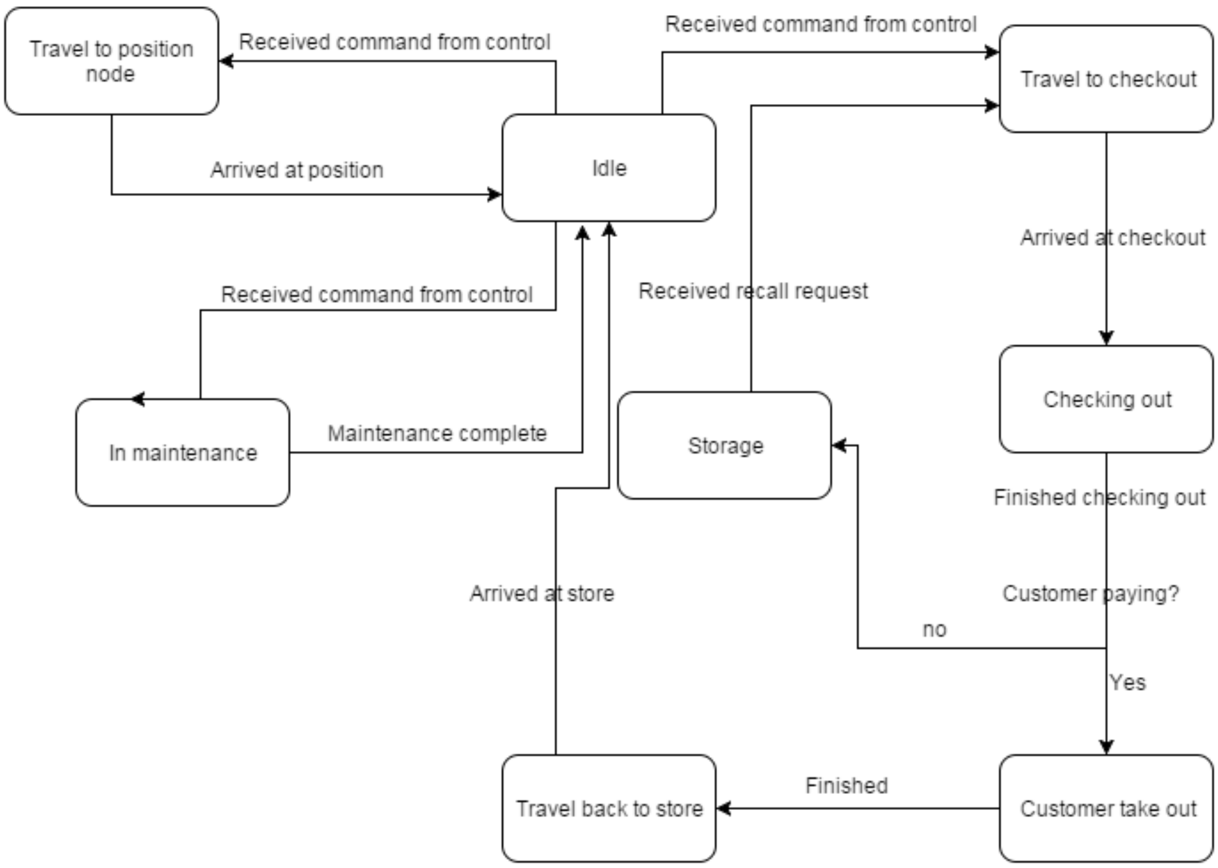
The application involves interacting with a mobile application and with the Servr shopping cart itself. To use the system, the customer must have the app downloaded.



The customer uses the mobile application to request for a new empty cart to their location or to send their existing full cart to checkout. The system determines the location of the customer and an available self driving cart is dispatched to them. The user marks the existing full cart with unique identification and the cart leaves for the checkout station and waits till the customer is done with shopping. When the customer is ready to checkout, they can use their unique identifier to bring up their checked total as well as recall the cart full of their goods.

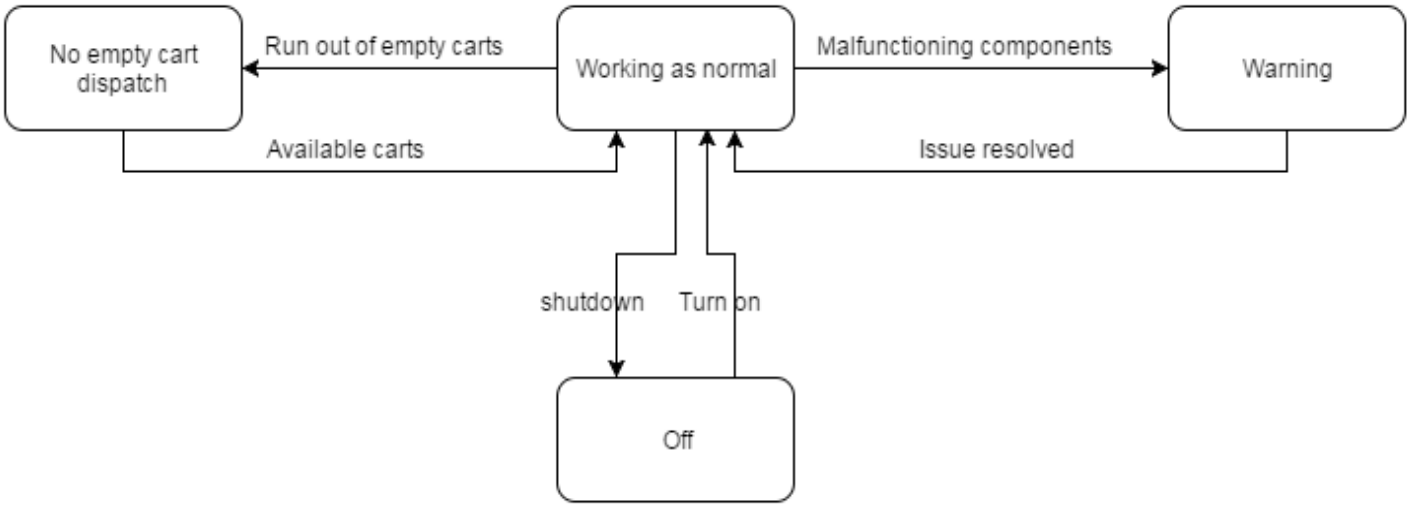
System Functions

Cart state Diagram



The cart can be in a number of different states. The states are changed in response to commands received and context such as current state. Idle is the state the cart is most often in. Idle state allows for actionable responses to commands received from the control center. A cart will experience all the states multiple times during its typical lifecycle.

Control State Diagram



The control system features three working states and one off state. The control system is typically in the “working as normal” state for most of the time. “Warning” and “no empty cart dispatch” are states of concern indicating a deficiency in the running system, these issues should be promptly resolved to avoid further inefficiencies or damages. The system will attempt to continue to work during states of distress.

Initial Configuration

Initial configuration is performed by store management. Management sends a floor plan of the store including aisle placement and other major structures to Almost Games. Almost Games then sends back customized software, shopping cart robots, and position nodes with instructions on node placement. Management then follows the instructions to place nodes then activates the system using the central controller.

App

Customers intending to use the system must download the mobile app to their smartphones from the app store. Upon opening the app for the first time, some personal information is requested, including name and date of birth. The app has a single button named “activate scanner”. Pressing this button activates the module that interacts with the position nodes and shopping carts. Positioning the phone near a position node initiates a call for an empty cart to the customer's location. Positioning the phone near a shopping cart makes that shopping cart link to the users unique mobile token and sends the old cart to checkout. Both actions feature prompts indicating the action to be performed with an “are you sure?” message that the user can confirm on. When the customer arrives at checkout, the scanner is once again activated and interacted with the checkout machine to bring their tagged carts to their location for item to be picked up.

Shopping Cart Robots

The shopping cart robots contain a scanner module and a status light near the handle. The scanner module is interacted with a mobile phone running the application. The status light is a small LED bulb. The light blinks a steady green for 5 seconds if the cart interaction is successful. The light blinks an intermittent red at a frequency of 15hz if an interaction was detected but unsuccessful. The Status light is off if no interaction is detected.

Central Control

The central control system runs as a desktop application on an in-store computer. Users can start and stop the system and see current status such as in-traffic carts and currently checked items. Analytics data is made available to the user, such as number of requests made, peak hours, popular areas for requests, etc.

Checkout

Checkout features a scanner module and a cashier. Customers interact with the scanner module using the mobile app function and placing the smartphone near it, this triggers the checkout to receive the customer's checked total and prompt for shopping cart delivery. The customer pays the checked total with usual methods, including either cash, debit or credit. The shopping cart delivery prompt contains a list of the customers checked carts, they can request to have all their carts delivered to their location or to have one at time delivered.

Sample Interactions

An intuitive way to envision how the system operates is to think of the system as if it were another person whom the user was talking to. Under this assumption, the following are sample dialogs which explain the overall operation of the system:

User: Please send a cart. Here is my location.

System: Please wait. A cart has been dispatched to your location.

System: A cart has arrived at your location. Please mark the cart as ready for future checkout by swiping your phone.

User: I have swiped my phone.

System: Thank you. Your items will be ready for you at checkout.

If the system detects that no empty carts are available, the dialogue would instead go something like the following:

User: Please send a cart. Here is my location.

System: Please wait. A cart has been dispatched to your location

System: It seems that all of the available shopping carts are in use. Would you like to be notified when there is one available.

User: Yes

System: Okay, I will notify you when a shopping cart is available for usage.

At a point later in time, the dialogue would continue:

System: A shopping cart is now available. Would you still like an empty shopping cart?

User: Yes, here is my location..

System: Please wait. A cart has been dispatched to your location

System: A cart has arrived at your location. Please mark the cart as ready for future checkout by swiping your phone.

User: I have swiped my phone.

System: Thank you. Your items will be ready for you at checkout.

The user may change their mind at any point as well. For instance, if the cart arrives, and the user decides to not consent purchase, the dialogue may change into the following:

User: Please send a cart. Here is my location.

System: Please wait. A cart has been dispatched to your location

System: A cart has arrived at your location. Please mark the cart as ready for future checkout by swiping your phone.

User: I have swiped my phone.

System: Thank you. Your items will be ready for you at checkout.

When the user is done shopping, they will be asked by the cashier if they had any additional carts. If they have had previous carts, they will be added to the total cost, and returned to the customer.

Cashier: Hi, thank your purchase. Did you have any other carts during your shopping experience with us?

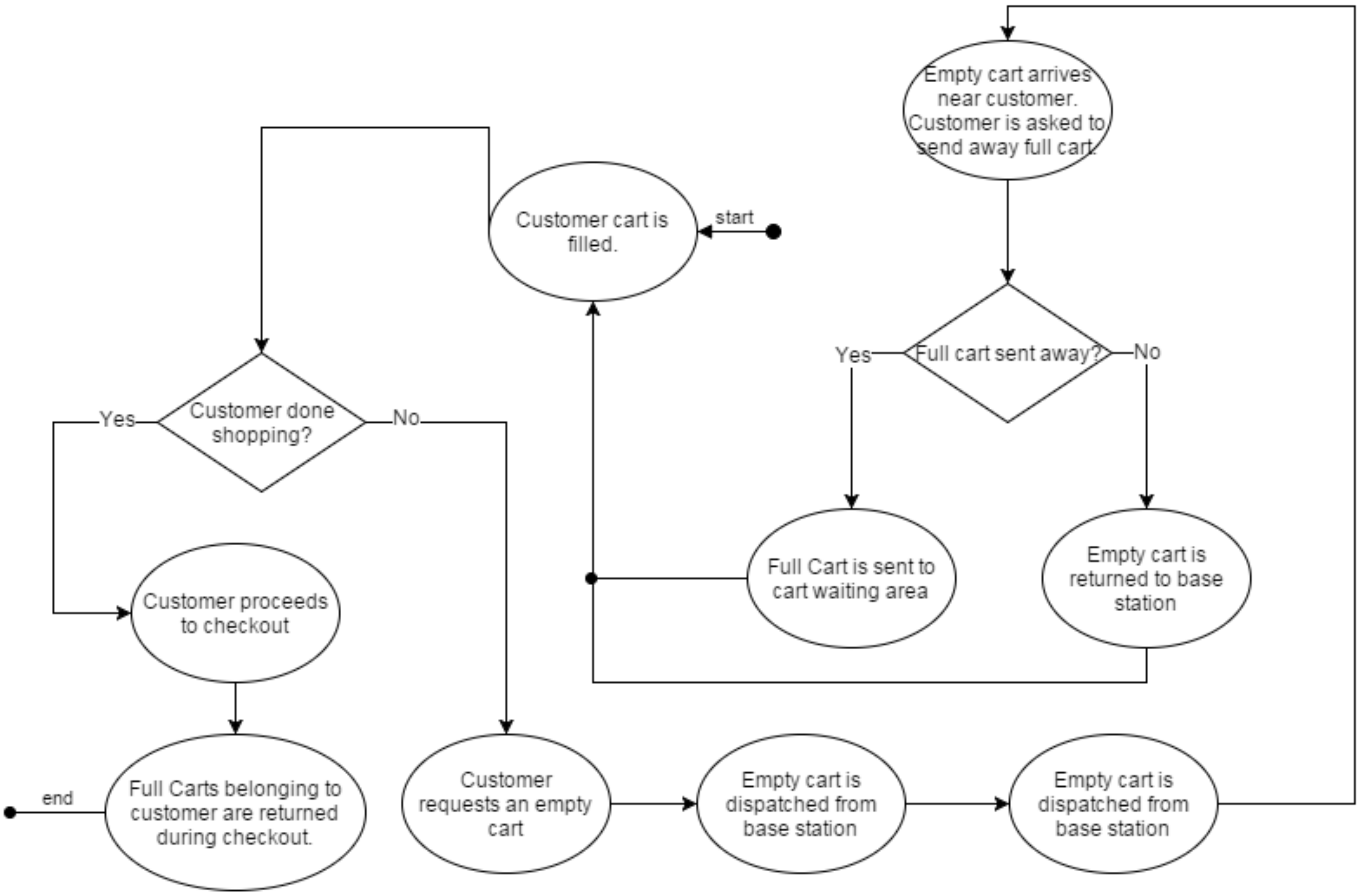
User: Yes, here is my phone

Cashier: Okay, I've let the system know that the carts should be returned to you, and added their total to your final price. How would you like to pay?

From that point on, the dialogue would remain the same as it is in grocery stores today, except that the full carts that the user sent away would be returned to the user.

Overview of the System:

Activity Diagram



The activity diagram above shows the expected flow that customers will take during their interaction with the system. Because it is expected that all users will enter and exit the store at some point, there are entry and exit points marked "start" and "end" that designate when the user enters or exits the store. In addition, the pictorial representation of the expected flow for the customer does not take into account potential errors that the system occurs, and assumes the "happy path".

The diagram is read by following the arrows to each bubble or diamond. Bubbles represent events that are expected occur, with the arrow leading out of the bubble to be followed after the event described by the bubble occurs. Diamonds represent different choices or possible events that may happen, and are read by following the arrow that exits the diamond that corresponds to the choice made.

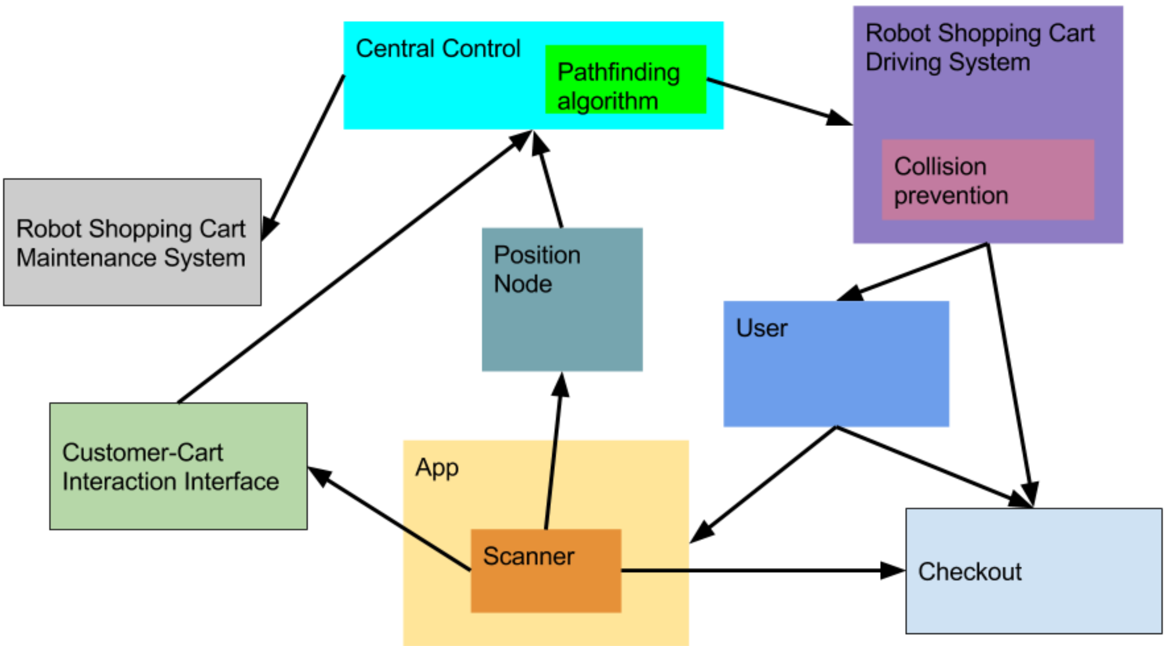
Context Awareness

The shopping cart robots can detect the collisions from all directions. The robot can detect any obstacles including the shoppers, the default isles, other robots nearby, and the speed bumps the stores put up in the the passageways occasionally. The system decides if the robot should go forward or not.

The central control system is aware of all in-traffic carts and all delivery and dispatch requests, the driving path of the robots is determined using this information.

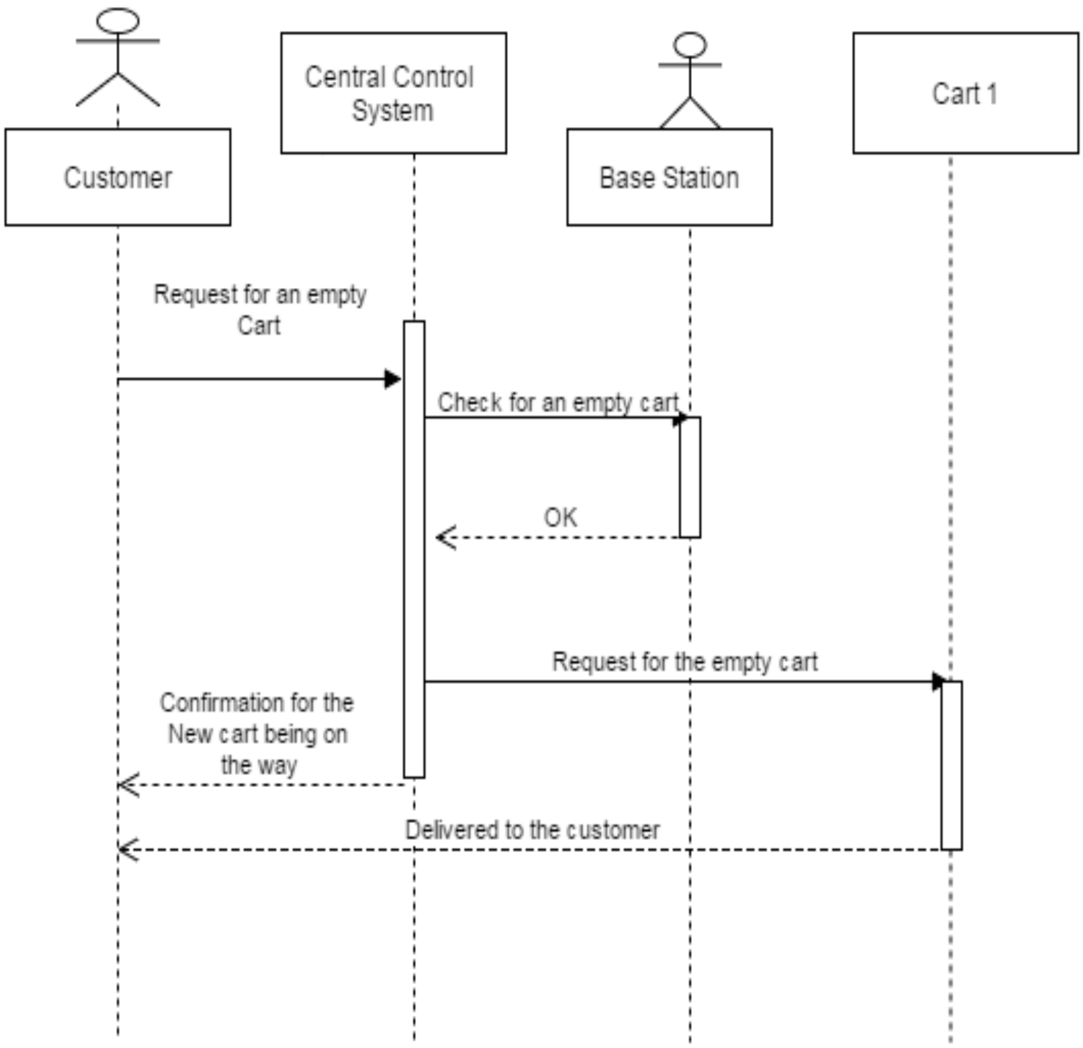
Management Plan

Function Classes and Relationships



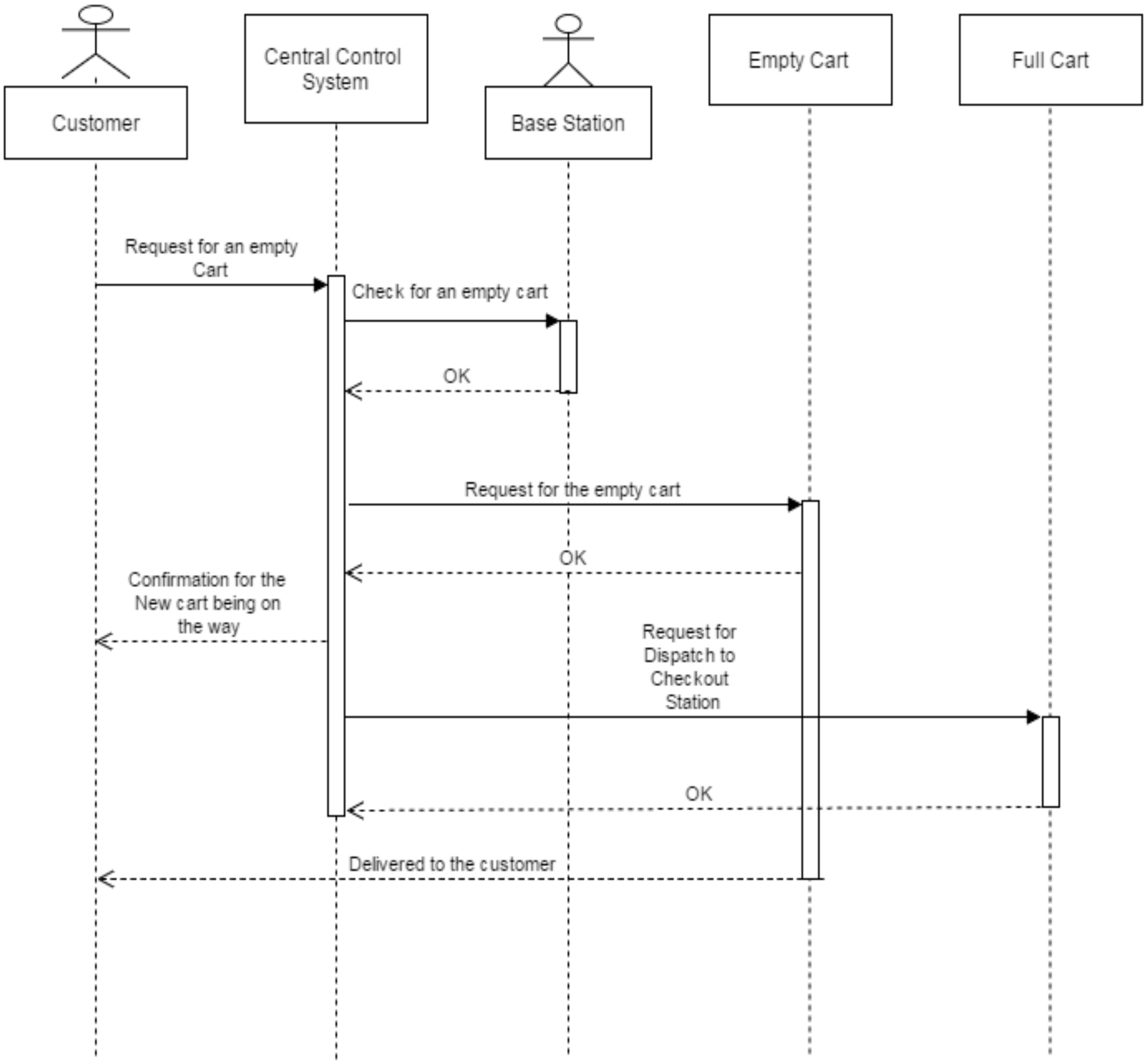
This is a rough function class relationship diagram. It describes how we would like to organize the system.

Sequence Diagram for an empty cart request



The customer requests for an empty cart from a particular location. The Central Control System gets the requests and checks if there are any empty cart in the base station. The base station returns an OK signal. The Central Control System requests an empty cart, and confirms the customer about the empty cart being on the way. In the meantime. the requested empty cart goes to the customer's location.

Sequence diagram for switching carts



The customer requests for an empty cart from a particular location. The Central Control System gets the requests and checks if there are any empty cart in the base station. The base station returns an OK signal. The Central Control System requests an empty cart, the empty cart sends an acknowledgement ; the Central Control System confirms the customer about the empty cart being on the way. The Central control requests the full cart to dispatch to the checkout station for the items to be checked out. The full cart sends a confirmation and heads to the checkout station. The empty cart arrives to the customer.

App

The mobile app is the primary interface for the user. The app receives input from the user to interact with the system.

Scanner

Within the mobile environment resides a scanner module. The scanner is activated by the user and physically placed near a receiving scanner node to trigger one of the three following events:

- Customer-cart interaction interface: this receiving node is on every shopping cart robot, when activated it initiates the command for the cart to go to checkout.
- Position node: this receiving node is placed at regular intervals in the store, when activated it initiates the command for an empty cart to go to the triggered node location.
- Checkout: this receiving node is present at checkout locations, when activated the customer's checked total is loaded and their full carts are prompted to arrive at the checkout station.

User

The user is interacting with the system using the mobile app. The users make decisions regarding when to dispatch and call for carts. Users have the option to shop without using the system.

Customer-Cart Interaction Interface

A module on each shopping cart receives scanner activation from a nearby smartphone running the system mobile app. Upon receiving the activation, the module receives a unique identifier from the customer and detects its location in the store. It sends the data it has acquired to the central control system and awaits direction for a path leading to checkout.

Position Node

Nodes located at regular intervals throughout the store, are activated with the scanner from a nearby smartphone. Upon receiving the activation a data package containing the nodes location in the store is sent to central control.

Robot Shopping Cart Maintenance System

The maintenance system is a storage facility that charges robot's batteries and performs other maintenance operations. Statuses, such as charge level, about robots currently in the facility is regularly sent to the central control system. The central control system directs robots in and out of the facility.

Central Control

Central control is a software system running on a computer in the store, it is initiated and configured by a store employee. The system coordinates events and processes within itself using received input and state information. Multiple modules directly interact with central control:

- Robot shopping cart maintenance system: send data to central control about the status of robots in maintenance. Central control decides when robots are sent and retrieved from maintenance based on contextual data about the store environment and system status.
- Customer Cart Interaction Interface: send data to central control when activated.
- Position Node: send data to central control when activated.

Pathfinding Algorithm
The pathfinding algorithm is a module inside central control, initiated when central control has received an activation signal from either a position node or a shopping cart robot. The algorithm determines a path that the requesting cart can take to safely reach its destination using contextual store and system status data. When the path is rendered it is downloaded to the requesting cart (if a position node is activated and an empty cart has been requested). After downloading the path data, the cart's driving system is activated.

Robot Shopping Cart Driving System

The driving system downloads directions and activation from central control. Once activated the driving system controls its motors and steering mechanism to drive the directions it has received.

Collision Prevention
Within the robot shopping cart driving system is a collision prevention module, it scan the surroundings of the shopping cart in real time detecting obstacles. If an obstacle is detected, the collision prevention system issues an emergency stop and requests new routes for the cart(s) to be generated to resolve the collision.

Checkout

The checkout is located near the exit of the store. Users must be physically present at the checkout station to interact with it. A scanner receiving node is present at each checkout station that the user can activate with the mobile app causing the checkout to load their checked total and owned carts data from central control. Data about the checkout event is sent to central control. Users pay for their goods using traditional methods.

Possible Implementations

Customer Features:

Customer-Cart Interaction Interface

Function 1: Interface needs to have a button customers can use to call carts to their location. There are two possible implementations. First, design a software button on the mobile app and the system will triangulate the user's location using a NFC communication system and send a cart to that location. The second method, we can place physical buttons around appointed locations in the store, and those locations would act as cart pick up locations.

Function 2: The carts can be claimed by customers and once claimed, the carts will remember their claimant's identity until discharged. Two possible implementations of this function: First, we can place QR code on the shopping cart. To claim the cart, the shopper scans the QR code with the camera on their smartphones, which syncs the cart to the mobile app. Second method is to build nfc chip on the handles of the shopping cart and the user can tap connect their phone to the cart.

Function 3: The customer can send the cart to the cashier for checkout. This feature can be implemented with a physical button on the cart, or a software button on the app.

Function 4: The user can un-claim the cart. This feature can be implemented with an unclaim button, a timeout system, or a proximity sensor, so if the customer is too far away from the store, the cart will become unclaimed. Once unclaimed, if the cart is empty, it will be set into a free roaming state, if there are items in the cart, it will be sent to an area to unload.

Robot shopping Cart Locating System

Function 1: Mobile app must have a compass that points the user to the location of the cart. This can be implemented with a graphical pointer on the app.

Function 2: When a customer owns multiple carts, each cart would have its own uniquely colored pointer.

Management Features:

Robot Shopping Cart Locating System

Function 1: Management system needs to determine the location of all carts in real time. One possible implementation is to triangulate each cart's location using NFC, and that information would be relayed to the central control management system.

Function 2: Carts need to be able to triangulate their own location. We can use NFC here as well.

Robot Shopping Cart Central Control

Function 1: Manage roaming free cart. One possible way to differentiate free cart and cart in use is with a owner tag in the program, so when a user claims a cart the ownership tag changes to "in use." When the cart is in storage, the owner tag changes to "waiting." If the cart does not have the "in use" or "waiting" tags, the cart is in free roam mode.

Function 2: Free roam carts must default to not in use or "waiting" when no users call for it. To implement this function, we can program all carts in free roam move toward the cart storage area.

Function 3: Carts in the "not in use" state will be deployed when there is no available "free roam" carts.

Function 4: When "in use," the customer should have full control of the cart. To implement this, we can build the interface app to communicate with Central Control, and Central control will relay the command to the cart. Or we can have a mini control center built into the app, and each mobile phone app would act as a temporary control independent of Central Control.

Function 5: The cart is not allowed to leave the perimeter of the shop. We can have a lock system where if a cart is too far away from the shop, the wheels would lock up and no longer able to move.

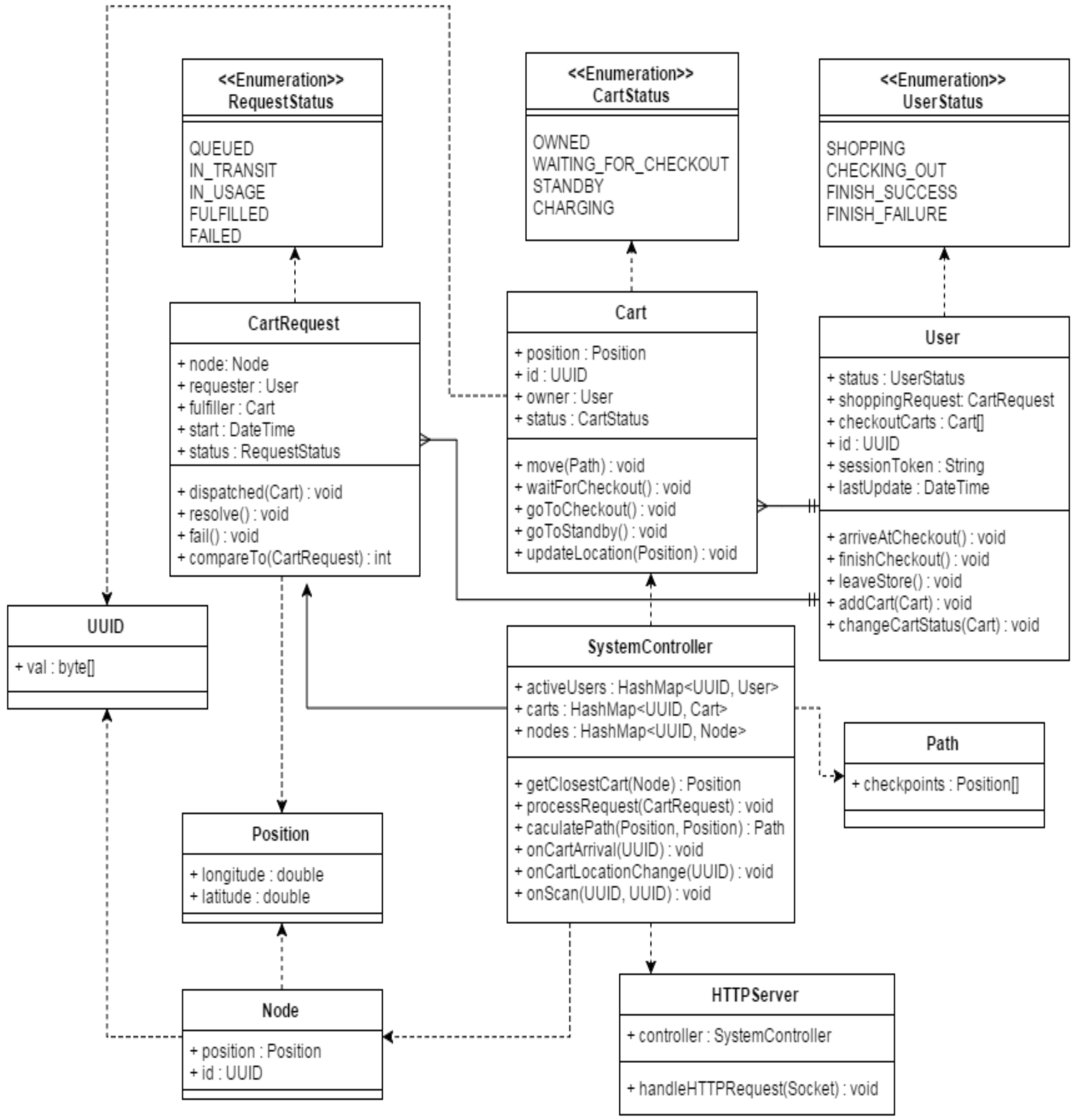
Robot Shopping Cart Pathfinding

Function 1: Application must be able to set lane maps for the cart to move on. One possible implementation is to have magnetic strips built into the floor, so the carts would snap to the magnetic grid when moving through the store. Another way is to have a digital map, and using NFC to locate and orient the cart into the direction they need to go based on the digital lane map.

Function 2: Shopping carts must be able to detect obstacles and avoid them. There are multiple ways to implement this features as well. One way is to have an ultrasound sensor for detecting obstacles, the other way is we can use a kinect type of webcam sensor. For obstacle avoidance, if the carts are on magnetic strips, we can simply stop the cart and restart the cart once obstacle is cleared. For free map, we can orient the cart and move around the obstacle.

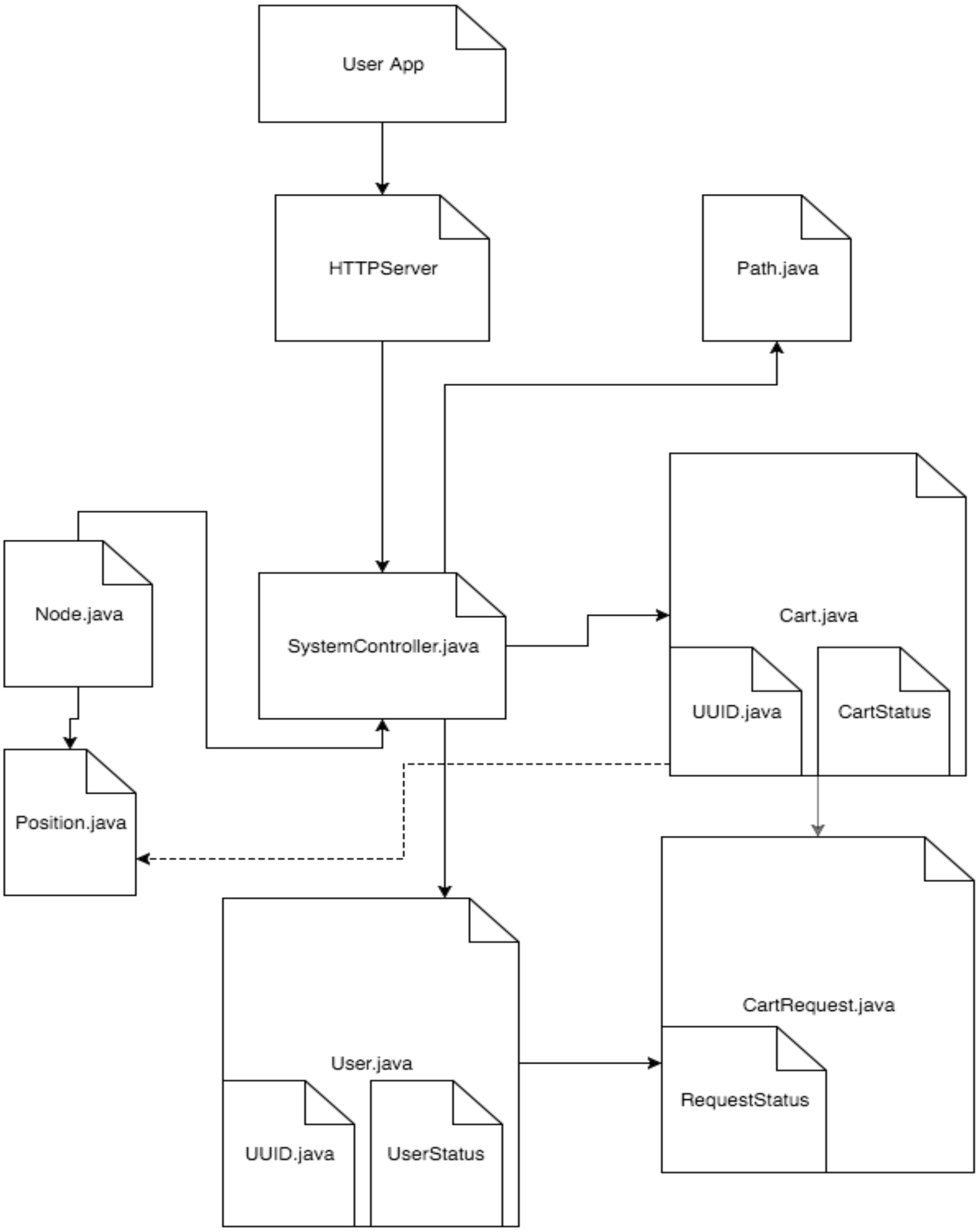
Function 3: Carts need to orient themselves. With magnetic strips built beneath the floor, the carts will automatically snap into place, so we don't need algorithm for cart orientation. We can also add two sensors on the the cart, one on the front one on the tail, then we can determine the orientation based on readings from both sensors.

Class Diagram



This is the class diagram. It outlines one possible way to organize our code. SystemController is the control Server for our application, and the user application communicates with the SystemController through the HTTPServer, where it sends requests and cart position information. All other objects such as User, CartRequest, and Cart are handled through the SystemController. Position and Node determines location of carts, and Path is calculated by the server and sent to the Carts.

Component Diagram



The above diagram describes the expected source files to be created for the final system. File displayed within other files correlate to the class diagram to show classes included on the class diagram that will not be separate source files in the final program. In addition, the phone will have a scanner which has a single function of making an HTTP request describing its scan to the store's base station. The cart, which is not in the scope of this project, is expected to make HTTP requests to the base station which will be handled by SystemController.java.

Minimal System

This section highlights the most barebone necessary functions needed to make our product useful. There are two main ideas. One is that the carts must move on their own, and the second is that the customer needs to be able to call a cart to his or her location. The following functions are absolutely necessary to realize the two ideas.

Customer-Cart Interaction Interface

Function 1: There must be button the users can use to call for a cart.

Robot Shopping Cart Locating System

Function 2: Carts need to be able to triangulate their own location for pathfinding and obstacle avoidance.

Robot Shopping Cart Path Finding

Function 1: Lane maps are needed for navigating the carts.
Function 2: Carts need to be able to detect obstacles.
Function 3: Carts need to be able to orient themselves for navigation and obstacle avoidance.

Note on Central Control

While Central Control is the crucial system that connects all listed features together, it is not necessary for the bare minimal system to work. For example, we don't need to have a central control to direct the carts to their destinations, we can have independent controls built into each smartphone app instead.

Enhancements

The possible future implementations include giving the store management the ability to customize the store layout in the application. The store management will be able to change the locations of the speed bumps, any temporary obstacles, restricted area etc on the system easily from the 'custom store layout option'. Only the authorized users will be able to access the functionality. Moreover, user authorized for accessing and managing the layout for a particular store will will not be able to access other stores' information unless they have been exclusively given the authority to do so.

Additional feature in the long term is having the carts deliver the products bought at the store to the user's home, provided the user leaves within a certain radius from the store. Here, instead of sending the carts to the cashier, the user will be able to checkout the items in the cart itself. The cart will have a machine to authorize debit and credit card transactions. If the user is paying with cash, he/she has to go through the cashier to checkout the items. If the user lives outside the radius covered by the shopping cart delivery system, the cart will alert the user of the situation. The user can either choose to deliver the products to the car or to the nearest bus station.

Future implementation for finding the carts after they have been sent to the checkout station is under consideration. The customer will be able to track down the cart(s) that has the user's items for at the checkout station from a crowd of similar looking carts full of items

Possible enhancements for checkout stations in the future include implementing an interface to checkout and store the total price for a particular user so that when the customer comes back for the items payment, it can be done immediately without having to go through the checkout process at the station.

Conclusion

The proposed project, Servr, focuses on benefiting both the departmental stores and their customers. The project enhances the customer experience in shopping by providing them the option to shop for more items which in turn increases the revenue of the store. The system consists of a mobile app, a central control system, and shopping cart robots. The control system will track the location of all the robots in the system in real time. The customer can request an empty cart to his or her location. The control system will determine a path with avoiding collision with the other carts and the stationary objects, then dispatch an available empty shopping cart to the requested location. The user can tag a full cart with their identity and send it to the cashier for checkout.

The possible implementations include the customer-cart interaction interface, robot shopping cart locating system. The future implementations include providing the store management the ability to modify the store layout at any time in the system. The process requires authorization for accessing the feature for each store. Another feature includes the shopping cart delivery system to the customer's home location within a certain radius. Future implementation of the locating the cart at the checkout station is under consideration.

Appendix A

Modifications

Changed checkout system from simple storage to check items as they arrive to checkout. Added to enhancements section.

Added to "needs" scan location identifiers within store.

Fixed typos.

Specified payment methods.

Consistency added throughout to reflect changes.

Added new route calculation as a solution to blockages.

Added awesome diagrams

Glossary

Almost Games The company that is researching, creating ,and developing the Servr system.

NFC Stands for Near Field Communication. It is a communication protocol which allows for electronic devices to communicate wirelessly with each other while in close proximity.

Nodes Devices which passively transmit/receive a signal.

Team

Brandon Mabey	The lead web developer for Almost Games. He is in his third year in his Software Engineering degree at the University of Victoria, and a member of the 1st place team in division II of the 2015 ACM-ICPC Pacific Northwest regionals.
Tania Akter	The analyst, editor and the lead software collaborator for Almost games. She manages the team conflicts at times. She is in third year Software Engineering degree at University of Victoria. She has worked with IBM for developing analytical software for providing visual statistics that are used for vital business decisions.
Haodong Tao	The lead product analyst and design architect. He is in his third/fourth year in the combined major of Computer Science and Health Information Science. He as worked on many projects before in classes like CSC375, which is systems analysis, and his experience will make this team awesome.
Tal Melamed	The lead solutions and design architect. He is in his third year(sort of) of a software engineering degree at the University of Victoria. He has worked at Intel developing industry leading solid state drives.